

Méthodes d'apprentissage supervisé appliquées à la conception d'un outil d'aide à la décision en anesthésie

Julien KOEUNE

Mémoire présenté sous la direction du **Prof. Gianluca BONTEMPI**
en vue de l'obtention du grade de Licencié en Informatique
Année académique 2007–2008

Remerciements

Je tiens à remercier tout particulièrement le Professeur Gianluca Bontempi pour les critiques, l'aide et les précieux conseils qu'il m'a prodigués tout au long de cette année,

Olivier Caelen pour son encadrement, les nombreuses relectures de mon mémoire et sa grande disponibilité,

le Docteur Luc Barvais pour avoir mis tout en oeuvre afin que je puisse réaliser les tests de mon outil en conditions réelles,

Olivier Cailloux qui m'a permis de résoudre de nombreux problèmes lors du déploiement de mon outil à l'hôpital Erasme,

Yves de Smet et Bernard Fortz de s'être intéressés à mon travail et d'avoir accepté de faire partie de mon jury,

Frédéric Lafitte d'avoir pris le temps de relire l'entièreté de mon mémoire,

Mon grand-père d'avoir corrigé de nombreuses fautes d'orthographe présentes dans les versions antérieures de ce travail,

Anais Dixheures de m'avoir aidé à corriger mon mémoire.

Je terminerai par témoigner ma reconnaissance à tous mes proches pour leur soutien et particulièrement à mes parents sans lesquels je ne serais jamais arrivé là où j'en suis à ce jour.

Encore merci à vous tous et à tous ceux de mon entourage que je n'ai pas cités.

Table des matières

Remerciements	ii
Table des notations	vi
1 Introduction	1
2 Méthodologie pour l'apprentissage supervisé	6
2.1 Les différents types de modèles d'apprentissage	8
2.2 Marche à suivre pour la construction d'un modèle de prédiction	10
2.3 Identification structurelle	11
2.3.1 L'erreur de généralisation d'un modèle	11
2.3.2 La validation de modèles	14
2.3.3 La sélection de variables	18
2.3.4 La sélection de modèles	19
2.4 Identification paramétrique	20
2.5 Le Lazy Learning	20
2.5.1 La fonction noyau	22
2.5.2 Génération des modèles locaux	23
2.5.3 Validation d'un modèle local	25
2.5.4 Algorithme du lazy-learning	25
3 L'apprentissage supervisé appliqué au domaine de l'anesthésie	27
3.1 Introduction de concepts fondamentaux en anesthésie	27
3.1.1 Anesthésie Intraveineuse à Objectif de Concentration	28
3.1.2 Déroulement de l'anesthésie	31
3.1.3 Monitoring de la profondeur de l'anesthésie	31
3.2 Apprentissage automatique appliqué à l'AIVOC	32
3.2.1 Infusion TOOLBOX 95	32
3.2.2 Prédiction du BIS	34
3.2.3 Prédiction de la variation de Propofol	40

TABLE DES MATIÈRES

3.3	Conclusion	43
4	Améliorations des modèles existants	44
4.1	Présentation des modèles existants	44
4.2	Construction de nouvelles variables d'entrée	45
4.3	Prédictions dans une fenêtre temporelle de 10 minutes	46
4.4	Données disponibles	47
4.5	Méthode d'évaluation des performances des différents modèles	49
4.6	Présentation des performances des modèles	51
4.6.1	Modèle linéaire sans variables additionnelles	51
4.6.2	Modèle LL sans variables additionnelles	52
4.6.3	Modèle linéaire avec variables additionnelles	53
4.6.4	Modèle LL avec variables additionnelles	55
4.6.5	Interprétation des résultats théoriques	55
5	BisPredict, un outil d'aide à la décision en anesthésie	57
5.1	Présentation de l'outil	57
5.2	Implémentation	59
5.2.1	Communication avec les appareils médicaux	59
5.2.2	Les différentes phases traversées par BisPredict	60
5.2.3	Diagramme des packages	61
5.2.4	Construction des variables	62
5.2.5	Détection des outliers	64
5.3	Analyse des risques	66
5.3.1	Tests à réaliser	67
5.3.2	Hypothèses de travail	68
5.3.3	Risques liés à BisPredict	69
5.4	Déploiement	69
5.4.1	Installation	69
5.4.2	Cas Test	69
5.5	Validation du modèle face aux données collectées en conditions réelles . .	70
5.5.1	validation statistique	71
5.5.2	Validation qualitative de BisPredict	71
5.6	Conclusion des tests	76
6	Conclusions et perspectives	77
A	Développements de calculs	79

TABLE DES MATIÈRES

B	Manuel de déploiement de BisPredict	82
C	Graphiques comparatifs des prédictions des différents modèles	85

Table des notations

AIVOC	Anesthésie Intraveineuse à Objectif de Concentration
AG	Anesthésie générale
Itb95	Infusion TOOLBOX 95
BIS	Index Bi-Spectral
$P_x(c) = \Pi(x)$	Générateur de données d'entrées
$P_f(y x)$	Dépendance stochastique qui transforme une entrée x en une sortie y
x	De manière générale, $x \in \mathbb{R}^n$
$f(x)$	Fonction cible
y	$y = f(x) \in \mathbb{R}$
LL	Lazy-Learning
MSE	Mean-Squared Error
NMSE	Normalized Squared Error
MISE	Mean Integrated Squared Error
MAE	Mean Absolute Error
P	Pourcentage de fois où les variations du BIS prédit ont été dans le même sens que celles concernant le BIS réel observé.
l-o-o	Leave-One-Out
l-o-o cv	Leave-One-Out Cross-Validation
G_N	Erreur de généralisation
\hat{G}_N	Estimation de l'erreur de généralisation
\hat{G}_N^{loo}	Estimation de l'erreur de généralisation par leave one out cross-validation
D_N	Ensemble d'apprentissage (dataset)
$z_i = (x_i, y_i)$	i^{eme} échantillon de l'ensemble d'apprentissage
N	Nombre d'échantillons contenus dans l'ensemble d'apprentissage
n	Nombre de variables d'entrée
α	Vecteur des paramètres d'un modèle
Λ^*	Ensemble de tous les types de modèles (structures) possibles
Λ	Un type (structure) de modèle
$E_\Lambda^{loo}(k)$	Erreur l-o-o d'un modèle Λ sur l'échantillon k
\hat{E}_i^{loo}	Estimation de l'erreur l-o-o pour l'échantillon $z_i \in D_N$
$h(\cdot, \alpha)$	Un modèle d'apprentissage
w	Bruit ajouté à $f(x)$
$C(f(x), h(x, \alpha))$	Fonction coût utilisée lors de l'apprentissage
$R_{emp}(\alpha)$	La fonction de risque empirique
α_N	$\arg \min_\alpha R_{emp}(\alpha)$

CHAPITRE 0. TABLE DES NOTATIONS

$\alpha_N^{(-i)}$	Vecteur des paramètres d'un modèle calculé sur l'ensemble d'apprentissage sans z_i
$E[X]$	Espérance de la variable aléatoire X
q	Query-point
k_m^c	Le nombre de voisins minimum pour le modèle constant dans le LL
k_M^c	Le nombre de voisins maximum pour le modèle constant dans le LL
k_m^l	Le nombre de voisins minimum pour le modèle linéaire dans le LL
k_M^l	Le nombre de voisins maximum pour le modèle linéaire dans le LL

Chapitre 1

Introduction

Depuis plusieurs années, on assiste à un effort visant à augmenter la sécurité et le confort des patients pendant les anesthésies. Malheureusement, l'anesthésiste n'est pas à l'abri d'une erreur de jugement qui pourrait avoir des conséquences dramatiques pour le patient. Malgré le fait qu'il soit rare qu'un patient décède pendant l'anesthésie, il a été prouvé qu'un mauvais dosage d'agents anesthésiques peut engendrer de lourdes répercussions sur les fonctions vitales du patient [1]. Il se peut par exemple que le patient se souvienne de l'opération après son réveil. Ce genre de situation a souvent abouti à de nombreuses plaintes médico-légales déposées par les patients [1].

Une fois dans le bloc opératoire, l'anesthésiste est chargé de choisir la technique d'anesthésie ainsi que les dosages de drogues adéquats à administrer au patient afin de stabiliser ses fonctions vitales. Ses choix sont fonction de son jugement, jugement qu'il détermine selon le profil du patient, c'est-à-dire en fonction de son âge, de sa taille, de son poids, de ses antécédents médicaux, etc. Bref, l'anesthésiste est le seul maître à bord.

Dans un tel contexte et au vu des responsabilités confiées à l'anesthésiste, il apparaît clairement qu'une assistance informatique n'est pas superflue. C'est dans cette optique que la Région Wallonne a débouqué des subsides pour deux projets de recherche réalisés au sein du Machine Learning Group (MLG) du Département d'Informatique de l'Université Libre de Bruxelles et ce en étroite collaboration avec le service de Anesthésie-Réanimation de l'hôpital Erasme. Le premier projet porte le nom de *TANIA* [2] pour Technologies pour l'ANesthésie Informatisée. Ce projet a comme objectif d'aider l'anesthésiste dans la conduite de l'anesthésie. Le second subside quant à lui a été octroyé dans le cadre d'un projet *First Europe Objectif 1* qui a permis d'extraire des informations précieuses à partir de données collectées par 15 chariots d'anesthésie intraveineuse. C'est dans le cadre du deuxième projet que j'ai réalisé mes propres recherches et abouti à la rédaction de ce mémoire.

La notion d'*anesthésie* présentée dans [4] vient du grec *αναίσθησια*, signifiant insensibilité d'un organisme ou également paralysie des sens. On entend par là l'abolition de toutes les sensibilités périphériques, qu'elles soient tactiles, kinesthésiques, douloureuses ou thermiques. La sous-entité analgésie ne décrit que la réalisation d'absence de douleur.

L'anesthésie s'efforce avant tout de permettre la réalisation d'interventions sur l'intégralité du corps humain sans douleur. Cet objectif principal peut être atteint par deux méthodes : d'un côté l'*anesthésie générale*, et de l'autre, l'*anesthésie loco-régionale*. L'anes-

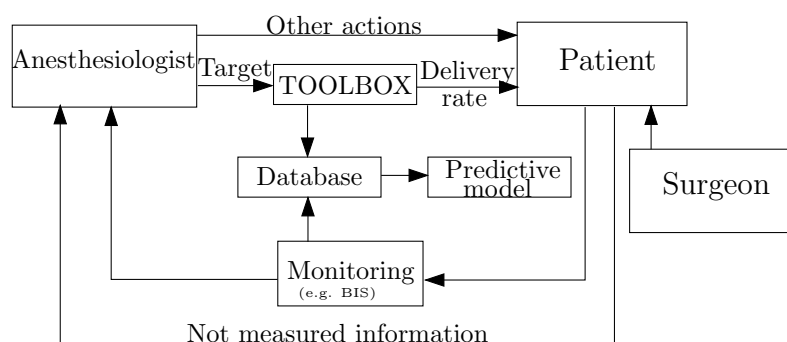


FIG. 1.1: Cette figure, issue de [3], met en évidence le fait qu'Itb95 (TOOLBOX) est un logiciel permettant d'assister l'anesthésiste dans son travail quotidien. Ce logiciel permet à l'anesthésiste d'implémenter ses algorithmes et d'automatiser le débit des perfusions en fonction des algorithmes. En plus d'aider l'anesthésiste, Itb95 permet d'enregistrer des données issues de l'anesthésie. La collecte de ces données a permis de créer un ensemble d'apprentissage D_N sur base duquel plusieurs modèles d'apprentissage ont été développés dans le cadre de ce mémoire.

thésie générale est synonyme de narcose, et se démarque des techniques d'anesthésie loco-régionale entre autres par l'abolition de l'état de conscience.

L'anesthésie loco-régionale, plus communément appelée anesthésie locale, est limitée à certaines parties du corps. L'anesthésie générale (AG) comprend l'anesthésie de tout le corps. Elle s'accompagne d'une diminution de la conscience. Pour la réalisation d'une AG, l'anesthésiste dispose d'agents inhalatoires (administrés par voie pulmonaire) et d'agents injectables par voie intraveineuse. Dans le cadre de ce mémoire nous nous limiterons aux AG par voie intraveineuse.

L'état de conscience comporte la perte de conscience (*hypnose*) et la perte des sensations de douleur (*analgésie*). Pendant la période d'anesthésie, il existe en général une perte de mémoire complète.

Utilisé et développé à l'hôpital Erasme, Infusion Toolbox 95 (Itb95) [5] est un logiciel qui permet d'assister l'anesthésiste dans l'administration des drogues. Itb95 assure entre autres le débit des perfusions selon le choix d'une concentration déterminée par l'anesthésiste dans une zone cible telle que le cerveau par exemple. En plus de contrôler le débit des perfusions, Itb95 exploite et archive les données provenant des appareils de monitoring. La place d'Itb95 au sein du processus d'anesthésie est illustrée par la figure 1.1.

C'est notamment grâce à ces données que les recherches dans le cadre du projet First Europe Objectif 1 ont abouti.

Les modèles d'apprentissage

Les modèles d'apprentissage aident le scientifique dans l'étude d'un phénomène physique pour lequel il dispose de données mais ne connaît aucune autre information pour lui permettre de mieux représenter ce phénomène. Un ensemble de données collectées résultant de l'observation du phénomène est stocké dans une base de données appelée ensemble d'apprentissage noté D_N . Ces données vont être utilisées afin d'apprendre le phénomène : on va créer un modèle d'apprentissage.

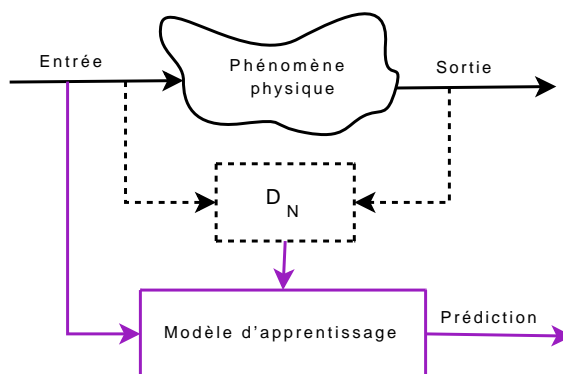


FIG. 1.2: Il existe des phénomènes naturels pour lesquels les scientifiques n'ont pas encore trouvé de bons modèles physiques. D_N est un ensemble d'observations du phénomène. Il est possible de paramétrer un modèle d'apprentissage sur les données observées. Ensuite, on pourra utiliser le modèle d'apprentissage afin de faire des prévisions pour ce phénomène.

Un modèle d'apprentissage peut être vu comme une fonction de type entrée/sortie avec des paramètres ajustables qui dépendent de la structure choisie pour construire le modèle. Il existe plusieurs manières de faire apprendre un modèle. Dans mon mémoire, je me focalise uniquement sur l'apprentissage supervisé. Cette technique a pour objectif de construire un modèle de prédiction uniquement sur base des données présentes dans D_N . Une fois que l'apprentissage est terminé on peut utiliser le modèle pour faire des prévisions sur le phénomène. Un tel processus est illustré par la figure 1.2.

Dans la branche de l'apprentissage supervisé, il existe un grand nombre d'algorithmes utilisés pour apprendre un modèle [6]. Deux exemples sont les modèles linéaires [7] et les modèles de type Lazy-Learning (LL) [7, 8]. Ces deux types de modèles reposent cependant sur des stratégies différentes. C'est ainsi que le modèle linéaire est un modèle global cherchant à modéliser un phénomène sur l'ensemble de son domaine d'entrée. Le Lazy-Learning quant à lui est une approche locale de l'apprentissage supervisé où LL construit un modèle chaque fois qu'une prédiction sur le phénomène est sollicitée. Le modèle construit par LL n'utilise que quelques échantillons présents dans D_N pour faire la prédiction.

L'apprentissage supervisé dans le domaine de l'anesthésie

A l'heure actuelle, et ce malgré le logiciel Itb95, les anesthésistes manquent encore de supports d'aide à la décision. De tels outils leur permettraient de suggérer ou de valider des actions devant être réalisées dans le cadre de leur travail quotidien. C'est dans l'optique de pallier à ce manque de supports que des chercheurs ont entrepris des recherches en appliquant des méthodes d'apprentissage supervisé au domaine de l'anesthésie. Parmi ces recherches, on peut citer [9, 10, 11, 12] qui ont permis de mettre en avant des techniques capables de prédire les variations d'agent anesthésique afin de maintenir le patient dans un état souhaité. De manière duale à ces travaux, des chercheurs du MLG ont, dans le cadre du projet First, créé un modèle d'apprentissage [3] capable de réaliser des prévisions sur l'évolution de la profondeur de l'anesthésie après que le praticien ait changé la concentration de l'agent hypnotique injecté au patient. Ce support

CHAPITRE 1. INTRODUCTION

d'aide à la décision permet de conforter l'anesthésiste dans ses choix lorsque le modèle a délivré des informations proches de ce à quoi le praticien s'attendait. Dans le cas contraire, cela permet à l'anesthésiste de reconsidérer le choix de l'action qu'il va entreprendre et éviter que l'action qu'il a choisi de réaliser ne mène le patient dans un état critique.

C'est dans ce contexte que j'ai débuté mon mémoire. Durant l'année, j'ai étudié des modèles alternatifs au modèle proposé dans [3]. Ces nouveaux modèles ont été construits sur base de données supplémentaires calculées durant la phase d'induction de l'anesthésie. Il s'agit de la phase pendant laquelle le patient va être soumis pour la première fois à une concentration élevée de l'agent hypnotique. Le but de ces variables additionnelles va être de représenter la façon dont le patient réagit aux drogues administrées de manière générale. Cela permet au modèle d'apprentissage de s'adapter au mieux à chaque patient.

En parallèle, j'ai eu l'occasion de travailler en étroite collaboration avec le Docteur Luc Barvais, chef du service d'anesthésie-réanimation à l'hôpital Erasme et Olivier Cailloux, chercheur responsable de la programmation du système informatique d'aide à la décision du projet *TANIA*. Cette collaboration fructueuse a abouti sur l'implémentation d'un logiciel, *BisPredict* qui embarque un modèle d'apprentissage capable d'effectuer des prédictions en temps réel lors d'anesthésies à l'hôpital Erasme.

Contributions principales de ce mémoire :

- **Création de nouvelles variables d'entrées :**
Un ensemble de variables d'entrées additionnelles a été créé à partir d'informations collectées pendant les dix premières minutes de l'anesthésie. Ces variables ont pour but de mieux calibrer le modèle au patient anesthésié. Une étude sur la plus-value d'informations apportée par ces variables a aussi été réalisée.
- **Sélection de variables :**
En plus de réduire la dimensionnalité du problème, la sélection de variables sur les différents modèles a permis de repérer les variables véhiculant le plus d'informations utiles.
- **Etude des performances de plusieurs modèles :**
Une étude comparative des performances des modèles linéaires et Lazy a été effectuée.
- **Programmation :**
J'ai implémenté un programme, intitulé *BisPredict*, destiné à fournir une aide à la décision en anesthésie. *BisPredict* interagit avec *Itb95* grâce au protocole de communication développé par O. Cailloux dans le cadre du projet *TANIA* [2].
- **Déploiement de BisPredict :**
L'implémentation du programme fut suivie d'une phase de déploiement pendant laquelle de nombreux tests ont été effectués afin de s'assurer de la bonne "entente" entre *BisPredict* et *Itb95*.
- **Tests en conditions réelles :**
Une fois les tests réussis, *BisPredict* a été utilisé en conditions réelles lors de trois

CHAPITRE 1. INTRODUCTION

interventions. Pendant ces interventions, en plus d'avoir fourni une aide à la décision à l'anesthésiste, BisPredict a collecté des données qui ont permis de réaliser une dernière validation sur base des données issues de l'anesthésie et non plus de l'ensemble d'apprentissage.

Structure du mémoire

- *Chapitre 1* - Introduction
- *Chapitre 2* - Méthodologie pour l'apprentissage supervisé.
Ce chapitre fait état de l'art en matière d'apprentissage supervisé. J'y introduis les notions qui m'ont été utiles dans le cadre de mes expériences. Après une présentation des différents types de modèles d'apprentissage, j'introduis des concepts permettant de construire un modèle de prédiction efficace. J'attache une grande importance au concept d'erreur de généralisation ainsi qu'aux techniques de validation. Deux types de modèles sont étudiés plus en détails : un modèle global linéaire et un modèle local de type Lazy-Learning.
- *Chapitre 3* - L'apprentissage supervisé appliqué au domaine de l'anesthésie.
Depuis deux décennies, des logiciels permettant d'aider l'anesthésiste dans son travail commencent à voir le jour. L'anesthésiste reste cependant le seul maître à bord, et de lourdes responsabilités lui incombent. Les conséquences de ses choix dans le déroulement des opérations est crucial et peuvent avoir des conséquences dramatiques sur le maintien des fonctions vitales du patient. Afin d'aider l'anesthésiste à prendre des décisions, plusieurs projets financés par la Région Wallonne ont vu le jour. Ce chapitre introduit de manière générale des concepts propres au domaine de l'anesthésie, ainsi que des outils permettant d'aider l'anesthésiste à prendre des décisions. L'étude de plusieurs modèles de prédiction et leurs performances sont aussi étudiés dans ce chapitre.
- *Chapitre 4* - Amélioration des modèles existants.
Sur base des modèles existants présentés dans le chapitre 3, de nouveaux modèles ont été construits afin d'améliorer les performances des modèles existants. Dans cette optique, de nouvelles variables construites lors des 10 premières minutes de l'anesthésie ont été ajoutées aux modèles. Une étude comparative des performances des modèles avec et sans ces variables additionnelles a été réalisée afin de mesurer la quantité d'informations utiles véhiculées dans ces nouvelles variables.
- *Chapitre 5* - BisPredict un outil d'aide à la décision en anesthésie.
Les résultats concluants issus de la validation des modèles construits dans le chapitre 4 m'ont permis d'implémenter un outil d'aide à la décision en anesthésie. L'objectif de cet outil est double : fournir une aide à la décision en anesthésie et collecter des données afin d'établir une validation sur un jeu de données collecté en temps réel lors de l'anesthésie. De plus, afin de sécuriser les prédictions, je me suis servi de techniques de détection d'outliers.
- *Chapitre 6* - Conclusions et perspectives.

Chapitre 2

Méthodologie pour l'apprentissage supervisé

Supposons qu'on veuille construire un outil d'aide à la décision en anesthésie. Une application intéressante pour les anesthésistes serait de pouvoir prédire l'état de conscience¹ du patient à l'instant $t + 30 \text{ sec}$, où t est l'instant lors duquel l'anesthésiste vient de modifier la concentration d'un agent anesthésiant dans le corps du patient.

Pour ce faire on va observer les valeurs prises par un ensemble d'inputs ou *variables d'entrée* lors de plusieurs anesthésies. De telles variables sont par exemple le poids, la taille et l'âge du patient. A ces variables d'entrée, on va associer la valeur de l'état de conscience observée à l'instant $t + 30 \text{ sec}$, on parle d'outputs ou de *variables de sortie*.

Sur base de ces données on va entraîner un modèle. Un modèle de prédiction peut être vu comme une boîte noire transformant un ensemble de variables d'entrée en une variable de sortie. Une fois le modèle entraîné, il sera capable d'effectuer des prédictions dans un contexte différent, notamment lors de l'anesthésie de nouveaux patients.

Un problème d'apprentissage supervisé peut être décrit par les éléments suivants [7] :

- un générateur de données d'entrée $x \in \mathcal{X} \subset \mathbb{R}^n$ indépendamment et identiquement distribuées suivant une distribution de probabilité inconnue mais fixe :

$$P_x(x) = \Pi(x) \quad (2.1)$$

- la dépendance stochastique qui transforme les entrées x en y où $y \in \mathcal{Y} \subset \mathbb{R}$ est décrite par :

$$P_f(y|x) \quad (2.2)$$

Le cas le plus simple serait $y = f(x) + w$ où w est une variable aléatoire de moyenne nulle et d'écart type fixe où $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$.

- un ensemble d'apprentissage $D_N = \{\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots, \langle x_N, y_N \rangle\}$ composé de N paires $\langle x_i, y_i \rangle \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ indépendamment et identiquement distribuées selon la distribution jointe x, y .

¹Typiquement un scalaire obtenu via un moniteur dont la valeur est comprise entre 0 et 100. La section 3.1.3 est dédiée à cette information.

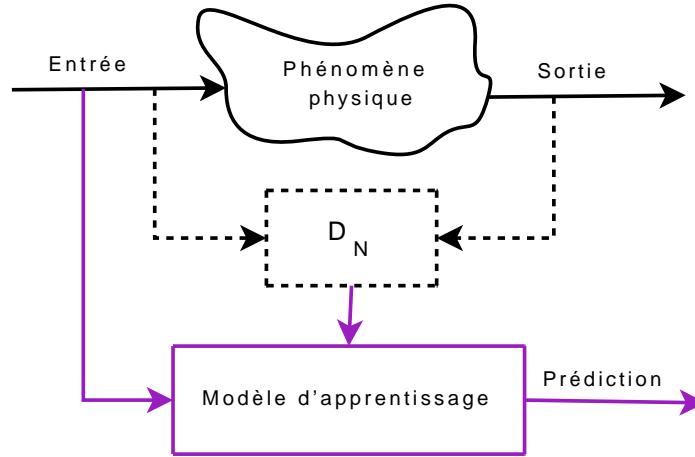


FIG. 2.1: Un phénomène physique $f()$ est observé. Grâce à ces observations nous avons pu créer D_N qui va être utilisé pour construire un modèle de prédiction. Une fois construit, ce modèle pourra effectuer des prédictions sur de nouvelles réalisations du phénomène.

$$P_z(z) = P(\langle x, y \rangle) = P_f(y|x)\Pi(x) \quad (2.3)$$

Le training set $D_N \in (\mathcal{X} \times \mathcal{Y})^N$ observé peut lui-même être vu comme la réalisation de la distribution jointe.

De manière générale, le schéma de l'apprentissage supervisé est présenté sur la figure 2.1.

Dans l'optique de pouvoir prédire la valeur prise par de nouvelles variables de sortie pour de nouveaux vecteurs d'entrée, il va falloir trouver un lien entre l'ensemble des variables d'entrée x_i et l'ensemble des variables de sortie $y_i \in \mathbb{R}$ que nous avons observées. Un tel problème est appelé une *régression*. Dans le cas où y_i est une variable discrète on cherchera à résoudre un problème de *classification* [13]. Au terme de la régression, notre modèle sera capable d'effectuer des prédictions pour de nouveaux inputs donnés. C'est ce qu'illustre la figure 2.3.

Au terme de la phase de collecte des données nous sommes donc en possession d'un ensemble d'apprentissage $D_N = \{z_1, z_2, \dots, z_N\}$ composé de N échantillons $z_i = \{x_i, y_i\}$ où $x_i \in \mathbb{R}^n$ est le vecteur comprenant les variables d'entrée tandis qu' $y_i \in \mathbb{R}$, une variable de sortie, est un scalaire. Supposons que les données aient été générées par le processus $y_i = f(x_i) + w_i$ où w_i est une variable aléatoire de moyenne nulle qui représente l'effet du bruit. La figure 2.2 illustre la réalisation d'un tel processus.

L'objectif de la régression est de trouver la dépendance entre les x_i et les y_i en se basant uniquement sur l'ensemble d'apprentissage et cela sans aucune connaissance de $f(\cdot)$. En d'autres termes, sur base de cet ensemble d'apprentissage, nous allons construire un modèle $h(\cdot, \alpha)$ en ajustant les paramètres α afin de minimiser l'erreur de la prédiction. Cette phase porte le nom d'*identification paramétrique* et sera détaillée dans la section 2.4. Une fois l'identification paramétrique terminée, on va alors pouvoir présenter un nouvel échantillon en entrée du modèle et faire une prévision. Les paramètres α dépendent de la structure du modèle choisie lors de l'*identification structurelle* détaillée dans la section 2.3.

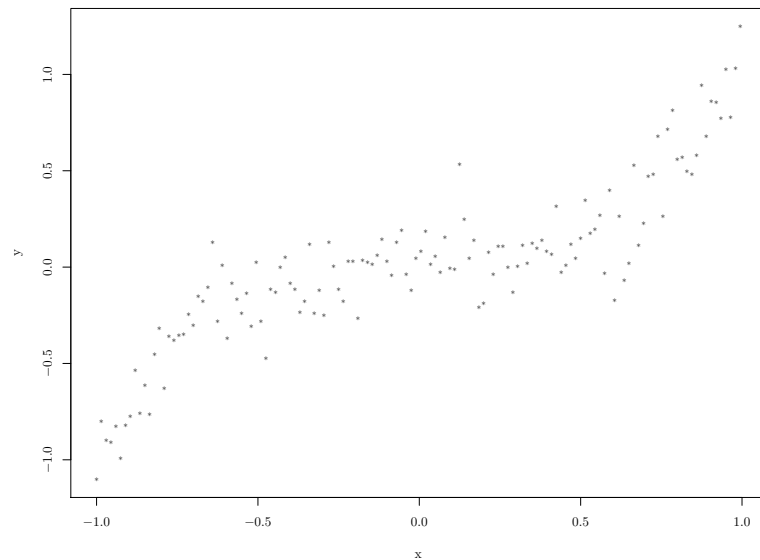


FIG. 2.2: Cette figure illustre le résultat de l'observation d'un phénomène physique. On s'est contenté d'observer la valeur y prise par le phénomène pour une seule variable d'entrée x . Les échantillons (*) sur le graphique représentent les entrées de l'ensemble d'apprentissage D_N .

Le but recherché est de trouver une fonction $h(\cdot, \alpha)$ qui se comporte le plus possible comme $f(\cdot)$.

Le processus d'apprentissage n'est pas un travail séquentiel. On peut voir cela comme deux boucles imbriquées : la boucle extérieure correspond à l'identification structurelle tandis que celle imbriquée se charge de l'identification paramétrique.

2.1 Les différents types de modèles d'apprentissage

On peut regrouper les modèles d'apprentissage au sein de deux catégories.

1. **Les modèles globaux** Dans cette catégorie de modèles d'apprentissage, on va tenter d'approcher la fonction $f(\cdot)$ sur l'ensemble du domaine. Plusieurs modèles appartiennent à cette catégorie. On peut citer par exemple le perceptron multicouches [14] ou encore les modèles linéaires qui feront l'objet d'une étude plus détaillée dans l'annexe A. Un exemple de modèle global est illustré sur la figure 2.3
2. **Les modèles divisés et conquis** Afin de rendre la tâche de modélisation plus facile, on va diviser le domaine de la fonction qu'on cherche à modéliser en plusieurs sous domaines. On fait cela dans le but d'utiliser des modèles faciles à calculer comme les modèles constants ou linéaires. Cette catégorie de modèles d'apprentissage peut encore être divisée en deux sous-catégories :
 - (a) **Les modèles modulaires** Ici on divise l'espace d'entrée lors de l'apprentissage. Ainsi on peut utiliser des modèles d'apprentissage différents sur chaque division du domaine des x_i . Le but de cette division est encore une fois d'utiliser des

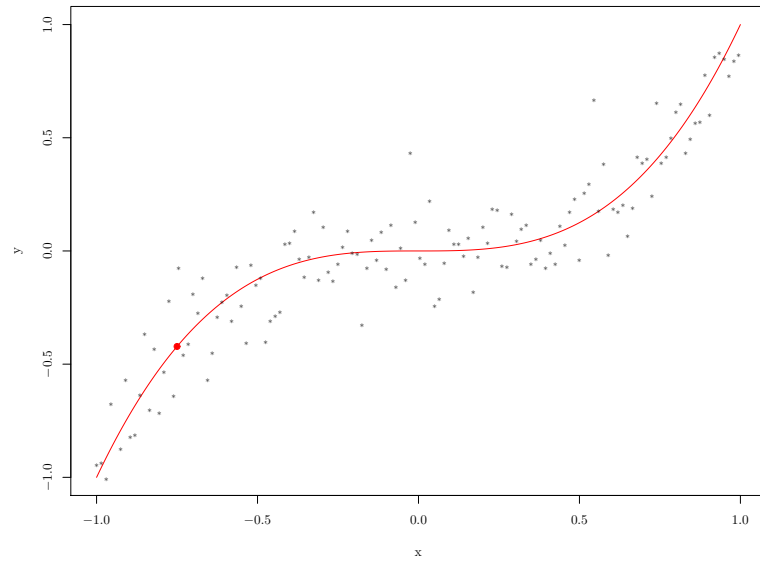


FIG. 2.3: La courbe en rouge nous donne les valeurs renvoyées par le modèle sur l'ensemble du domaine. Le modèle s'est entraîné sur les échantillons représentés par des astérisques (*). Ainsi quand on présente -0.75 en entrée du modèle, celui-ci nous renvoie -0.42 comme valeur.

modèles d'apprentissage simples. Par contre la recherche des divisions optimales est un problème d'optimisation non linéaire. Radial-Basis Function Network [15] et Multivariate Adaptive Regression Splines [16] sont des exemples de modèles modulaires. Un autre exemple de modèle modulaire est illustré sur la figure 2.4.

- (b) **Les modèles locaux** Dans le cas des modèles globaux, on commençait par estimer les coefficients de régression du modèle par rapport à l'ensemble d'apprentissage. Une fois que ce calcul était effectué, on pouvait alors commencer à faire des prévisions. La démarche suivie par les modèles locaux est assez différente. Ainsi, dans cette catégorie de modèles d'apprentissage, la phase d'apprentissage se résumera à la mémorisation de l'ensemble d'apprentissage. L'apprentissage sera effectué lorsqu'on souhaitera faire une prédiction pour un point q aussi appelé *query point*. Le modèle n'utilisera que quelques points proches dans l'espace des variables d'entrée du query-point afin de réaliser l'apprentissage. Le but ici est toujours d'utiliser des modèles les plus simples possible. C'est pour cette raison qu'on utilisera des modèles de types constants, linéaires et plus rarement polynomiaux. Dans la section 2.5, je présenterai le *Lazy-Learning* [7, 8] qui est un modèle d'apprentissage faisant partie de la catégorie des modèles locaux.

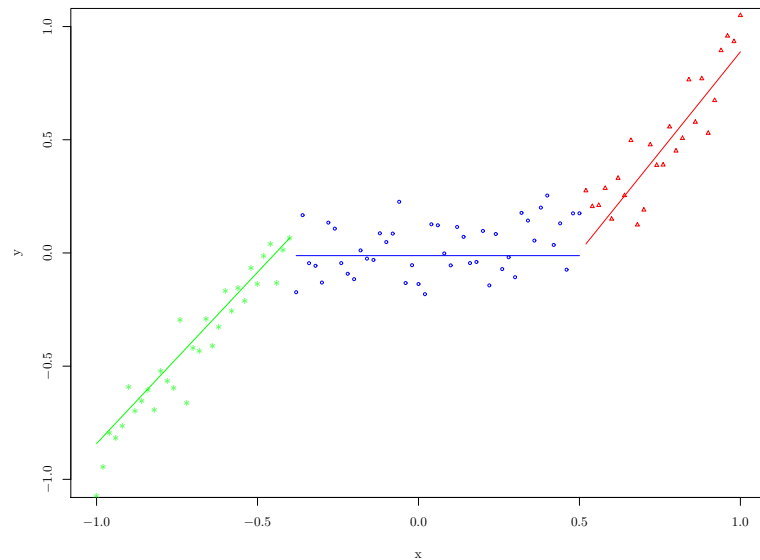


FIG. 2.4: Ce graphique illustre le cas où le modélisateur a choisi de découper le domaine en plusieurs sous-domaines afin d'utiliser des modèles simples. Ainsi, dans ce cas, l'espace a été découpé en trois parties et le modélisateur a utilisé trois modèles linéaires.

2.2 Marche à suivre pour la construction d'un modèle de prédiction

Le travail du modélisateur peut être découpé en deux grandes parties : la phase préliminaire et la phase d'apprentissage. Dans la première phase on débute avec la formulation d'un problème donné et on aboutit lorsqu'on dispose des données D_N prêtes à être utilisées pour l'apprentissage du modèle. Dans la seconde phase, sur base des données collectées, on va tenter de construire le meilleur modèle d'apprentissage $h(\cdot, \alpha_N)$.

– Phase préliminaire

- La formulation du problème : le modélisateur recherche le domaine d'application du phénomène et émet des hypothèses à son sujet. C'est ici qu'il va décider des variables d'entrée et de sorties qu'il souhaite observer. Il va aussi faire des hypothèses sur le processus stochastique sous-jacent à la génération des données.
- La phase expérimentale : le modélisateur va générer des échantillons les plus représentatifs du problème afin de créer un ensemble d'apprentissage optimal. Il est cependant rare que le modélisateur ait l'opportunité de générer lui-même les données. Il devra donc se contenter de données déjà disponibles.
- Le pré-traitement des données : il s'agit ici de transformer les données brutes obtenues lors de la phase expérimentale en données raffinées et prêtes afin d'être utilisées pour l'apprentissage. On peut par exemple retirer des échantillons où il manque de l'information, ou encore des échantillons dits *outliers* qui ne sont pas représentatifs de l'ensemble des données. C'est aussi dans cette phase qu'on pourra réduire la dimensionnalité du problème en appliquant des méthodes de

- type filtre [17] sur l'ensemble d'apprentissage. Principal Components Analysis (PCA) [17] est un exemple de méthode de type filtre.
- **Phase d'apprentissage** C'est dans cette phase que l'apprentissage va être réalisé. La procédure de recherche du modèle optimal est constituée de deux processus imbriqués : l'identification structurelle qui est le processus principal et l'identification paramétrique qui va ajuster les paramètres α du modèle en fonction des données présentes dans l'ensemble d'apprentissage D_N .
 - L'identification structurelle : il s'agit de la phase qui effectue le plus de travail. Elle va commencer par faire le choix d'une structure pour le modèle. Un réseau de neurones à 6 couches cachées par exemple. Dans le cas du lazy-learning, on va faire varier le nombre de voisins utilisés pour faire une prédiction. Si cela n'a pas été effectué par une méthode de type filtre [17] lors de la phase préliminaire, il convient aussi d'exécuter une procédure de sélection de variables afin de faire le tri parmi les variables d'entrée qui ont de l'importance et celles qui n'apportent pas ou peu d'informations sur le phénomène observé. L'identification structurelle s'occupe aussi de la *validation* des modèles (après que l'identification paramétrique ait été effectuée). La validation est un procédé qui va estimer la capacité du modèle à généraliser le phénomène observé. Une fois que la validation des modèles candidats est terminée, il reste à sélectionner la meilleure structure $\Lambda \subset \Lambda^*$ afin de garder le meilleur modèle. C'est le rôle de la *sélection de modèles*.
 - L'identification paramétrique : dans cette phase, sur base de l'ensemble d'apprentissage D_N construit dans la phase préliminaire et sur base d'une structure $\Lambda_s \subset \Lambda^*$, on va calculer les valeurs des paramètres qui vont minimiser l'erreur commise par le modèle sur D_N .

2.3 Identification structurelle

Il s'agit de la phase durant laquelle on va choisir une structure Λ d'un modèle parmi l'ensemble de toutes les structures de modèles Λ^* . La structure qui sera utilisée par le modèle dans le but de faire des prédictions sera choisie selon des critères bien précis expliqués dans les sections à venir.

Le rôle de cette phase est de déterminer la structure $\Lambda \subset \Lambda^*$ optimale sur base du calcul de l'erreur de généralisation G_N qui mesure l'erreur moyenne commise par les modèles construits sur base d'une structure Λ . Cette notion est expliquée dans la section 2.3.1. Pour ce faire il va falloir réaliser l'identification paramétrique 2.4, la validation 2.3.2 et la sélection de modèle 2.3.4.

Un autre rôle de l'identification structurelle est la *sélection de variables*. Il agit de déterminer quelles sont les variables d'entrée qui ont une influence significative sur la diminution de l'erreur de généralisation.

2.3.1 L'erreur de généralisation d'un modèle

Soit $h(\cdot, \alpha)$ un modèle avec comme paramètres $\alpha \in \Lambda$. A cela on associe la fonction *coût*, $C(f(x_i), h(x_i, \alpha))$ qui sert de mesure pour l'écart entre la prédiction $h(x_i, \alpha)$ et la valeur prise par la fonction $f(x_i)$ pour un input x_i donné appartenant à l'ensemble d'apprentissage.

CHAPITRE 2. MÉTHODOLOGIE POUR L'APPRENTISSAGE SUPERVISÉ

On définit l'erreur empirique d'un modèle $h(\cdot, \alpha)$ sur un ensemble d'apprentissage D_N comme ceci :

$$R_{emp}(\alpha) = \frac{1}{N} \sum_{i=1}^N C(y_i, h(x_i, \alpha)) \quad (2.4)$$

où x_i et y_i appartiennent à D_N . Il s'agit en fait de l'évaluation des performances du modèle lorsqu'on l'utilise sur D_N .

L'identification paramétrique (section 2.4) va, sur base de l'ensemble d'apprentissage D_N , nous fournir $h(\cdot, \alpha_N)$ avec $\alpha_N \in \Lambda$ tel que

$$\alpha_N = \arg \min_{\alpha \in \Lambda} R_{emp}(\alpha) \quad (2.5)$$

minimise l'erreur empirique.

A présent, si la fonction coût est le carré de la différence entre la prévision du modèle et la sortie désirée, $C(f(x), h(x, \alpha)) = (f(x) - h(x, \alpha))^2$, alors on parlera du MSE (*mean-square error*) pour la fonction suivante :

$$MSE(x) = E_{D_N}[C(f(x), h(x, \alpha_N))] = E_{D_N}[(f(x) - h(x, \alpha_N))^2] \quad (2.6)$$

Le $MSE(x)$ est une mesure de l'erreur moyenne commise par les modèles $h(\cdot, \alpha_N)$ lorsqu'ils effectuent une prédiction pour le point x lorsque le D_N change. Etant donné qu'on ne connaît pas la distribution des y_i , voici une méthode théorique pour calculer le MSE d'un modèle $h(x, \alpha)$ pour un point x donné :

1. Générer un ensemble d'apprentissage D_N
2. Exécuter l'identification paramétrique (équation (2.5)) afin de trouver l' α_N optimal
3. Calculer l'erreur $(f(x) - h(x, \alpha_N))^2$
4. Recommencer une infinité de fois depuis l'étape 1
5. Le MSE est la moyenne des erreurs obtenues en 3

L'adoption de la fonction coût dans sa forme quadratique nous permet de décomposer le MSE en la somme du biais au carré et de la variance du modèle. Le biais du modèle $BIAIS[h(x, \alpha)] = E_{D_N}[h(x, \alpha)] - f(x)$ et la variance du modèle $Var[h(x, \alpha)] = E_{D_N}[(h(x, \alpha_N) - E_{D_N}[h(x, \alpha_N)])^2]$. Le calcul de la décomposition est réalisée dans l'annexe A.

Il s'agit de la décomposition *BIAIS/VARIANCE* du MSE. On peut réécrire $MSE(x)$ comme ceci :

$$MSE(x) = \sigma_w^2 + BIAIS^2 + variance \quad (2.7)$$

Pour une entrée x donnée, le biais mesure la quantité d'erreurs entre la moyenne des sorties du modèle entraîné sur différents D_N et la valeur de la fonction qu'on cherche à modéliser.

La variance indique la sensibilité du modèle lorsqu'on change d'ensemble d'apprentissage. Une variance élevée indique que le modèle est potentiellement trop sensible au

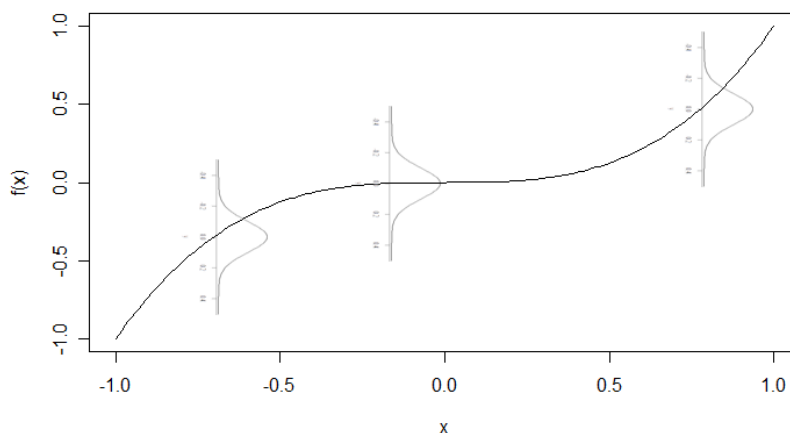


FIG. 2.5: Le calcul du MSE suppose qu'on connaisse la fonction de distribution des y . Cependant ce n'est pas le cas. On sait juste qu'elle suit une loi inconnue à laquelle vient s'ajouter un bruit. c'est-à-dire que pour un même x issu de la génération de différents datasets D_N , l'image de x sera différente. On a supposé que le bruit était une variable aléatoire de moyenne nulle et de variance constante. Ici, en supposant que le bruit est distribué selon une normale, l'image des x est distribuée comme sur le graphique, c'est-à-dire avec une grande probabilité d'être près de la moyenne. Le fait de générer une infinité de fois D_N sert à recréer cette distribution des y et ainsi nous permettre de s'approcher de la valeur de $f(x)$. Du bruit peut aussi entacher la mesure sur les x mais on le suppose négligeable.

bruit généré sur l'ensemble d'apprentissage. Une grande variance est souvent synonyme d'un modèle trop puissant. La variance est une information qui ne dépend que du modèle et pas de la fonction cible.

σ_w^2 est quant à lui la variance du bruit w .

Le MSE représente l'erreur quadratique du modèle en un point. L'erreur de généralisation étend le concept du MSE à l'ensemble \mathcal{X} du domaine de modélisation. On parle alors de *MISE* (Mean Square Integrated Error). La notation utilisée pour décrire l'erreur de généralisation est G_N telle que :

$$G_N = \int_{\mathcal{X}} \text{MSE}(x) d\Pi(x) \quad (2.8)$$

où $\Pi(x)$ est le générateur de la variable aléatoire $x \in \mathcal{X}$.

G_N est donc une fonction d'une structure Λ et de D_N à minimiser afin d'obtenir une structure Λ_s qui généralise le mieux $f(\cdot)$. Malheureusement comme $\Pi(x)$ et $P_f(x|y)$ sont inconnus, on ne peut calculer qu'une estimation de G_N . Le problème de l'estimation de l'erreur de généralisation apparaît lors de la *validation* du modèle.

Pour conclure, on dira qu'un modèle avec un petit biais et une grande variance a tendance à faire de l'*overfitting*, c'est-à-dire qu'il s'est trop focalisé sur les données d'apprentissage et n'a pas suffisamment généralisé le problème. De l'autre côté, un modèle

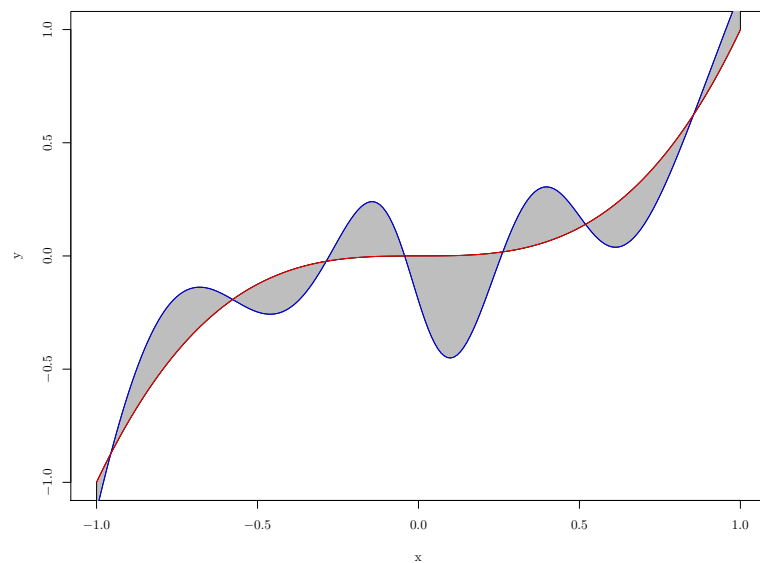


FIG. 2.6: La courbe rouge est la fonction $f(\cdot)$ que nous essayons d'estimer, tandis que la bleue représente le modèle $h(\cdot, \alpha)$. L'aire grisée entre les deux courbes représente l'erreur absolue commise par le modèle sur l'entièreté du domaine de $f(\cdot)$.

avec un grand biais et une petite variance fait de l'*underfitting*. Dans ce cas-ci, il n'a pas suffisamment appris. Ces deux cas représentent des modèles inefficaces. Afin de construire un modèle efficace, le modélisateur devra trouver un compromis entre le biais et la variance du modèle sur base de l'ensemble d'apprentissage comme seule donnée. Ce problème est plus connu sous le nom *trade-off biais/variance* [7].

2.3.2 La validation de modèles

Sur la figure 2.6, l'aire grisée entre les deux courbes est l'erreur absolue commise par un modèle donné. Au plus cette quantité est proche de zéro au plus on dira que le modèle est précis.

Pour chercher le modèle optimal, on utilise G_N qui mesure l'erreur moyenne commise par les modèles d'une certaine structure Λ .

Voici une analogie qui permet d'illustrer le problème qu'engendre le calcul de l'erreur de généralisation.

L'apprentissage peut être comparé à la situation d'étudiants qui doivent étudier en vue d'une interrogation. Pour ce faire, ils disposent d'un syllabus de théorie et des exercices résolus lors des séances de tp. Cela correspond au D_N . Le but de l'interrogation est de tester les étudiants afin d'évaluer leurs connaissances. Il y a deux types de professeurs :

- les professeurs cools : ils demandent aux étudiants de résoudre des exercices vus pendant l'année.
- les professeurs sévères : ils demandent aux étudiants de résoudre des exercices différents de ceux vus au cours de l'année.

CHAPITRE 2. MÉTHODOLOGIE POUR L'APPRENTISSAGE SUPERVISÉ

Le professeur cool va tester la mémorisation du cours tandis que le sévère se focalise sur la capacité de généralisation de l'étudiant. Un étudiant qui n'a rien compris et qui a étudié le cours par coeur pourra donc bien réussir l'interrogation proposée par le professeur cool. En contre partie il n'aura aucune chance de réussir l'interrogation du professeur sévère.

C'est typiquement ce genre de situations qui se retrouve dans le domaine de l'apprentissage.

Pour en revenir à la validation, une fois qu'on a choisi la structure Λ du modèle et que l'identification paramétrique est terminée nous disposons d'un modèle $h(\cdot, \alpha_N)$. La validation consiste à estimer l'erreur de généralisation (notée G_N) de ce modèle. Malheureusement on ne dispose que d'un nombre limité d'échantillons. De ce fait, on utilisera ces mêmes échantillons pour l'apprentissage et pour la validation.

L'identification structurelle pour un réseau de neurones peut par exemple consister à calculer l'estimation de l'erreur de généralisation pour des structures comprenant un nombre de couches cachées différent afin de dégager la structure engendrant l'erreur de généralisation la plus petite.

Ainsi, lorsqu'on souhaite estimer l'erreur de généralisation d'un modèle, la première idée qui vient à l'esprit est de calculer son erreur empirique

$$R_{emp}(\alpha_N) = \frac{1}{N} \sum_{i=1}^N (y_i - h(x_i, \alpha_N))^2 \quad (2.9)$$

où x_i et y_i appartiennent à D_N .

La valeur de R_{emp} sera toujours inférieure à celle de G_N car le modèle s'est entraîné sur les données D_N . Ceci montre que R_{emp} est un estimateur biaisé de G_N .

Un modèle faisant de *overfitting* est caractérisé par un risque empirique très faible et une grande variance. C'est le cas de l'étudiant qui ne comprend rien et qui étudie par coeur. C'est ce phénomène qui est illustré sur la figure 2.7 où le modèle à été construit sur base d'une structure Λ trop riche. En contre partie, un modèle ayant un risque empirique plus élevé n'est pas forcément un mauvais estimateur. C'est ce qu'illustre la figure 2.8.

C'est notamment pour cette raison qu'il est fortement déconseillé d'utiliser R_{emp} comme estimateur de G_N . Pour pallier ce manque d'efficacité, on va utiliser des techniques de rééchantillonnage.

Utilisation d'un test-set

La première solution est d'utiliser un *test-set*. Dans cette situation, on dispose d'un jeu de données, D_{tr} de taille N_{tr} destiné à l'apprentissage et un autre D_{ts} , de taille N_{ts} , réservé pour la validation.

$$D_{ts} = \left(\langle x_{N_{tr}+1}^{ts}, y_{N_{tr}+1}^{ts} \rangle, \dots, \langle x_{N_{tr}+N_{ts}}^{ts}, y_{N_{tr}+N_{ts}}^{ts} \rangle \right) \quad (2.10)$$

L'estimation de l'erreur de généralisation est dans ce cas-ci

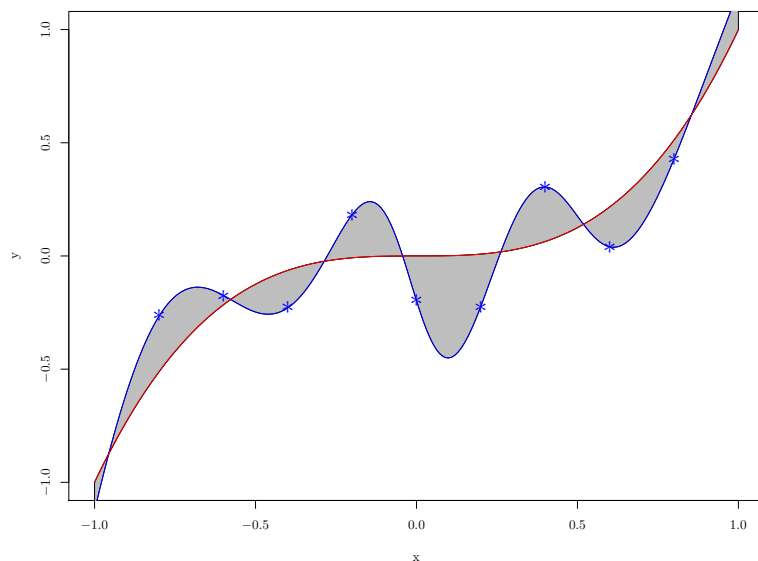


FIG. 2.7: Les étoiles (*) bleues représentent le training set (les échantillons destinés à l'apprentissage). La courbe rouge est la fonction $f(\cdot)$. La courbe bleue représente les valeurs prises par le modèle sur l'entièreté du domaine de $f(\cdot)$. Le modèle a très (trop) bien appris, ce qui se traduit par un risque empirique nul. On voit pourtant que l'erreur de généralisation (la partie grise entre les deux courbes) est élevée. On dit que ce modèle fait de l'overfitting sur les données. Il ne généralise pas assez la fonction $f(\cdot)$ que nous cherchons à estimer. A nouveau il est important de rappeler que le risque empirique n'est pas un bon estimateur de l'erreur de généralisation.

$$R_{ts}(\alpha) = \frac{1}{N_{ts}} \sum_{i=N_{tr}+1}^{N_{tr}+N_{ts}} (y_i^{ts} - h(x_i^{ts}, \alpha_{N_{tr}}))^2 \quad (2.11)$$

Malheureusement il est rare en pratique de disposer d'un tel surplus d'informations car on préférera l'utiliser pour l'apprentissage. Pour remédier à ce problème on aura recours à des techniques dites de *resampling* telles que la *k-fold cross-validation* [18].

K-fold cross-validation

Cette technique itérative va construire un modèle sur une partie de l'ensemble d'apprentissage et effectuer les prédictions sur la partie restante des données. L'ensemble d'apprentissage est divisé en k parties. Voir figure 2.9. La première est un ensemble d'apprentissage et la seconde est un ensemble destiné à la validation. Ces 2 ensembles contiennent respectivement N_{tr} et $N_{ts} = N - N_{tr}$ échantillons. A chaque étape $i = 1, \dots, k$, N_{tr} échantillons sont utilisés pour réaliser l'identification paramétrique tandis que les N_{ts} restants sont utilisés pour estimer l'erreur de $h(\cdot, \alpha_{N_{tr}}^i)$. Ainsi pour chaque $h(\cdot, \alpha_{N_{tr}}^i)$ on a :

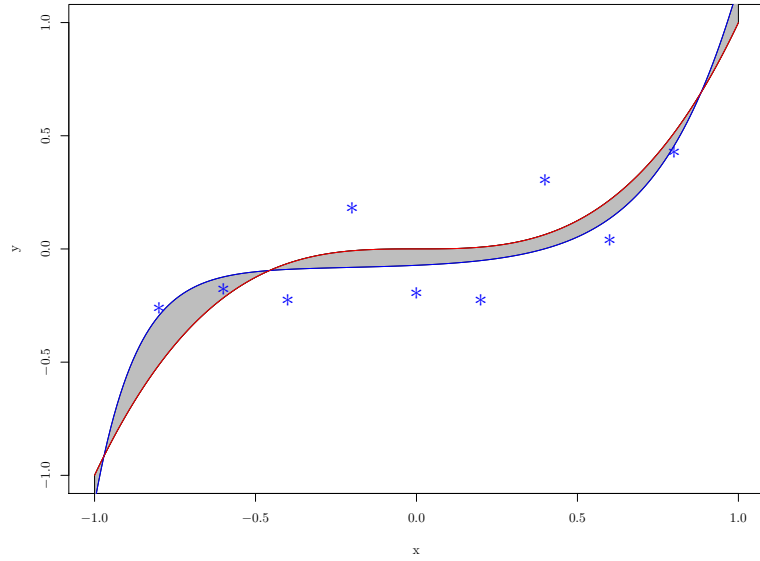


FIG. 2.8: Sur ce graphique, on voit que l'erreur empirique est plus élevée que celle du modèle de la figure 2.7. Cependant, ce modèle généralise mieux la fonction $f(\cdot)$ que nous cherchons à estimer.

$$R_{ts}(\alpha_{N_{tr}}^i) = \frac{1}{N_{ts}} \sum_{j=1}^{N_{ts}} C(y_j, h(x_j, \alpha_{N_{tr}}^i)) \quad (2.12)$$

L'estimation de l'erreur de généralisation via K-fold cross-validation \hat{G}_{cv} est le résultat de la moyenne arithmétique des $R_{ts}(\alpha_{N_{tr}}^i)$ avec $i = 1, \dots, k$ où C est la fonction coût définie dans la section 2.3.1.

$$\hat{G}_k^{cv} = \frac{1}{k} \sum_{i=1}^k R_{ts}(\alpha_{N_{tr}}^i) \quad (2.13)$$

C'est un processus qui peut prendre beaucoup de temps car on effectue k apprentissages. Il conviendra de trouver une valeur de k adéquate en fonction du modèle utilisé pour l'apprentissage. Plus k est grand plus la variance de l'estimateur de l'erreur de généralisation sera petite. $k = 10$ semble être un bon compromis d'après [19].

Pour des modèles dont la phase d'apprentissage est rapide, les modèles linéaires ou les modèles locaux où l'apprentissage consiste à sauvegarder l'ensemble d'apprentissage en mémoire par exemple, on pourra choisir $k = N$. Cela nous amène au cas extrême de la cross-validation : la *leave-one-out cross-validation* (*l-o-o cv*).

Dans ce cas, la taille du test set est réduite à un seul échantillon z_i alors qu'on effectuera l'apprentissage sur le reste de l'ensemble d'apprentissage. On notera \hat{G}_N^{loo} l'estimation de l'erreur de généralisation obtenue par l-o-o cv

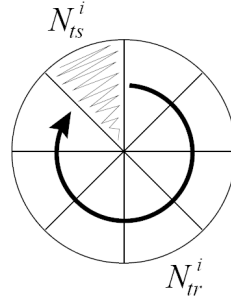


FIG. 2.9: Ce schéma [7] illustre le partitionnement de l'ensemble d'apprentissage lors de la $i^{\text{ème}}$ étape du processus de cross-validation. N_{tr}^i est le nombre d'échantillons destinés à l'apprentissage tandis que N_{ts}^i est le nombre d'échantillons destinés à la validation.

$$\hat{G}_N^{loo} = \frac{1}{N} \sum_{i=1}^N C(y_i, h(x_i, \alpha_N^{(-i)})) \quad (2.14)$$

où $h(\cdot, \alpha_N^{(-i)})$ est le modèle construit sur l'ensemble d'apprentissage privé de l'échantillon z_i .

2.3.3 La sélection de variables

La sélection de variables consiste à sélectionner un sous-ensemble des variables d'entrée afin de maximiser les performances de l'apprentissage du modèle. On fait le tri entre les variables qui contiennent de l'information utile et les autres qui ne véhiculent pas suffisamment d'informations. De plus réduire le nombre de variables d'entrée du modèle facilitera le problème de modélisation notamment en accélérant le processus d'identification paramétrique.

Il existe plusieurs techniques afin de réduire les variables d'entrée d'un modèle. Il y a par exemple l'analyse des composantes principales (PCA) [17] ou encore des techniques dites de wrapping [20].

Techniques de wrapping

Cette méthode crée un sous-ensemble de variables avec les variables d'entrée qui contiennent de l'information utile pour un modèle donné. On cherche donc à construire cet ensemble en utilisant le type de modèle retenu lors de l'identification structurelle.

On peut voir cette technique comme une recherche dans un espace $W \in \{0, 1\}^n$ où un vecteur $w \in W$ est de la forme

$$w[i] = \begin{cases} 0 & \text{si l'input } x_i \text{ ne fait pas partie du sous-ensemble} \\ 1 & \text{si } x_i \text{ appartient à ce sous-ensemble} \end{cases}$$

On cherche donc le vecteur $w^* \in W$ tel que

$$w^* = \arg \min_{w \in W} MISE_w \quad (2.15)$$

où $MISE_w$ est l'erreur de généralisation du modèle construit sur base de l'ensemble de variables w^* . Cependant comme c'est une valeur inconnue on la remplacera par \widehat{MISE}_w qui est une estimation de l'erreur de généralisation.

Le problème de cette technique est le fait qu'on doit réaliser 2^n apprentissages afin de déterminer le meilleur sous-ensemble de variables. En pratique, cette technique est inutilisable car le coût en temps de calcul est trop élevé et cela même pour un n assez petit.

Forward Selection

Il s'agit d'une technique permettant de réduire le nombre de sous-ensembles à tester lors de la sélection de variables lorsqu'on utilise la technique de wrapping.

La procédure commence par considérer un sous-ensemble de variables vide. La première variable sélectionnée est celle qui engendre la plus petite erreur de généralisation. La seconde variable sélectionnée est celle qui associée à la précédente engendre à son tour la plus petite erreur de généralisation. On continue ce raisonnement jusqu'à ce qu'on ait atteint un seuil sur l'erreur de généralisation ou qu'on ait considéré toutes les variables d'entrée possibles. L'ensemble w^* retenu est celui qui a généré la plus petite erreur de généralisation. Dans ce cas-ci, on effectuera $n!$ apprentissages au lieu de 2^n .

D'autres techniques de wrapping, assez similaires à la forward selection, existent. On peut ainsi citer la *Backward Selection* qui au contraire de la forward selection commence avec toutes les variables et on retire à chaque étape la variable qui induit la plus grande erreur de généralisation.

Autres techniques

Hormis les techniques de wrapping qui dépendent du modèle utilisé, on peut citer les techniques de type filtre qui ne dépendent pas du modèle utilisé. On retrouve entre autres parmi ces méthodes :

- *PCA* : qui a pour but de projeter les données de l'espace d'origine dans un espace de dimension réduite en essayant de minimiser au maximum la perte d'information. C'est ainsi que PCA crée un nouveau set d'axes, correspondant aux composantes principales, orthogonaux. [17]
- Les techniques de *ranking* [13] qui consistent à classer les variables d'entrée selon leur corrélation avec l'output.

2.3.4 La sélection de modèles

Le but de cette phase est de déterminer la structure finale du modèle qui sera utilisé dans l'optique de faire des prédictions.

Au terme de l'étape de validation des modèles candidats construits sur base des structures Λ_s , $s = 1, \dots, S$, nous disposons, pour chacune de ces structures, de la mesure

$\hat{G}_N(\Lambda_s)$ correspondant à l'erreur de généralisation.

La technique *winner-takes-all* choisit simplement la structure

$$\bar{s} = \arg \min_s \hat{G}_N(\Lambda_s) \quad (2.16)$$

qui minimise l'erreur de généralisation des structures candidates. Le modèle retenu au terme de la sélection est par conséquent $h(\cdot, \alpha_N^{\bar{s}})$.

Il existe néanmoins une autre technique de sélection de modèle. Il s'agit de la *combinaison de modèles* [21]. Cette technique s'avère être plus efficace que l'approche *winner-takes-all* car elle a comme conséquence de réduire la variance du modèle final.

2.4 Identification paramétrique

Une fois qu'on a fixé la structure d'un modèle, il faut identifier les paramètres α correspondant à cette structure Λ du modèle $h(\cdot, \alpha)$. L'identification paramétrique consiste à trouver les paramètres α_N qui minimisent le risque empirique

$$R_{emp}(\alpha) = \frac{1}{N} \sum_{i=1}^N C(y_i, h(x_i, \alpha)) \quad (2.17)$$

où C est la *fonction coût* qui représente l'erreur commise par $h(\cdot, \alpha)$ pour le point x et qui dans le cas d'une fonction quadratique est définie par :

$$C(y, h(x, \alpha)) = (y - h(x, \alpha))^2 \quad (2.18)$$

Il s'agit typiquement d'un problème d'optimisation qui dépend de la structure Λ choisie lors de l'identification structurelle

$$\alpha_N = \arg \min_{\alpha \in \Lambda} R_{emp}(\alpha) \quad (2.19)$$

Ainsi on utilisera par exemple la méthode dite de rétro-propagation du gradient [14] dans le cas des réseaux de neurones ou la méthode *least-squares* [7] dans le cas d'un modèle linéaire. Cette méthode est détaillée dans l'annexe A.

2.5 Le Lazy Learning

Le Lazy-Learning (LL) [7, 8] est une méthode d'apprentissage automatique faisant partie de la catégorie des modèles locaux introduits dans la section 2.1. Comme son nom l'indique, LL est une méthode d'apprentissage paresseuse, dans le sens où elle attend la dernière minute pour réaliser une prédiction. Il n'y a pas de phase d'apprentissage à proprement parlé, si ce n'est la mémorisation de l'ensemble d'apprentissage. Les calculs sont effectués au moment où l'on désire effectuer une prédiction pour le point q , aussi appelé *query point*. Lorsqu'une prévision est sollicitée, on va effectuer le calcul des paramètres

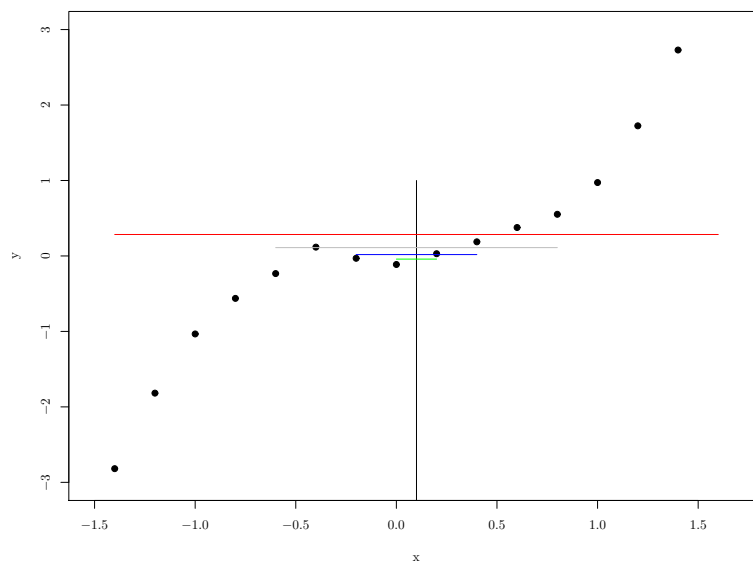


FIG. 2.10: Génération de modèles constants locaux. Il s'agit de l'application du LL pour une prédiction pour le point q . Dans ce cas-ci, LL va construire quatre modèles constants, en calculant la moyenne des sorties des plus proches voisins. Le vert a été construit en prenant les 2 plus proches voisins. Le bleu a été construit en prenant les 4 plus proches voisins. Le gris en prenant les 8 plus proches voisins et pour terminer, le rouge, en prenant les 16 plus proches voisins.

en ne considérant que les plus proches voisins de q . Plusieurs modèles locaux vont être construits en faisant varier le nombre de voisins de q . Ces modèles locaux utilisés par LL sont typiquement des modèles constants ou linéaires. Les figures 2.10 et 2.11 illustrent respectivement ce genre de modèles locaux. On peut aussi utiliser des modèles quadratiques, mais on s'éloigne alors de la philosophie de la technique qui est de construire des modèles les plus simples possible. La raison pour laquelle on n'utilise pas de modèles polynomiaux réside dans le fait qu'on utilise peu de points d'apprentissage et donc qu'un modèle même quadratique risque de faire de l'overfitting dans ces conditions.

La figure 2.11 est un exemple d'application de l'algorithme LL pour une prédiction au point q . Le domaine d'apprentissage est représenté par les points noirs. On souhaite effectuer une prédiction pour le point 0.1. Le LL va construire plusieurs modèles linéaires locaux en faisant varier le nombre de voisins de q .

Pour chacun de ces modèles locaux, LL va calculer une estimation de l'erreur de généralisation. Il existe 2 approches pour la sélection finale du modèle local.

- La technique du winner-takes-all vue dans la section 2.3.4, où LL choisit le modèle qui a la plus petite erreur de généralisation pour réaliser la prédiction.
- La combinaison de modèles qui va construire le modèle local définitif par combinaison des modèles construits sur base d'un nombre de voisins différents.

J'ai jusqu'ici parlé de plus proches voisins sans pour autant introduire la notion de *distance*. Il va donc falloir définir une fonction distance pour trier les voisins selon leur distance par rapport au point q . Il va aussi falloir définir la fonction noyau, qui évaluera

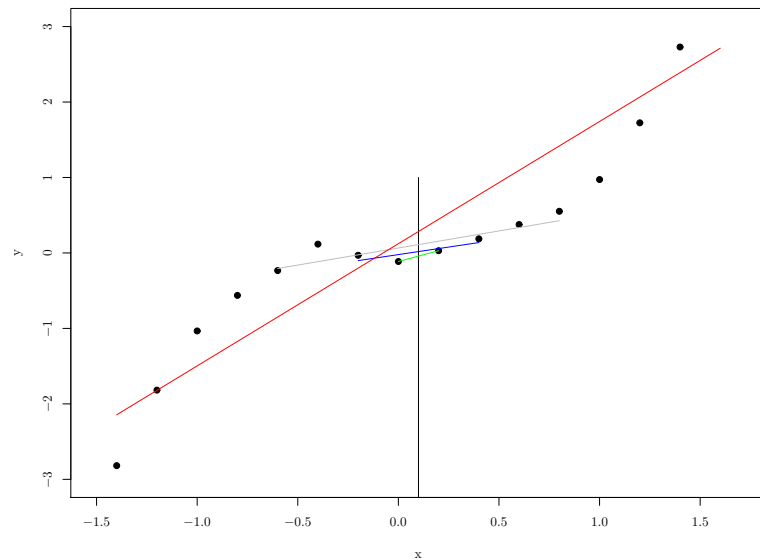


FIG. 2.11: Génération de modèles linéaires locaux. Il s'agit de l'application du LL pour une prédiction pour le point q . Dans ce cas-ci, LL va construire quatre modèles linéaires. Les paramètres de ces modèles locaux sont calculés en utilisant la méthode des least-squares. Le vert a été construit en prenant les 2 plus proches voisins. Le bleu a été construit en prenant les 4 plus proches voisins. Le gris en prenant les 8 plus proches voisins et pour terminer, le rouge, en prenant les 16 plus proches voisins.

la pertinence de l'information fournie par un point dans D_N en fonction de sa distance par rapport à q .

Il est important de rappeler que les modèles utilisent des modèles simples comme par exemple des modèles constants ou des modèles linéaires. Des techniques permettant d'accélérer la validation existent pour de tels modèles. Par exemple, *PRESS* [7, 22] (PRediction Sum of Squares) permet d'accélérer le calcul de l'erreur de généralisation par leave-one-out cross-validation.

2.5.1 La fonction noyau

La fonction noyau ou kernel est une fonction $K(x, q, B)$ qui pour un point donné x va fixer son importance dans la régression quand on veut faire une prédiction pour le point q . Cette fonction dépend de la distance entre x et q et du paramètre B . Elle est définie par :

$$0 \leq K(x, q, B) \leq 1 \quad (2.20)$$

$$K(q, q, B) = 1 \quad (2.21)$$

On distingue deux types de fonction noyau.

- Les fonctions noyaux de type uniforme :

$$K(x, q, B) = \begin{cases} 1 & \text{si } \|x - q\| < \frac{B}{2} \\ 0 & \text{sinon} \end{cases}$$

C'est-à-dire que tous les échantillons de l'espace d'entrées qui sont à une distance inférieure à $\frac{B}{2}$ de q vont être considérés lors de la régression.

- Les fonctions noyaux de type gaussienne :

$$K(x, q, B) = \exp\left(-\frac{\|x-q\|^2}{B^2}\right) \quad (2.22)$$

Dans ce cas-ci, au plus la distance séparant x de q est grande au moins on attachera d'importance au point x lors de la construction du modèle.

D'autres exemples de fonctions noyaux sont donnés dans l'annexe F de [7].

Par la suite j'utilise la fonction noyau uniforme car elle permet d'accélérer l'identification structurale et paramétrique des modèles locaux en utilisant la méthode PRESS décrite dans [7, 22].

2.5.2 Génération des modèles locaux

Le Lazy Learning génère un grand nombre de modèles locaux lorsqu'on souhaite effectuer une prédiction pour un query point q . Le nombre de voisins utilisés pour la construction du modèle est le seul paramètre structurel qui varie dans ces différents modèles. LL doit chercher la structure optimale de ces modèles locaux. Dans cette optique nous allons adopter une recherche dans laquelle le paramètre structurel est le paramètre B de la fonction noyau.

Supposons :

1. qu'une métrique dans l'espace des entrées est définie.
2. que la fonction $d(x, q)$ nous donne la distance entre le query point et x . Cette fonction peut par exemple être définie en utilisant la notion de distance euclidienne. Ainsi,

$$d(x, q) = \sqrt{\sum_{j=1}^n (x_j - q_j)^2} \quad (2.23)$$

3. que la paire $\langle x(k), y(k) \rangle$ est le $k^{\text{ième}}$ plus proche voisin du query point appartenant à D_N . Les échantillons sont triés selon leur distance par rapport à q tel que

$$d(x(i), q) \leq d(x(j), q) \quad \forall i \leq j \quad (2.24)$$

4. que le paramètre B est défini comme une fonction du nombre de voisins désirés du query point. $B(k)$ est égal à la distance séparant le query point du $k^{\text{ième}}$ voisin.

$$B(k) = d(x(k), q) \quad (2.25)$$

Il peut être vu comme la bande passante autour de q dans laquelle se trouvent les points utilisables pour la régression.

5. que k prend une série de valeurs dans un domaine borné inférieurement par k_m et supérieurement par k_M . Ces bornes désignent respectivement le nombre minimal de voisins et le nombre maximal de voisins.

Dès lors, le problème de l'identification structurelle locale peut être vu comme un problème d'optimisation de $B(k)$ qui est le voisinage de q considéré comme optimal avec k prenant des valeurs dans $[k_m, k_M]$. Lorsque k prend des valeurs dans $[k_m, k_M]$, la procédure de génération de modèles locaux va créer un ensemble de modèles locaux avec un nombre k de voisins. $B(k)$ est en fait la limite de proximité par rapport au point q pour laquelle on s'intéressera aux échantillons de l'ensemble d'apprentissage à retenir pour l'apprentissage. Ainsi lors de l'apprentissage, on ne considèrera que les points x de l'ensemble d'apprentissage tels que $d(x, q) < B(k)$. Les modèles générés sont des modèles constants et linéaires.

Le modélisateur devra donc définir les quatre paramètres suivants : k_m^c, k_M^c, k_m^l et k_M^l où k_m^c est le nombre de voisins minimum pour le modèle constant et k_M^c est le nombre de voisins maximum toujours dans le cas d'un modèle constant. L'interprétation de k_m^l et k_M^l sont les mêmes bornes, mais dans le cas de modèles linéaires cette fois.

Identification paramétrique d'un modèle constant et local

Afin de réaliser l'identification paramétrique d'un modèle local constant, il convient de calculer

$$h(q) = \frac{\sum_{i=1}^N K(x_i, q, B) y_i}{\sum_{i=1}^N K(x_i, q, B)} \quad (2.26)$$

$h(q)$ retourne la prédiction pour le query-point q dans le cas de modèles locaux constants.

Comme on a supposé que $K(\cdot, \cdot, \cdot)$ est une fonction noyau de type uniforme, le calcul de $h(q)$ revient à effectuer une moyenne arithmétique sur les sorties des k plus proches voisins de q :

$$h(q) = \frac{1}{k} \sum_{i=1}^k y(i) \quad (2.27)$$

Il existe une méthode récursive pour l'identification paramétrique d'un modèle constant et local [8]

Identification paramétrique d'un modèle linéaire et local

Le least-squares permet de calculer les paramètres optimaux d'un modèle linéaire. Dans le cas d'un modèle local et étant donné que la fonction noyau est uniforme, la régression sera effectuée grâce à la méthode des least squares en utilisant comme données un nombre fixé de plus proches voisins.

Il existe une méthode récursive pour accélérer l'identification paramétrique d'un modèle constant et local [8]

2.5.3 Validation d'un modèle local

Comme expliqué dans la section 2.3.2, le but de la validation est de tester la capacité de généralisation d'un modèle. Dans le cas de modèles locaux, on recherche l'erreur de généralisation locale. On cherche à ce que le modèle effectue de bonnes prédictions près du query-point. La validation considèrera uniquement les points $x \in D_N$ tels que $d(x, q) < B(k)$, où k est le nombre optimal de voisins.

2.5.4 Algorithme du lazy-learning

A présent voyons comment fonctionne le lazy-learning. Nous sommes en possession d'un ensemble d'apprentissage $D_N = \{z_1, \dots, z_N\}$ composé de N échantillons $z_i = (x_i, y_i)$ où $x_i \in \mathbb{R}^n$ et $y_i \in \mathbb{R}$. L'apprentissage consiste à copier l'ensemble d'apprentissage en mémoire. Supposons qu'on souhaite effectuer une prévision pour le point q . Les étapes mentionnées ci-après décrivent la marche à suivre.

1. Classer les échantillons de D_N selon leur distance par rapport à q

$$d(x(i), q) \leq d(x(j), q) \quad \forall i \leq j \quad (2.28)$$

2. On fait le choix d'utiliser une fonction noyau uniforme et B tel que

$$B(k) = d(x(k), q) \quad (2.29)$$

Pour rappel, B est le seul paramètre structurel à prendre en compte dans la construction d'un modèle local.

3. Le modélisateur du LL impose une valeur pour les quatre bornes k_m^c, k_M^c, k_m^l et k_M^l . Le LL va construire des modèles locaux en se basant sur k voisins ($k \in \{k_m, k_M\}$). Ces modèles locaux sont de deux types : linéaires et constants. Ainsi pour $k_m^c = 5, k_M^c = 10, k_m^l = 6$ et $k_M^l = 16$ LL construira 6 modèles locaux constants et 11 modèles locaux linéaires.
4. L'identification paramétrique est effectuée. Dans le cas d'un modèle constant, LL effectuera une simple moyenne sur la sortie des k plus proches voisins, tandis que dans le cas linéaire, LL fera appel à la procédure des least-squares.
5. On calcule l'erreur de généralisation pour les 17 modèles locaux.
6. La sélection du modèle utilisé pour réaliser la prédiction pour le point q peut se faire selon deux variantes. La première consiste simplement à choisir le modèle

CHAPITRE 2. MÉTHODOLOGIE POUR L'APPRENTISSAGE SUPERVISÉ

local qui présente la plus petite erreur de généralisation. Il s'agit de la technique dite du winner-takes-all. L'autre méthode consiste à choisir un nombre de modèles à combiner entre eux. Expérimentalement, la combinaison de modèles donne de meilleurs résultats.

7. On effectue la prédiction avec le modèle retenu en 6. Toute cette procédure doit être répétée à chaque fois qu'on désire faire une prédiction pour un point q .

Chapitre 3

L'apprentissage supervisé appliqué au domaine de l'anesthésie

Dans la première section de ce chapitre, je parle de concepts, liés au domaine de l'anesthésie, nécessaires à la compréhension de la suite de mon mémoire. Pour commencer, j'introduis le concept d'Anesthésie Intraveineuse à Objectif de Concentration (AIVOC). Il s'agit d'un procédé pendant lequel l'anesthésiste dispose d'un ordinateur pour l'aider à atteindre la profondeur d'anesthésie souhaitée. Ensuite, je présenterai brièvement les trois étapes d'une anesthésie, à savoir l'induction, l'entretien et le réveil du patient ainsi que les objectifs à atteindre pour chacune de ces phases. Plusieurs signaux servent à monitorer la profondeur de l'anesthésie. De tels signaux sont l'*entropie* [23], l'*index bispectral* (BIS) [24, 1] ou encore le Neurosense [25]. Dans le cadre de ce mémoire, je me limiterai à étudier les facteurs qui influencent le signal BIS.

La deuxième section présente le logiciel qui aide les anesthésistes à ajuster les débits des pousse-seringues. Ce logiciel est utilisé à l'hôpital Erasme depuis 2000 et porte le nom d'Infusion TOOLBOX 95 [5] ou en abrégé, Itb95. La suite de la section présente deux techniques basées sur l'apprentissage automatique, construites sur base des données collectées grâce à Itb95, ayant pour but d'aider l'anesthésiste à prendre des décisions lors de l'anesthésie. Ces deux techniques énoncées par O. Caelen, G. Bontempi, et L. Barvais dans [3] et [12] sont duales. La première tente de prédire la profondeur de l'anesthésie à un instant correspondant à un changement de la concentration d'un agent anesthésique, tandis que la deuxième tente de prédire la quantité d'un agent anesthésique à injecter au patient afin que celui-ci atteigne la profondeur d'anesthésie souhaitée par l'anesthésiste.

3.1 Introduction de concepts fondamentaux en anesthésie

Le but de l'anesthésie générale [24] consiste à amener un patient dans un état tel que pendant l'opération il ne ressent pas la gêne engendrée par les stimuli chirurgicaux et qu'il ne se souvienne pas de l'intervention. L'anesthésiste va donc devoir ajuster régulièrement les concentrations d'agents anesthésiques afin d'amener puis de maintenir le patient dans un tel état durant l'entièreté de l'anesthésie.

Il y a différents types d'agents anesthésiques pouvant être groupés dans plusieurs

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

catégories. Celles qui nous intéressent sont : les hypnotiques, les analgésiques et les curares. Ils produisent respectivement comme *effets pharmacologiques* [4] de faire plonger le patient dans un profond sommeil, et d'inhiber la douleur et les réactions motrices.

Les agents utilisés dans le cadre de ce travail sont le Propofol pour les hypnotiques, le Remifentanil pour les analgésiques et le Cisatracurium comme curare. Il est à noter que les curares quant à eux n'ont pas d'influence majeure sur la profondeur de l'anesthésie et n'entrent donc pas en compte dans la suite du travail.

Il se peut donc qu'un patient curarisé et sous hypnose soit plongé dans un sommeil trop léger mais qu'il ne présente extérieurement aucun signe de gêne face aux stimuli (normal puisque les curares bloquent les réactions motrices) alors qu'en réalité il est entrain de vivre un véritable cauchemar puisqu'en l'absence d'analgésique, il ressent toutes les gênes provoquées par l'intervention. Ce type de problème a souvent fait l'objet de plaintes médico-légales. Ce genre de situation est donc à éviter absolument.

Il est important de noter qu'il existe des interactions entre les différents types d'agents anesthésiques. Ainsi on peut diminuer la concentration d'agent analgésique en augmentant la dose d'agent hypnotique, et inversement.

Afin d'aider l'anesthésiste à maintenir le patient dans un état souhaité, une nouvelle technique d'anesthésie a vu le jour à la fin des années septante. Il s'agit de l'Anesthésie Intraveineuse à Objectif de Concentration (AIVOC) [26].

3.1.1 Anesthésie Intraveineuse à Objectif de Concentration

L'objectif qu'on souhaite atteindre avec l'AIVOC est de pouvoir entrer la concentration effective de drogue sans que l'anesthésiste n'ait à se soucier du calcul des débits à injecter pour obtenir la concentration souhaitée à la zone d'effet. Déterminer la concentration plasmatique n'est pas une chose aisée, car il s'agit de résoudre un problème de cinétique multicompartimentale [27, 28]. Dans ce problème de cinétique multicompartimentale on cherche à calculer le débit de la perfusion qu'il faut injecter au patient de manière à assurer une concentration constante dans la zone d'effet. Un tel problème de cinétique multicompartimentale peut être modélisé par le schéma de la figure 3.1.

La dissolution des agents anesthésiques dans l'organisme ne se fait pas instantanément. On va dès lors s'intéresser à deux types de concentrations. La *concentration plasmatique* correspondant au compartiment central est la concentration de drogue présente dans le sang tandis que la *concentration effective* est la concentration de drogue dans la zone d'effet (le cerveau par exemple). L'évolution des concentrations d'agents dans les différentes zones est illustrée par la figure 3.2.

La figure 3.1 illustre le concept de cinétique multicompartimentale. La concentration plasmatique correspond à la concentration de l'agent dans le compartiment central. Les autres compartiments sont dits périphériques. Il est courant de représenter les compartiments périphériques comme étant les muscles et les graisses. Il s'agit toutefois d'une interprétation. Le dernier compartiment est ce que j'ai appelé précédemment la zone d'effet.

Un *bolus* peut être vu comme une courte période pendant laquelle le débit de la perfusion va être élevé. Cela sert à remplir le compartiment central. Après, il conviendra

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

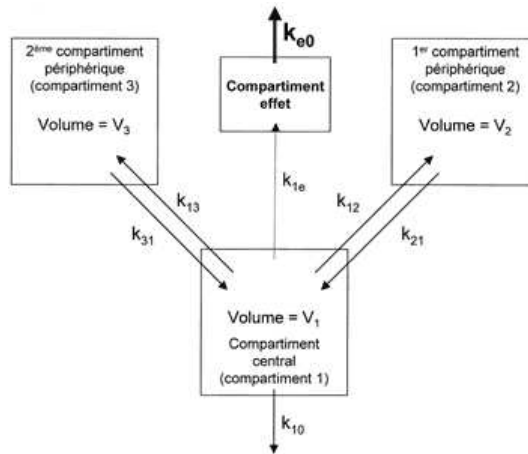


FIG. 3.1: Représentation d'un modèle pharmacocinétique tricompartimental associé à un compartiment effet [27]. Ce modèle est paramétré par les volumes de ses compartiments (V_1 , V_2 , V_3), par ses constantes de transfert (k_{12} , k_{21} , k_{13} , k_{31} , k_{1e}), par sa constante d'élimination (k_{10}) et par k_{e0} .

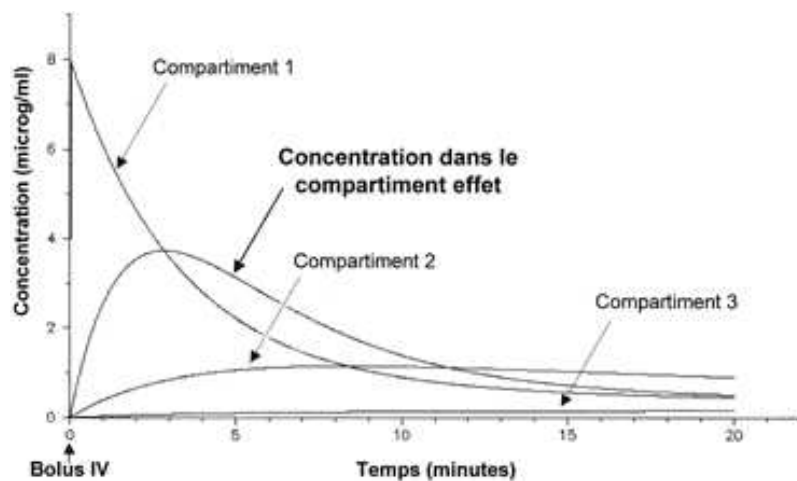


FIG. 3.2: Ce graphique, emprunté à [27], montre l'évolution en fonction du temps des concentrations dans les différents compartiments après injection d'un bolus intraveineux unique de 150 mg de Propofol. Un compartiment effet (ou site d'action) est intégré dans le modèle. C'est la concentration dans ce compartiment qui nous intéresse.

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

de réguler le débit de la perfusion afin de maintenir la concentration au site d'effet. Un exemple de bolus est illustré sur la figure 3.2.

F. Servin présente le concept d'anesthésie intraveineuse à objectif de concentration dans [26] comme ceci :

En anesthésie, l'effet pharmacologique d'un agent est fonction de la concentration de cet agent en face des récepteurs impliqués (la zone d'effet). Par ailleurs, l'effet est constant si la concentration est constante, et ce quelle que soit la durée pendant laquelle cette concentration est maintenue [...]. Ainsi, chercher à obtenir un effet revient à obtenir une concentration, et la connaissance de la zone de concentrations efficaces d'un agent chez un patient donné, comme on peut l'acquérir à l'induction de l'anesthésie, permet de conduire de façon plus précise l'ensemble de l'administration de l'agent. Dans cette optique, l'administration en débit massique n'est qu'un pis-aller.

Avec les techniques habituelles, obtenir instantanément une concentration souhaitée, donc un effet recherché est impossible du fait de la cinétique complexe, multicompartimentale, des agents intraveineux. Par définition, les injections IV directes en bolus provoquent un pic instantané de concentration plasmatique. La concentration diminue ensuite constamment jusqu'au bolus suivant. Pour pallier l'effet oscillant, lié aux bolus itératifs, l'entretien de l'anesthésie fait souvent appel à une perfusion continue. Malheureusement, si le débit de perfusion est constant, comme le transfert vers les compartiments périphériques diminue au cours du temps, puisqu'il dépend de la différence de concentration entre le sang et la périphérie, la concentration résultante augmente progressivement. Pour obtenir une concentration recherchée, il faut d'abord administrer un bolus calibré, puis prendre instantanément le relais par une perfusion à débit progressivement décroissant. Le seul moyen d'y parvenir, qui présente par ailleurs au quotidien l'avantage de rendre invisible l'étape de calcul pharmacocinétique, est de calculer précisément la dose pour obtenir la concentration à l'aide d'un microprocesseur.

L'anesthésie intraveineuse à objectif de concentration (AIVOC) représente ce mode d'administration où l'anesthésiste recherche à atteindre une concentration (un effet), à charge à un programme de calcul validé de pourvoir au débit nécessaire pour l'obtenir.

Le principe de l'AIVOC peut donc se résumer au scénario suivant : l'anesthésiste va décrire une concentration cible à l'ordinateur afin d'atteindre la profondeur de l'anesthésie souhaitée. L'ordinateur va transformer cette concentration cible en un débit de perfusion et va commander les pousse-seringues.

On distingue deux types d'équipements permettant de réaliser des AIVOC. Les systèmes à éléments intégrés et les systèmes à éléments séparés. Pour les systèmes à éléments séparés, on dispose d'un ordinateur relié à un appareil de perfusion. Sur cet ordinateur s'exécute un logiciel qui transforme les concentrations suggérées par l'anesthésiste en débits de perfusion et commande les pousse-seringues. Ce type d'équipement est bien adapté pour la recherche. A l'hôpital Erasme, le logiciel utilisé est Itb95 [5]. J'explique brièvement son fonctionnement dans la section 3.2.1.

3.1.2 Déroulement de l'anesthésie

Au cours d'une AIVOC, la continuité entre l'induction et l'entretien de l'anesthésie est acquise grâce à la notion de concentration cible. Cette section a comme objectif d'introduire les différentes phases de l'anesthésie [26] ainsi que les objectifs à atteindre dans chacune d'elles.

Induction

Il s'agit de la première phase de l'anesthésie. Pendant cette phase, l'anesthésiste va choisir la concentration cible de Propofol à atteindre de sorte que le patient perde conscience. Deux cas de base interviennent.

Si le logiciel permet de calculer les concentrations plasmatiques, l'anesthésiste cible la concentration plasmatique dont il sait qu'elle induira une perte de conscience chez 95% des patients et attend le délai pendant lequel le transfert du sang vers la zone d'effet va se produire.

Dans l'autre cas, si le logiciel permet de cibler les concentrations effectives, il suffit à l'anesthésiste de prescrire la concentration dans la zone d'effet qui correspond à la perte de conscience chez 95% des patients.

Entretien

Durant cette phase l'anesthésiste va tenter de maintenir une concentration efficace adaptée aux stimuli chirurgicaux mais aussi en tenant compte de l'état du patient. Les concentrations adéquates varient d'un patient à un autre c'est pourquoi, il conviendra d'adapter les concentrations en fonction de l'individu.

Réveil

A partir du moment où l'on a arrêté la perfusion, il va falloir attendre un certain moment avant que les agents anesthésiques cessent d'agir dans les zones d'effet. Cette durée est liée au temps de décroissance (decrement time) à l'arrêt de la perfusion [26] qui caractérise le devenir de l'agent à l'arrêt de la perfusion. Ce paramètre est le temps mis, à l'arrêt de la perfusion, pour atteindre une concentration choisie qui est celle pour laquelle la fin de l'effet est attendue. Le calcul de ce temps de décroissance fait lui aussi intervenir la pharmacocinétique [27] et la pharmacodynamique¹ [4].

3.1.3 Monitoring de la profondeur de l'anesthésie

Jusqu'ici j'ai insisté sur le fait qu'il était important de surveiller la profondeur de l'anesthésie, mais je n'ai pas encore parlé des signaux servant à mesurer ce phénomène. En fait, mesurer la profondeur de l'anesthésie revient à évaluer la relation entre les doses de drogues injectées au patient et les effets que celles-ci ont sur ce dernier. *L'index bispectral*

¹La pharmacodynamique décrit l'impact d'un médicament sur l'organisme.

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

(BIS) [1, 24] illustré sur la figure 3.4 est une valeur utilisée afin de mesurer la profondeur de l'anesthésie. Le BIS est une découverte assez récente et son utilisation n'est pas encore très répandue à travers le monde. Le calcul du BIS repose sur la décomposition de Fourier du signal de l'électro-encéphalogramme. Plus d'informations à ce sujet sont disponibles dans l'article [1]. Il s'agit typiquement d'un scalaire compris entre 0 et 100 qui correspondent respectivement à un électro-encéphalogramme plat et à un patient éveillé.

De nombreux facteurs influencent la valeur du BIS. Il existe en effet une corrélation entre la concentration de Propofol/Remifentanil et le niveau du BIS. C'est ce que montrent D. Longrois, M. Hirshi, E. Junke et C. Meistelman dans [24] en décrivant notamment les effets des hypnotiques et/ou des morphiniques sur le BIS. C'est ainsi qu'on y apprend que les valeurs du BIS diminuent proportionnellement avec l'augmentation des concentrations de la majorité des hypnotiques tels que le Propofol.

Comme je l'avais expliqué avant, le but de l'anesthésie est de permettre au patient de subir des interventions douloureuses ou désagréables sans en ressentir les effets. Afin d'atteindre cet objectif, l'anesthésiste va essayer de stabiliser le BIS dans un intervalle de valeurs comprises entre 40 et 55 (ces chiffres ont été obtenus sur des patients uniquement sous hypnose) [1].

Pour conclure, le BIS permet de mesurer principalement les effets hypnotiques. La douleur quant à elle, relative aux analgésiques peut être monitorée grâce à des signes physiologiques tels que le rythme cardiaque ou la pression artérielle [24].

3.2 Apprentissage automatique appliqué à l'AIVOC

Depuis plusieurs années, des recherches ont été entreprises dans le domaine de l'anesthésie afin de fournir des outils permettant d'aider l'anesthésiste dans son travail. Le contrôle de la profondeur de l'anesthésie fait partie de ces domaines de recherche, dont le concept est de déterminer les concentrations de drogues à injecter au patient afin que le niveau de BIS de ce dernier soit maintenu dans une zone cible. On peut, parmi ces travaux, citer [9, 10, 11]. Une autre approche utilisant des techniques d'apprentissage supervisé, toujours afin de contrôler la profondeur de l'anesthésie, a été étudiée par des chercheurs du MLG dans [12]. Cette approche fera l'objet d'une étude plus détaillée dans la suite de mon mémoire.

Un autre domaine de recherche original est l'utilisation de techniques d'apprentissage supervisé afin de prédire l'évolution du BIS d'un patient après que l'anesthésiste ait modifié la concentration de drogues à injecter au patient. Ce projet est financé par la Région Wallonne dans le cadre du projet *First Europe Objectif 1*. Cette méthode décrite dans [3] fera elle aussi l'objet d'une étude approfondie dans la suite de mon mémoire.

3.2.1 Infusion TOOLBOX 95

Itb95 est un logiciel qui assiste l'anesthésiste dans l'ajustement des débits de drogues injectées au patient. Dans notre cas d'étude, nous nous intéresserons uniquement au pilotage des injecteurs de Propofol comme hypnotiques et de Remifentanil comme analgésiques.

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

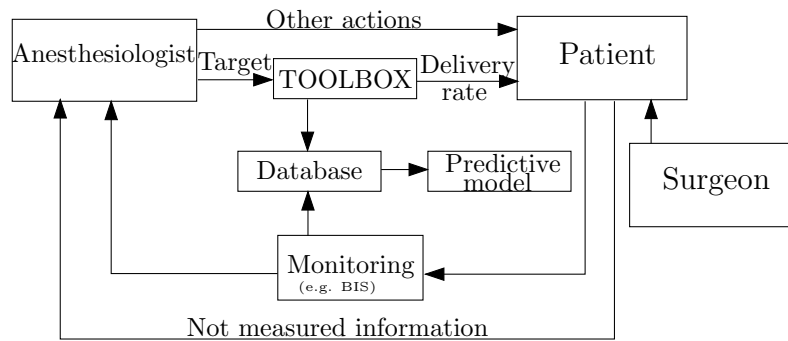


FIG. 3.3: Cette figure issue de [3] montre à quel endroit se situe Infusion Toolbox 95 dans le processus de l'anesthésie. Toolbox réalise plusieurs tâches : (i) il contrôle les débits des perfusions en fonction de la concentration cible entrée par l'anesthésiste et (ii) il monitore et enregistre tous les signaux du patient ainsi que les actions réalisées par l'anesthésiste.



FIG. 3.4: Ces trois illustrations montrent le moniteur de BIS en action. C'est un dispositif propriétaire fabriqué par la société Aspect. Le graphique de l'évolution du BIS montre qu'on est dans la phase de maintien de l'anesthésie et que la valeur du BIS courant est de 58. L'induction a déjà été réalisée car on assiste à une chute brutale des valeurs du BIS dans les premiers instants de monitoring. Les signaux sont collectés grâce à un stick frontal relié au dispositif de monitoring.

Itb95 ajuste le débit de Propofol selon le modèle pharmacocinétique de Schnider [27] et le Remifentanyl selon le modèle pharmacocinétique de Minto [29]. Afin de communiquer avec les appareils médicaux, Itb95 est relié aux appareils grâce à des câbles série. Dans notre cas, ils sont connectés à un injecteur de Propofol, un injecteur de Remifentanyl et le moniteur de Bis.

Le signal de BIS est mesuré en fonction d'un indice de qualité car des stimuli chirurgicaux tels que l'utilisation du bistouri électrique induisent des perturbations sur le signal. Dans ce cas de figure, les valeurs de BIS ne sont plus fiables et Itb95 n'en tiendra pas compte.

Dans la section 3.1.1 j'introduisais l'Anesthésie à Objectif de Concentration comme étant un procédé d'anesthésie générale où l'anesthésiste décrit une concentration de drogue à atteindre dans un compartiment cible. A charge pour l'ordinateur de commander les injecteurs selon un débit de perfusion adéquat. C'est exactement le rôle que remplit Itb95. La figure 3.5 montre Itb95 en cours de fonctionnement.

Le choix de l'utilisation d'Itb95 à l'hôpital Erasme provient du fait que des injecteurs embarquant les modèles de calculs cités précédemment sont excessivement chers et sur-

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

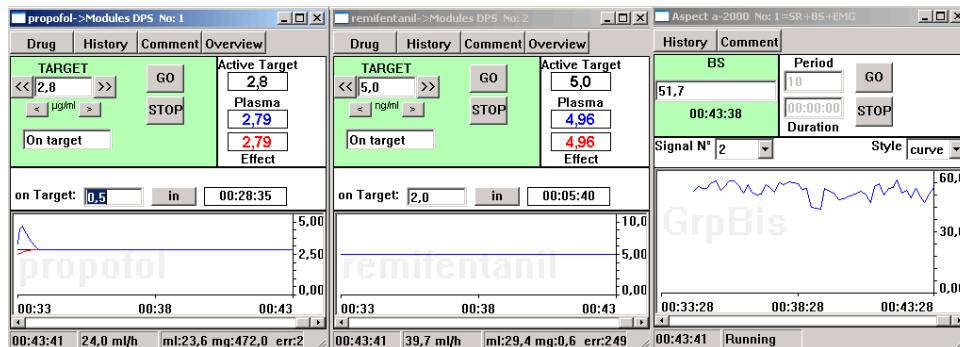


FIG. 3.5: Ce graphique illustre le fonctionnement d'Itb95. Les deux fenêtres de gauche servent respectivement à contrôler les injecteurs de Propofol et de Remifentanyl. La courbe rouge représente la concentration plasmatique tandis que la bleue représente la concentration effective (dans la zone d'effet). Pour une concentration cible de 2.8 µg/ml de Propofol, Itb95 assure un débit de 24 ml/h. La fenêtre de droite représente la courbe de BIS. C'est une copie des informations transmises par le moniteur de BIS.

tout, ne sont pas utilisables pour la recherche scientifique. Ainsi grâce à Itb95 une large base de données créée au travers de plus de 1000 anesthésies a pu être construite afin d'être exploitée dans plusieurs projets de recherche à l'ULB.

3.2.2 Prédiction du BIS

Les résultats présentés dans cette section sont issus de l'article [3]. Il a pour but d'appliquer des techniques de data mining et d'apprentissage automatique dans le domaine de l'anesthésie. L'objectif recherché est la prédiction du BIS à un certain temps après un changement de la concentration cible de Propofol effectué par l'anesthésiste.

Pour commencer, il a fallu trouver un intervalle de temps optimal Δt^* maximisant la corrélation entre le changement de la concentration cible de Propofol et la variation de la profondeur de l'anesthésie. Ensuite, sur base de cet intervalle optimal, plusieurs modèles ont été construits afin de prédire le niveau du BIS à l'instant $t + \Delta t^*$ où t est l'instant correspondant au moment où l'anesthésiste change la concentration de Propofol. Pour terminer, la validation des modèles a été effectuée selon trois critères bien précis.

Estimation du délai optimal

Cette contribution a pour but de trouver l'intervalle de temps optimal qui maximise la corrélation entre le changement de la concentration cible de Propofol et son impact sur la valeur du BIS.

Pour ce faire, 589 échantillons ont été créés en respectant la condition suivante : si la concentration cible de Propofol a été changée à l'instant t , aucun autre changement de la concentration cible de Propofol n'a été effectué dans l'intervalle de temps $[t - 600 \text{ sec}, t + 600 \text{ sec}]$.

Soit $\Delta B(\Delta t) = B(t + \Delta t) - B(t)$ c'est-à-dire la différence entre le niveau de BIS à l'instant $t + \Delta t$ et celui à l'instant t . $B(t)$ est calculé en effectuant la moyenne arithmétique des

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

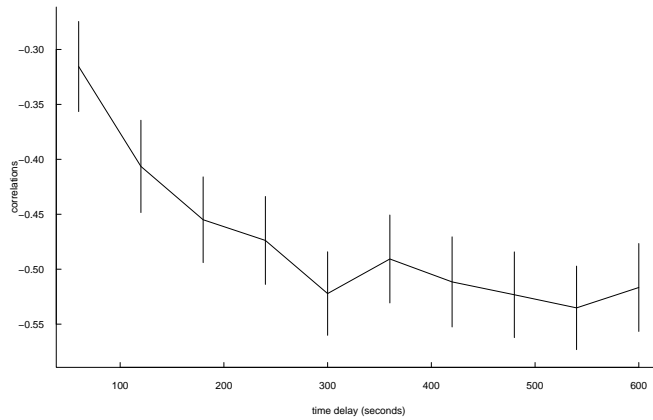


FIG. 3.6: Cette figure issue de [3] illustre la corrélation entre le changement de la concentration cible de Propofol et la variation de BIS observée cela en fonction du temps. La corrélation augmente en valeur absolue dans l'intervalle de temps $[0, 300]$ puis se stabilise dans l'intervalle $[300, 800]$. Le graphique montre que la corrélation maximale est atteinte après 300 secondes. Le signe de corrélation est négatif, car lorsqu'on augmente la concentration de Propofol, le niveau du BIS chute.

niveaux de BIS dans l'intervalle $[t - 30, t]$. Tandis que $B(t + \Delta t)$ est calculé en prenant la moyenne des niveaux de BIS sur l'intervalle $[t + \Delta t - 30, t + \Delta t + 30]$. ΔP quant à lui est égal à la variation de la concentration cible de Propofol entre l'instant t et la concentration précédente au changement de Propofol effectué à ce même instant t .

Les corrélations entre $\Delta B(\Delta t)$ et ΔP ont été calculées lorsque Δt prend des valeurs dans $\{60, 120, 180, \dots, 540, 600\}$. Pour analyser les résultats, il faut comprendre que l'augmentation de la concentration cible de Propofol va entraîner une chute du niveau de BIS. C'est la raison pour laquelle les corrélations entre $\Delta B(\Delta t)$ et ΔP sont négatives. La figure 3.6 montre que l'intervalle de temps optimal est $\Delta t^* = 300$ secondes.

Construction des modèles

La figure 3.7 montre les variables observées et collectées via Itb95. Sur base de ces données, le but est de construire un modèle de prédiction du BIS de sorte à renseigner l'anesthésiste sur l'évolution possible de l'état du patient à un instant précis, Δt^* secondes après que le praticien ait changé la concentration de Propofol à l'instant t . Le phénomène peut être décrit par le processus stochastique suivant :

$$B(t + \Delta t^*) = f(B(t), tpo(t), tpn(t), \Delta timeP, tr(t), a, w, h, s, lbm) + w(t) \quad (3.1)$$

Cela signifie qu'à l'instant t , $B(t)$ est la valeur du BIS courant, $tpo(t)$ est la valeur de l'ancienne concentration cible de Propofol, $tpn(t)$ est la nouvelle concentration cible de Propofol, $\Delta timeP$ est le temps écoulé entre la dernière modification de la concentration cible de Propofol et l'instant t et $tr(t)$ est la concentration de Remifentanil. a, w, h, s et lbm correspondent respectivement à l'âge, au poids, à la taille, au sexe et au poids maigre

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

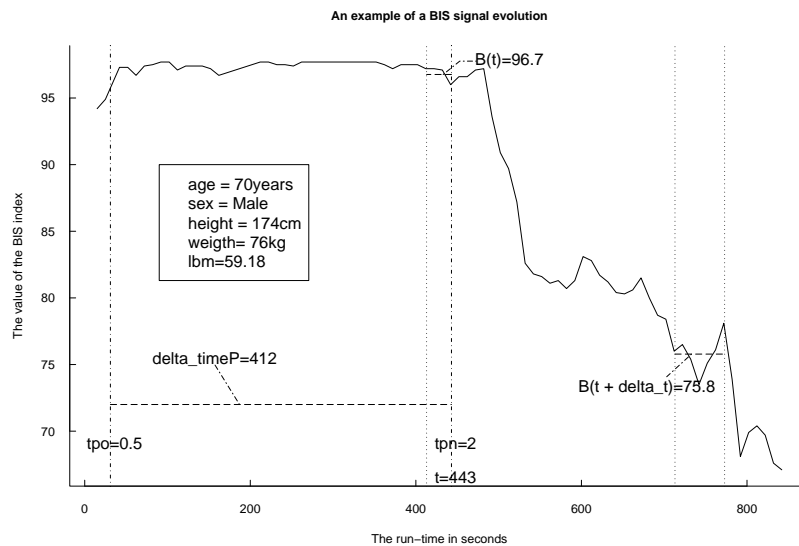


FIG. 3.7: Ce schéma issu de [3] illustre le contexte relatif à une prédiction. A l'instant $t = 443$ secondes, l'anesthésiste cible la concentration de Propofol à $2 \mu\text{g/ml}$, tandis que l'ancienne valeur $tpo = 0.5$. Le temps qui s'est écoulé entre ces deux changements de concentration, ΔtimeP vaut quant à lui 412. A l'instant t le BIS vaut $B(t) = 96.7$. Les autres variables sont statiques car elles sont propres au patient, il s'agit de l'âge, du sexe, de la taille, du poids maigre (lbm). $B(t + \Delta t)$ est la valeur du BIS mesurée à l'instant $t + \Delta t$. C'est cette valeur que les modèles vont tenter de prédire en fonctions des variables d'entrée au préalable calculées.

(Lean Body Mass)² qui est interprété comme la masse totale d'un individu de laquelle on a soustrait la masse grasseuse. Δt^* est le délai maximisant la corrélation entre la variation de Propofol et la variation du BIS expliqué dans la section 3.2.2. Quant à $w(t)$, il s'agit du bruit.

Une solution pour résoudre le problème de la prédiction du BIS futur pour un changement de Propofol donné est de réaliser une identification paramétrique (voir section 2.4) pour un type de modèle donné. C'est ce qui a été réalisé ici. De manière générale, le modèle est décrit par

$$\widehat{B}(t + \Delta t^*) = h(\{B(t), tpo(t), tpn(t), \Delta timeP, tr(t), a, w, h, s, lbm\}, \alpha_N) \quad (3.2)$$

où α_N est le vecteur résultant de l'identification paramétrique.

Un ensemble d'apprentissage D_N a été créé sur base de $N = 1702$ échantillons $\langle B_i(t + \Delta t^*), q_i(t) \rangle$, $i = 1, \dots, N$ générés par Itb95. Un échantillon q_i est un vecteur contenant toutes les variables d'entrée $(B(t), tpn(t), tpo(t), \Delta timeP, tr(t), a, w, h, s, lbm)$, et appartient à l'ensemble d'apprentissage si lorsqu'un changement de la concentration cible de Propofol se produit à l'instant t aucun autre changement de la concentration cible de Propofol n'est observé dans l'intervalle de temps $[t, t + \Delta t^*]$.

Deux types de modèles différents ont été implémentés afin de résoudre le problème de prédiction de $\widehat{B}(t + \Delta t^*)$ lorsqu'on est en possession de $q(t)$. La première approche réside dans l'utilisation d'un modèle linéaire sur l'ensemble du domaine. Cela a été résolu grâce à la méthode des least-squares décrite dans l'annexe A.

$$\widehat{B}(t + \Delta t^*) = \alpha_N^T x(t) \quad (3.3)$$

où α_N est le vecteur de paramètres estimé par la méthode des least-squares sur base de l'ensemble d'apprentissage D_N et où $x(t) = [1, B(t), tpo(t), tpn(t), \Delta timeP, tr(t), a, w, h, s, lbm]^T$. La deuxième approche a été d'utiliser le lazy-learning (voir section 2.5).

Sélection de variables

Dans la section 2.3.3 j'ai introduit plusieurs techniques afin de réduire la dimensionnalité d'un problème. La solution choisie ici est la technique de forward-selection validée grâce à la technique de leave-one-out cross-validation décrite dans la section 2.3.2.

Les figures 3.8 et 3.9 illustrent respectivement le résultat de la sélection de variables pour le modèle linéaire et pour le lazy-learning. On remarquera que le BIS, le tpn et le tpo sont les trois variables qui contiennent le plus d'informations.

Validation

Dans cette section les deux modèles à valider utilisent uniquement les trois variables les plus significatives choisies par la procédure de sélection de variables. Pour rappel, ces variables sont le BIS à l'instant t , la concentration cible de Propofol à ce même instant ainsi que l'ancienne valeur de la concentration cible de Propofol.

²<http://www.medterms.com/script/main/art.asp?articlekey=25887>

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

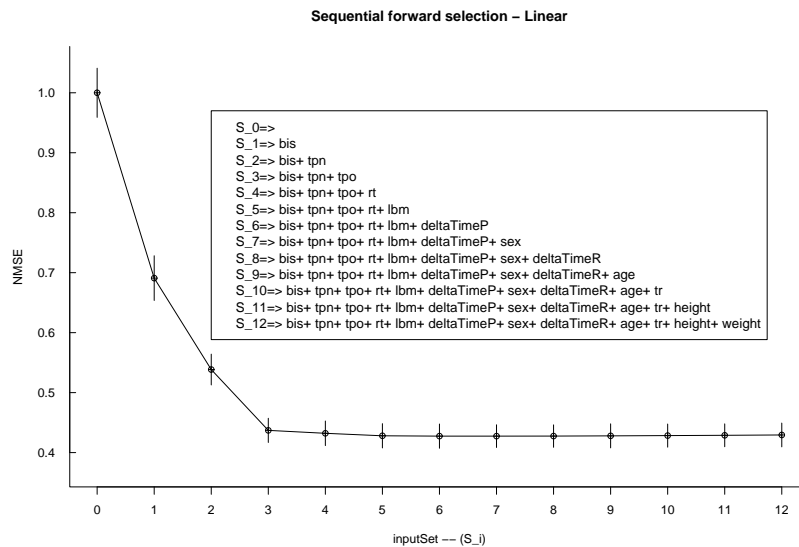


FIG. 3.8: Illustration tirée de [3] montrant le résultat de la procédure de forward sélection sur l'ensemble des variables d'entrée du modèle linéaire. Les S_i représentent la combinaison de variables qui, prises ensemble, minimisent l'erreur de généralisation. Le modélisateur a estimé que S_3 contenait assez d'informations et a donc été retenu comme étant la meilleure structure.

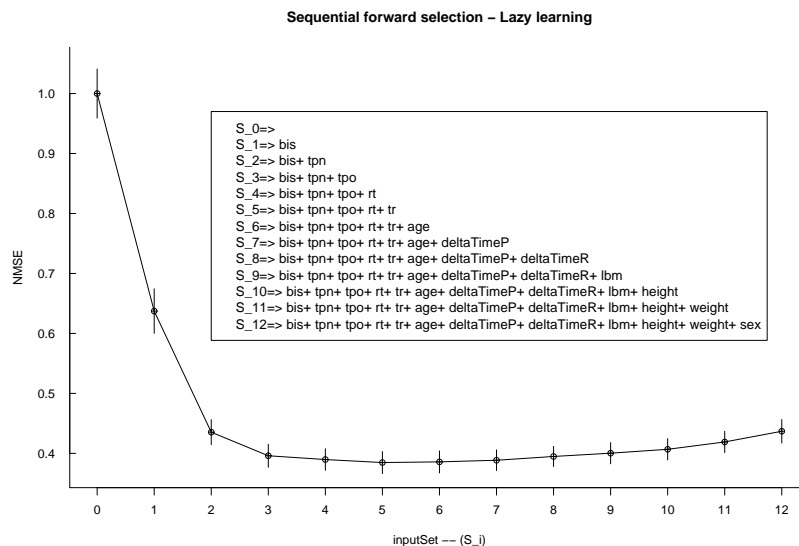


FIG. 3.9: Illustration tirée de [3] montrant le résultat de la procédure de forward sélection sur l'ensemble des variables d'entrée du modèle Lazy-learning. Les S_i représentent la combinaison de variables qui prises ensemble minimisent l'erreur de généralisation. Ici aussi, le modélisateur a estimé que S_3 contenait assez d'informations et a donc été retenu comme étant la meilleure structure.

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

La validation de ces deux modèles a été effectuée en utilisant la méthode de leave-one-out décrite dans la section 2.3.2. L'estimation de l'erreur de généralisation par la méthode l-o-o cv pour l'échantillon i est définie comme ceci :

$$\widehat{E}_i^{loo} = \widehat{B}_{(-i)}(t + \Delta t^*) - B_i(t + \Delta t^*) \quad (3.4)$$

où $\widehat{B}_{(-i)}(t + \Delta t^*)$ est la prédiction pour l'échantillon i renvoyée par le modèle qui s'est entraîné sur tous les échantillons de l'ensemble d'apprentissage hormis le $i^{\text{ème}}$ et $B_i(t + \Delta t^*)$ est la valeur réelle du BIS à l'instant $t + \Delta t^*$ enregistrée dans l'ensemble d'apprentissage pour l'échantillon i .

Plusieurs critères ont été évalués afin de sélectionner la meilleure structure. Le premier est l'erreur quadratique moyenne normalisée (NMSE) définie par

$$NMSE = \frac{\sum_{i=1}^N (\widehat{E}_i^{loo})^2}{\sum_{i=1}^N (\hat{\mu}_b - B_i(t + \Delta t^*))^2} \quad (3.5)$$

où $\hat{\mu}_b = \frac{1}{N} \sum_{i=1}^N B_i(t + \Delta t^*)$ est la moyenne arithmétique des BIS futurs stockés dans l'ensemble d'apprentissage. Normaliser l'erreur quadratique moyenne permet de comparer la qualité du modèle par rapport à l'estimateur moyenne arithmétique. C'est à dire que si $NMSE < 1$ alors l'erreur commise par le modèle est moindre que si on avait utilisé la moyenne arithmétique comme estimation du BIS futur.

Le deuxième critère utilisé dans l'article est le MAE ou *Mean Absolute Error* qui est défini par

$$MAE = \frac{1}{N} \sum_{i=1}^N |\widehat{E}_i^{loo}| \quad (3.6)$$

Cette mesure a l'avantage de quantifier l'erreur commise par le modèle dans la même unité de mesure que le BIS. Elle nous donne des indications sur la moyenne de l'amplitude de l'erreur commise par le modèle.

Enfin, le dernier critère est le pourcentage (P) de cas pour lesquels la variation du BIS futur prédit par le modèle a le même signe que la variation du BIS réel dans l'ensemble d'apprentissage. P est défini par

$$P = \frac{100}{N} \sum_{i=1}^N I \left[(\widehat{B}_{(-i)}(t + \Delta t^*) - B_i(t)) \cdot (B_i(t + \Delta t^*) - B_i(t)) \right] \quad (3.7)$$

où

$$I[A] = \begin{cases} 1 & \text{si } A \geq 0, \\ 0 & \text{sinon} \end{cases}$$

Par exemple : si $B_i(t) = 45$, $B_i(t + \Delta t^*) = 50$ et $\widehat{B}_{(-i)}(t + \Delta t^*) = 36$ on a que $I[]$ vaut 0 car le modèle a prédit que le BIS allait descendre alors qu'en réalité il est remonté.

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

Modèle	NMSE	MAE	P
$\Lambda_{S_3}^{lin}$	0.437	6.35	81.8
$\Lambda_{S_3}^{lazy}$	0.395	5.91	83.1

Tab. 3.1: Tableau reprenant le résultat des critères de validation pour le modèle Linéaire $\Lambda_{S_3}^{lin}$ avec le set de variables S_3 (voir figure 3.8) et pour le modèle Lazy-Learning $\Lambda_{S_3}^{lazy}$ avec lui aussi le set de variables S_3 (voir figure 3.9)

Conclusions

Soit les structures $\Lambda_{S_3}^{lin}$ et $\Lambda_{S_3}^{lazy}$ correspondant respectivement aux modèles linéaires et Lazy où dans les deux cas l'espace des variables d'entrée est composé des trois variables les plus significatives d'après la forward-selection, à savoir le BIS à l'instant t , le $tpn(t)$ et $tpo(t)$. Le tableau 3.1 est un récapitulatif des résultats du calcul des trois critères de validation. On remarquera ainsi que $\Lambda_{S_3}^{lazy}$ présente les meilleures performances pour tous les critères. Le fait que le modèle Lazy soit significativement meilleur est un argument en faveur du fait que la relation entre le $tpn(t)$ et la valeur du BIS à ce même instant n'est pas linéaire [3].

3.2.3 Prédiction de la variation de Propofol

L'approche présentée dans l'article [12] est duale à celle énoncée dans [3]. Elle ne consiste plus à prédire la profondeur de l'anesthésie, mais à prédire la variation de Propofol nécessaire afin d'amener la profondeur de l'anesthésie à un niveau souhaité.

Le processus stochastique est le même que celui qui a généré les données de la section précédente, à savoir

$$B(t + \Delta t^*) = f(B(t), p(t), \delta p(t), r(t), a, w) + w(t) \quad (3.8)$$

On remarquera qu'on a transformé $tpo(t)$ en $\delta p(t)$ car ici on s'intéresse non plus aux valeurs du BIS mais à la variation de la concentration cible de Propofol nécessaire afin d'amener le BIS au niveau désiré. En supposant qu'on puisse exprimer $\delta p(t)$ en fonction des variables d'entrée, on réécrit le processus stochastique comme ceci :

$$\delta p(t) = f(b(t), b(t + \Delta t^*), p(t), r(t), a, w) + w(t) \quad (3.9)$$

Deux méthodes ont été construites et analysées. La première consiste à construire un modèle de prédiction pour la variation de la concentration cible de Propofol et ce en fonction du niveau de BIS à atteindre. Cette méthode porte aussi le nom d'*inverse control* [30]. L'équation générale du modèle de prédiction utilisé est la suivante :

$$u_I(t) = h_I(\{b(t), b^*, p(t), r(t), a, w\}, \alpha_N) \quad (3.10)$$

où b^* est la valeur du BIS qu'on souhaite atteindre et α_N est le vecteur de paramètres renvoyé par la procédure d'identification paramétrique (voir section 2.4). Deux modèles

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

seront construits selon cette équation : un linéaire et un se basant sur les techniques de Lazy-Learning introduites dans la section 2.5.

L'autre technique porte le nom de *forward control* [30]. Il s'agit d'un problème d'optimisation reposant sur un modèle de prédiction du BIS en fonction de la variation de Propofol. On va ainsi proposer au modèle plusieurs valeurs du $\delta p(t)$ proches des valeurs cibles et le problème d'optimisation renverra la meilleure valeur. A nouveau la méthode sera testée selon les résultats fournis par deux types de modèles : un linéaire et un autre de type Lazy-Learning. L'équation générale du modèle utilisé ici est la suivante :

$$\hat{B}(t + \Delta t^*) = h_d(\{b(t), p(t), \delta p(t), r(t), a, w\}, \alpha_N) \quad (3.11)$$

où α_N est le vecteur de paramètres retournés lors de l'identification paramétrique (voir section 2.4).

Soit b^* la valeur du BIS à atteindre et δp la variable à prédire. Une fois l'identification paramétrique effectuée sur le modèle décrit par la formule (3.11) et que nous disposons d'un ensemble U comprenant des valeurs alternatives imposées pour la variation de Propofol, le processus d'optimisation à résoudre est le suivant.

$$\delta p(t) = \arg \min_{u \in U} |h_d(\{b(t), p(t), u, r(t), a, w\}, \alpha_N) - b^*| \quad (3.12)$$

La valeur renvoyée est la variation de Propofol qui a généré la prédiction de BIS la plus proche de la valeur b^* du BIS souhaitée.

Validation

On dispose donc de quatre méthodes :

- Λ_{inv}^{Lin} correspondant au contrôleur inverse qui utilise un modèle linéaire,
- Λ_{inv}^{Lazy} correspondant au contrôleur inverse qui utilise un modèle Lazy,
- Λ_{for}^{Lin} correspondant au contrôleur forward qui utilise un modèle linéaire,
- Λ_{for}^{Lazy} correspondant au contrôleur forward qui utilise un modèle Lazy.

Les quatre méthodes ont été validées selon les mêmes critères que dans l'article [3]. Remis dans le contexte de cet article, voici comment les calculer.

La validation de ces méthodes a été effectuée en utilisant la méthode de leave-one-out cross-validation décrite dans la section 2.3.2. L'estimation de l'erreur de généralisation par la méthode l-o-o pour l'échantillon i est définie comme ceci

$$\widehat{E}_i^{loo}(t_i) = \delta p(t_i) - u^{(-i)}(t_i) \quad (3.13)$$

où $u^{(-i)}$ est la valeur retournée par le modèle qui s'est entraîné sur toutes les données à l'exception du i ème échantillon. $\delta p(t)$ est la modification de Propofol effectuée par l'anesthésiste enregistrée dans l'ensemble d'apprentissage.

Le NMSE servait d'estimation pour l'estimation de l'erreur de généralisation. Il est calculé comme ceci :

CHAPITRE 3. L'APPRENTISSAGE SUPERVISÉ APPLIQUÉ AU DOMAINE DE L'ANESTHÉSIE

$$NMSE = \frac{\sum_{i=1}^N (\widehat{E}_i^{loo}(t_i))^2}{\sum_{i=1}^N (\hat{\mu}_p - \delta p(t_i))^2} \quad (3.14)$$

où $\hat{\mu}_p = \frac{1}{N} \sum_{i=1}^N \delta p(t_i)$. A nouveau on utilise cet estimateur afin de comparer la précision du modèle par rapport à l'estimateur moyenne arithmétique défini par $\hat{\mu}_p$.

Le second critère était le MAE. Il a l'avantage d'être exprimé dans la même unité de mesure que δp et nous donne la valeur moyenne des erreurs commises par le modèle. Il est calculé comme ceci :

$$MAE = \frac{1}{N} \sum_{i=1}^N |\widehat{E}_i^{loo}(t_i)| \quad (3.15)$$

Le dernier critère était P , qui exprime le pourcentage de fois que le modèle a prédit une variation de Propofol ayant le même signe que la modification effectuée par l'anesthésiste lui-même. L'équation suivante montre comment calculer le pourcentage.

$$P = \frac{100}{N} \sum_{i=1}^N I[\delta p(t_i) \cdot u^{(-i)}] \quad (3.16)$$

où

$$I[A] = \begin{cases} 1 & \text{si } A \geq 0, \\ 0 & \text{sinon} \end{cases}$$

et où $u^{(-i)}$ est la prédiction de la variation de Propofol calculée par la méthode leave-one-out. $\delta p(t_i)$ est quant à lui le changement de Propofol qu'a réalisé l'anesthésiste.

Les résultats obtenus pour le contrôleur inverse sont présentés dans le tableau 3.2 tandis que ceux du contrôleur inverse/forward sont présentés dans le tableau 3.3.

Modèle	NMSE	MAE	P
$\Lambda_{S_3}^{lin}$	0.67	0.28	75.7
$\Lambda_{S_3}^{lazy}$	0.65	0.26	75.7

TAB. 3.2: Contrôleur inverse

Modèle	NMSE	MAE	P
$\Lambda_{S_3}^{lin}$	0.70	0.30	72.43
$\Lambda_{S_3}^{lazy}$	0.67	0.28	72.1

TAB. 3.3: Contrôleur forward

Conclusions

Les résultats obtenus dans le cas du contrôleur inverse indiquent que le *NMSE* est significativement inférieur à un et que le signe de la variation de la concentration cible de Propofol sera dans 75 % des cas, identique à celui choisi par l'anesthésiste. Cependant, le tableau 3.3 nous indique que la technique forward n'apporte pas de précision supplémentaire à la prédiction. On remarque aussi que dans les deux cas l'utilisation de la méthode lazy-learning améliore la précision des prédictions.

3.3 Conclusion

Jusqu'ici j'ai introduit des concepts relatifs au domaine de l'apprentissage automatique et de ses différentes applications dans le milieu de l'anesthésie. Mon apport personnel pour ce mémoire se base lui aussi sur tous ces concepts. A partir des résultats énoncés dans la section [3] de nouveaux modèles ont été développés par Olivier Caelen, sur des intervalles de temps plus courts. C'est ainsi qu'on pourra prévoir l'évolution du BIS du patient toutes les 30 secondes dans l'intervalle de temps $[t, t + 600]$ où t est comme précédemment, l'instant correspondant à un changement de la concentration cible de Propofol. De plus en ajoutant un ensemble de variables d'entrée additionnelles construites lors de l'induction de l'anesthésie on espère que les modèles gagneront en précision.

Le développement de ces modèles est étudié dans les chapitres suivants. L'approche de leur construction et de leur validation est similaire à la démarche entreprise dans l'article 3.2.2. En plus du développement de ces modèles, un outil d'aide à la décision a été implémenté afin d'être utilisé par les anesthésistes en conditions réelles lors des anesthésies.

Chapitre 4

Améliorations des modèles existants

En partant des travaux réalisés dans le cadre de l'article [3] détaillés dans le chapitre précédent, j'ai eu l'occasion de développer et de tester de nouveaux modèles de prédiction du BIS. Ce chapitre a pour but de montrer les modifications apportées aux modèles existants et de comparer les performances des différents modèles de manière théorique selon plusieurs critères.

4.1 Présentation des modèles existants

Dans le chapitre précédent j'ai introduit deux modèles d'apprentissage permettant de prédire le BIS cinq minutes après le dernier changement de la concentration cible de Propofol. Ces deux modèles diffèrent de par leur structure où l'un se base sur les techniques de régressions linéaires tandis que le second utilise la technique Lazy-Learning.

Chaque échantillon de l'ensemble d'apprentissage a été créé à l'instant correspondant à une modification de la concentration de Propofol. Les variables d'entrée enregistrées dans chaque échantillon sont :

- *bis* : la valeur du BIS à l'instant de la prédiction
- *tpn* : la valeur de la nouvelle concentration cible de Propofol
- *tpo* : la valeur de l'ancienne concentration cible de Propofol
- *tr* : la concentration cible de Remifentanyl courante
- *deltaTimeP* : le temps écoulé entre l'ancienne et la nouvelle concentration cible de Propofol
- *deltaTimeR* : le temps écoulé entre la dernière modification de Remifentanyl et la nouvelle concentration cible de Propofol
- *rt* : le runtime en secondes, il s'agit de l'instant courant relatif à l'instant de la première modification de la concentration cible de Propofol
- *age* : il s'agit de l'âge du patient
- *sexe* : variable binaire indiquant si le patient est un homme (1) ou une femme (0)
- *taille* : taille de l'individu en centimètres
- *poids* : masse de l'individu en kilogrammes
- *lbm* : il s'agit de la masse maigre du patient c'est-à-dire la masse totale moins la masse grasseuse

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

La variable de sortie associée aux variables d'entrée est *bisF300* c'est-à-dire la valeur du BIS réel observé 300 secondes (5 minutes) après le dernier changement de la concentration cible de Propofol.

Le processus de sélection de variables a retenu les mêmes variables d'entrée pour les deux modèles, à savoir le *bis*, le *tpn* et le *tpo* à l'instant du dernier changement de Propofol. Ces deux modèles ont montré de bonnes performances théoriques selon l'étude du NMSE, de la MAE et de P. Pour rappel, ces critères sont respectivement l'erreur quadratique moyenne normalisée qui permet de juger de l'efficacité du prédicteur par rapport à la moyenne arithmétique, l'erreur moyenne absolue qui quantifie l'erreur dans la même unité de mesure que le BIS et P qui est le pourcentage de fois que la prédiction a été dans le sens du BIS réel. Ces critères ont été introduits dans la section 3.2.2.

4.2 Construction de nouvelles variables d'entrée

En plus des douze variables d'entrée utilisées dans les modèles précédents, on va créer de nouvelles variables dans l'optique que le modèle de prédiction prenne en compte des informations propres au patient anesthésié. Pour ce faire une série de dix variables additionnelles va être créée pendant les dix premières minutes de l'anesthésie. Le but de ces variables est de mettre en évidence la manière dont le patient réagit à l'injection de Propofol lors de l'induction et de la sorte, évaluer la réactivité du patient face aux injections de Propofol de manière générale.

Ces dix variables construites pendant les dix premières minutes de l'anesthésie sont :

- *pb60* : la concentration de Propofol lorsque la valeur du BIS est la plus proche de 60.
- *pb70* : la concentration de Propofol lorsque la valeur du BIS est la plus proche de 70.
- *pb80* : la concentration de Propofol lorsque la valeur du BIS est la plus proche de 80.
- *meanRemi* : la moyenne des concentrations de Remifentanil :
$$meanRemi = \frac{1}{600} \sum_{i=1}^{N_{tr}} tr_i d(tr_i)$$
 où N_{tr} est le nombre de changements de la concentration cible de Remifentanil, tr_i est la valeur de la concentration cible de Remifentanil et $d(tr_i)$ est la durée pendant laquelle cette concentration cible de Remifentanil a été maintenue. Cette variable donne une indication sur la concentration cible moyenne de Remifentanil qui couplée à la concentration de Propofol a amené le patient au niveau de BIS souhaité par l'anesthésiste.
- *meanProp* : la moyenne des concentrations de Propofol : $meanProp = \frac{1}{600} \sum_{i=1}^{N_{tp}} tr_i d(tp_i)$ où N_{tp} est le nombre de changements de la concentration cible de Propofol, tp_i est la valeur d'une concentration cible de Propofol et $d(tp_i)$ est la durée pendant laquelle cette concentration cible de Propofol a été maintenue. De la même manière que *meanRemi*, cette variable donne une indication sur la concentration cible moyenne de Propofol ayant été nécessaire pour amener le patient au niveau de BIS souhaité par l'anesthésiste.
- *maxProp* : la concentration cible maximale de Propofol injectée. Il s'agit de la concentration choisie par l'anesthésiste en espérant que le BIS du patient chute dans des valeurs comprises entre 40 et 60.
- *timeMaxProp* : l'instant où a été injectée la concentration maximale de Propofol. Typiquement cette valeur vaudra 0, car la première injection de Propofol sera souvent la plus élevée étant donné que c'est elle qui a pour but d'endormir le patient.

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

- *minBis* : la valeur minimale du BIS enregistré. Elle permet de voir la valeur minimale du BIS atteinte lors de la phase d'induction. Cela donne une idée à l'anesthésiste de la manière dont le patient a réagi à l'injection d'une concentration *maxProp* choisie par l'anesthésiste. Prenons par exemple le cas de *minBis* ayant été mesurée à 30. Ceci indique que l'état de conscience a chuté plus que prévu car l'anesthésiste espérait amener le patient à un BIS de 45 par exemple. Ceci indique à l'anesthésiste qu'il pourra, en vue d'améliorer le confort du patient, diminuer la concentration de Propofol à injecter au patient pendant la phase de maintien.
- *timeMinBis* : l'instant où le BIS a atteint sa valeur minimale.
- *r* : information permettant d'estimer la réactivité du patient aux injections de Propofol. Cette variable va estimer si le patient réagit plus ou moins rapidement à la concentration effective, *maxProp*, de Propofol injectée en bolus. *r* se calcule comme ceci : $r = \frac{100 - \text{minBis}}{\text{timeMaxProp} - \text{timeMinBis}}$ qui représente le coefficient de variation du BIS lors de l'induction. Au plus *r* est proche de 1 au plus le patient a réagi rapidement au bolus de Propofol.

Cela porte donc le nombre de variables d'entrée à 22. Dans l'optique de ne garder que les variables qui apportent de l'information à la prédiction, une sélection de variables sera effectuée. Les résultats seront présentés dans la suite de ce chapitre.

4.3 Prédications dans une fenêtre temporelle de 10 minutes

Les modèles disponibles jusqu'ici ne nous permettaient de prédire le BIS qu'à un instant donné dans le futur. Dans notre cas, on va créer plusieurs modèles capables d'effectuer des prédictions toutes les 30 secondes à partir d'un changement de la concentration cible de Propofol et cela dans une fenêtre temporelle de dix minutes. Pour ce faire, l'ensemble d'apprentissage a été modifié de telle sorte qu'à chaque vecteur de paramètres d'entrées corresponde 20 variables de sortie représentant les valeurs du BIS observées toutes les 30 secondes après le dernier changement de la concentration cible de Propofol et cela sur une période de 10 minutes.

Etant donné qu'on a 20 variables de sortie correspondant aux valeurs du BIS toutes les 30 secondes à partir du dernier changement de la concentration cible de Propofol, on va devoir construire 20 sous-modèles capables de faire chacun des prédictions pour chacune des variables de sortie. Cet ensemble de sous-modèles constituera alors le modèle capable d'effectuer la prédiction sur l'intervalle de 10 minutes.

Pour résumer, on va devoir construire un ensemble d'apprentissage sur base d'échantillons contenant les 22 variables d'entrée ainsi que les 20 variables de sortie. Un échantillon de l'ensemble d'apprentissage devra contenir les valeurs pour les variables suivantes :

En entrée : *bis*, *tpn*, *tpo*, *tr*, *deltaTimeP*, *deltaTimeR*, *rt*, *age*, *sexe*, *taille*, *poids*, *lbm*, *pb60*, *pb70*, *pb80*, *meanRemi*, *meanProp*, *maxProp*, *timeMaxProp*, *minBis*, *timeMinBis*, *r*.

En sortie : *bisF30*, *bisF60*, *bisF90*, *bisF120*, *bisF150*, *bisF180*, *bisF210*, *bisF240*, *bisF270*, *bisF300*, *bisF330*, *bisF360*, *bisF390*, *bisF420*, *bisF450*, *bisF480*, *bisF510*, *bisF540*, *bisF570*, *bisF600*.

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

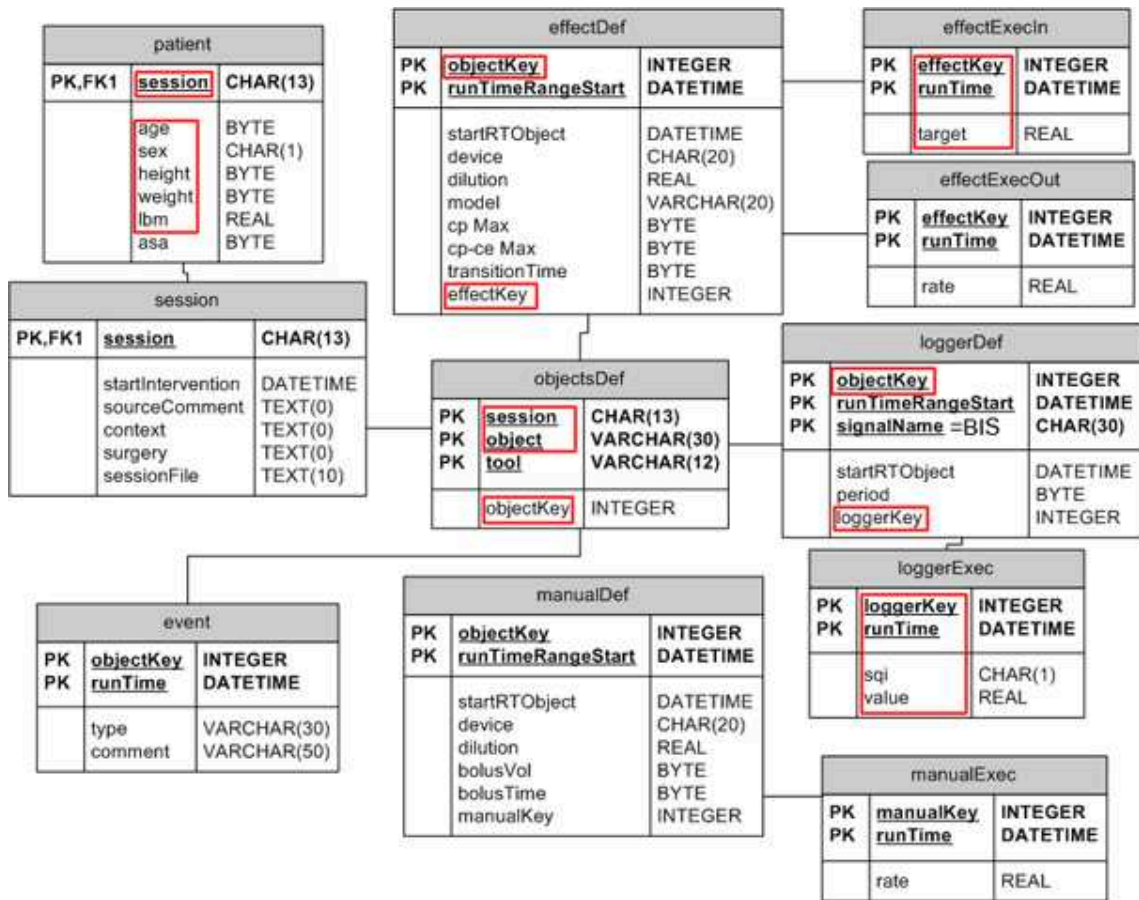


FIG. 4.1: Les composants encadrés en rouge sont ceux qui ont été utilisés pour la création de l'ensemble d'apprentissage.

4.4 Données disponibles

Une base de données contenant des informations qu'Itb95 a collecté pendant plus de 1000 sessions d'anesthésies a été créée. Afin d'optimiser la base de données en fonction du problème qu'on cherche à résoudre, plusieurs pré-traitements ont dû être effectués sur cette base de données. Le schéma relationnel de la base de données utilisée est illustré sur la figure 4.1. Le but de cette section est de montrer les différents pré-traitements qui ont été effectués sur la base de données afin de construire l'ensemble d'apprentissage qui a été mis à ma disposition pour réaliser les apprentissages.

Le premier pré-traitement a consisté à enlever les sessions où :

- des agents autres que le Propofol et le Remifentanil ont été utilisés en débit continu.
- moins de trois changements des concentrations d'agents ont été effectués.
- le temps entre le premier et le dernier changement de concentration de Propofol est inférieur à vingt minutes.
- aucun signal BIS n'a été enregistré.

Au terme de cette opération, il reste 906 sessions contenant de l'information utile.

Le second pré-traitement a consisté à retirer toutes les sessions où les signaux dont

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

l'index de qualité (*sqi* pour Signal Quality) n'était pas suffisamment élevé. Un *sqi* trop faible est signe qu'il y a eu des distortions du signal, liées à l'utilisation du bistouri électrique par exemple, et que l'information délivrée sur le signal est erronée. Cela a amené le nombre de sessions utiles à 752.

Les informations contenues dans ces 752 sessions ont été filtrées afin de créer l'ensemble d'apprentissage. Pour chaque session on va rechercher les runtimes où il y a eu une modification de Propofol ou de Remifentanyl. On va aussi récupérer l'âge, la taille, le poids, le sexe et le lbm du patient. Sur base de ces informations on va construire les variables additionnelles ¹ grâce aux informations dont le runtime est inférieur à dix minutes. Il reste maintenant à tester si la modification de Propofol survenue à l'instant rt est à conserver selon les critères suivants :

- l'action a lieu dans un intervalle de temps pendant lequel la concentration cible de Propofol n'est pas modifiée,
- il y a au moins 3 valeurs de BIS dans l'intervalle $[rt - 30, rt]$,
- il y a au moins 10 valeurs de BIS dans l'intervalle $[rt, rt + 300]$,
- il y a au moins 2 valeurs de BIS dans l'intervalle $[rt + 600, rt + 630]$.

Si tous ces critères sont respectés, alors, on va construire une entrée de l'ensemble d'apprentissage. tpo prend la valeur de la concentration cible de Propofol correspondant à l'action précédente et tpr la valeur de la concentration cible de Propofol de l'action courante. $deltaTimeP$ est l'intervalle de temps entre le changement de la concentration cible tpr et le dernier changement de la concentration cible de Propofol correspondant à tpo . rt est le runtime, c'est l'instant correspondant au changement de la concentration cible tpr . Les runtimes sont relatifs à l'instant de la première modification de la concentration cible de Propofol. bis est la moyenne des valeurs prises par le BIS dans un intervalle de temps $[rt - 30, rt]$. tr correspond à la concentration cible de Remifentanyl à l'instant rt et $deltaTimeR$ le temps qui s'est écoulé depuis l'instant correspondant au dernier tr jusqu'à l'instant rt .

A ce stade on dispose d'un vecteur contenant les valeurs des 22 variables d'entrée : bis , tpr , tpo , $deltaTimeP$, tr , $deltaTimeR$, age , sex , $height$, $weight$, lbm , rt , $pb80$, $pb70$, $pb60$, $meanRemi$, $meanProp$, $maxProp$, $timeMaxProp$, $minBis$, $timeMinBis$, r . Comme on souhaite pouvoir prédire le BIS toutes les 30 secondes dans une fenêtre temporelle de 10 minutes, il convient d'ajouter les valeurs prises par le BIS aux instants $rt + 30 \times i$, $i \in [1, 20]$. Ces outputs sont enregistrés dans l'ensemble d'apprentissage sous le nom de $bisF30$, $bisF60$, ... , $bisF600$.

Une fois que le filtre a été appliqué sur toutes les sessions, l'ensemble d'apprentissage contient $N=332$ échantillons. C'est sur base de ces données que se sont basés les modèles pour réaliser leur apprentissage. Un exemple des valeurs que peuvent prendre les variables d'un échantillon de l'ensemble d'apprentissage est illustré dans le tableau 4.1.

¹ $pb80, pb70, pb60, meanRemi, meanProp, maxProp, timeMaxProp, minBis, timeMinBis, r$

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

bis	59.8	age	41	pb80	4	timeMaxProp	130
tpn	3	sexe	0	pb70	4	minBis	25.5
tpo	2	taille	160	pb60	4	timeMinBis	474
deltaTimeP	934	poids	50	meanRemi	1.78	r	0.1957666
tr	2	lbm	39.04	meanProp	1.795		
deltaTimeR	918	rt	1591	maxProp	4		
bisF30	59.9	bisF180	41.4	bisF330	36.86667	bisF480	30.5
bisF60	49.26667	bisF210	43.2	bisF360	38.26667	bisF510	36.6
bisF90	42.73333	bisF240	41.6	bisF390	36.33333	bisF540	36.46667
bisF120	39	bisF270	40.8	bisF420	36.7	bisF570	35.7
bisF150	38.93333	bisF300	38.9	bisF450	33.93333	bisF600	36.66667

TAB. 4.1: Ce tableau illustre le contenu d'un échantillon de l'ensemble d'apprentissage. Il contient 22 variables d'entrée (tableau du haut) ainsi que 20 variables de sortie (tableau du bas). On peut voir que dans le contexte des variables d'entrée le BIS du patient est passé de 59.9 à 36.6 en dix minutes.

4.5 Méthode d'évaluation des performances des différents modèles

Sur base des données contenues dans l'ensemble d'apprentissage, j'ai réalisé plusieurs apprentissages basés sur des modèles de structures différentes. Λ^{Lin} est un modèle linéaire qui n'exploite pas les informations fournies par les variables additionnelles. Λ^{Lazy} est un modèle de type LL qui n'exploite pas non plus les informations fournies par les variables additionnelles. Les deux autres modèles, Λ^{LinAdd} et $\Lambda^{LazyAdd}$ se basent respectivement sur les mêmes structures que les deux précédents si ce n'est qu'ils tirent parti des informations fournies par les 10 variables additionnelles.

L'étude des performances de ces quatre modèles a été effectuée sur base des critères présentés dans la section 3.2.2. Je vais cependant les remettre dans le contexte actuel et montrer comment ils ont été construits. Pour ce faire, je vais me baser sur la construction du modèle linéaire qui prédit la valeur de *bisF330* soit 5 minutes et 30 secondes après le dernier changement de la concentration de Propofol. Par la suite $h_{bisF330}^{lin}(x, \alpha_N^S)$ désignera la prédiction du BIS lorsque h reçoit x comme vecteur contenant les variables d'entrée. La démarche pour la construction des autres sous-modèles est similaire à celle décrite ci-après.

La sélection de variables est le noyau central de cette étude de performance. La technique utilisée est la Forward Selection introduite dans la section 2.3.3. A chaque étape le processus va construire un ensemble de variables qui va minimiser le NMSE. Voici un exemple de l'évolution de la construction d'un tel ensemble, appliqué à Λ^{Lin} pour la prédiction de *bisF330* :

```
S_1 => bis
S_2 => bis+tpn
S_3 => bis+tpn+tr
S_4 => bis+tpn+tr+tpo
S_5 => bis+tpn+tr+tpo+sex
```

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

S_6 => bis+tpn+tr+tpo+sex+height
 S_7 => bis+tpn+tr+tpo+sex+height+lbm
 S_8 => bis+tpn+tr+tpo+sex+height+lbm+rt
 S_9 => bis+tpn+tr+tpo+sex+height+lbm+rt+deltaTimeP
 S_10 => bis+tpn+tr+tpo+sex+height+lbm+rt+deltaTimeP+deltaTimeR
 S_11 => bis+tpn+tr+tpo+sex+height+lbm+rt+deltaTimeP+deltaTimeR+weight
 S_12 => bis+tpn+tr+tpo+sex+height+lbm+rt+deltaTimeP+deltaTimeR+weight+age

A la fin de chaque étape i de la procédure de sélection de variables, on va créer un vecteur $\hat{E}_{S_i}^{loo}$ contenant l'erreur leave-one-out du modèle dont la structure a été construite sur base de l'ensemble S_i des variables d'entrée sélectionnées par la Forward Selection.

$\hat{E}_{S_i}^{loo} = \langle \hat{E}_{S_i,1}^{loo}, \hat{E}_{S_i,2}^{loo}, \dots, \hat{E}_{S_i,N}^{loo} \rangle$, où $\hat{E}_{S_i,j}^{loo} = h(x_j, \alpha_N^{S_i}) - bisF330_j$ et où $j = 1, \dots, N$ et $bisF330$ est la valeur du BIS enregistré dans l'ensemble d'apprentissage pour l'instant $rt + 300$.

On a donc créé un ensemble de structures qui diffèrent de par les variables d'entrée qu'elles contiennent. A chacune de ces structures S_i a été associé un vecteur d'erreurs $\hat{E}_{S_i}^{loo}$. Grâce à ces vecteurs, on va pouvoir sélectionner la structure S_{best} en choisissant celle qui

génère la plus petite $NMSE$. Si $NMSE(S_i) = \frac{\sum_{j=1}^N (\hat{E}_{S_i,j}^{loo})^2}{var(bisF330)}$ où $var(bisF330)$ est la variance des valeurs prises par $bisF330$ dans l'ensemble d'apprentissage, on choisit S_{Best} comme ceci :

$$S_{best} = \arg \min_{S_i} NMSE(S_i) \quad (4.1)$$

Dès lors, afin de réaliser les prédictions correspondant à la variable de sortie $bisF330$ on utilisera la structure S_{Best} qui a engendré la plus petite $NMSE$. Ce modèle sera par la suite noté $h_{bisF330}^{lin}(\cdot, \alpha_N^{S_{Best}})$

Le calcul de la MAE et de P a aussi été effectué sur base du vecteur d'erreur $\hat{E}_{S_{Best}}^{loo}$.

On notera que

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{E}_{S_{Best},i}^{loo}| \quad (4.2)$$

et que si bis_i est la valeur du BIS courant dans l'échantillon i de l'ensemble d'apprentissage,

$$P = \frac{100}{N} \sum_{i=1}^N I \left[(h_{bisF330}^{lin}(x_i, \alpha_N^{S_{Best}}) - bis_i) \cdot (bisF330_i - bis_i) \right] \quad (4.3)$$

Le calcul de ces trois critères a été réalisé pour les quatre modèles Λ^{Lin} , Λ^{Lazy} , Λ^{LinAdd} et $\Lambda^{LazyAdd}$. Chacun de ces modèles est composé de 20 sous-modèles permettant de prédire le BIS toutes les 30 secondes pendant 10 minutes. La remise en contexte du calcul de ces trois critères n'a été fait que dans le cas de la création d'un modèle de prédiction pour la variable de sortie $bisF330$. Il va de soi qu'il faut répéter la tâche pour tous les 19 autres sous-modèles correspondant aux 19 autres variables de sortie. L'utilisation de ces 20 sous-modèles constitue la prédiction réalisée par le modèle global à savoir celui qui prédit sur un intervalle de temps de 10 minutes.

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

Les résultats de la sélection de variables, du calcul des trois critères ainsi qu'un intervalle de confiance sur la prédiction seront présentés pour chacun des quatre modèles dans la section suivante.

4.6 Présentation des performances des modèles

J'ai pu calculer pour chacun des quatre modèles décrits précédemment les valeurs des trois critères $NMSE$, MAE et P grâce à des scripts écrits en R. A ces trois critères viendra s'ajouter l'écart type du vecteur $\hat{E}_{S_{Best}}^{loo}$. Ce dernier ajout permet de construire un intervalle de confiance autour de la prédiction réalisée par le modèle.

4.6.1 Modèle linéaire sans variables additionnelles

Le tableau suivant présente les résultats du calcul des quatre critères de performance pour le modèle linéaire sans les variables additionnelles. On peut voir que le *bis* courant est la variable qui à elle seule apporte le plus d'informations dans la sélection de variables. Les variables concernant les concentrations de drogues jouent elles aussi un rôle important. On peut aussi constater que les modèles utilisent encore un grand nombre de variables d'entrée et ce malgré la sélection de variables.

rt	Variables sélectionnées	NMSE	MAE	P	sd(errorLoo)
bisF30	bis + height + weight + deltaTimeR + lbm + tpn + tpo + tr	0.2091	3.7467	62.3494	5.4269
bisF60	bis + height + deltaTimeR + sex	0.2556	4.3706	59.3373	6.0075
bisF90	bis + sex + deltaTimeP + deltaTimeR	0.2896	4.459	59.3373	6.2171
bisF120	bis + sex + deltaTimeR + tpo + tpn + deltaTimeP	0.3263	4.7328	62.3494	6.6222
bisF150	bis + sex + tpo + deltaTimeR + deltaTimeP	0.2998	4.7197	61.747	6.4108
bisF180	bis + sex + tpo	0.335	5.0501	62.6506	6.6238
bisF210	bis + sex + deltaTimeR	0.3449	5.0888	61.4458	6.5746
bisF240	bis + sex + tpn + deltaTimeP	0.3251	4.7369	62.0482	6.1472
bisF270	bis + sex + tpn + tpo + tr	0.3252	4.5575	61.1446	6.1995
bisF300	bis + rt + deltaTimeP + sex	0.4072	5.1111	63.8554	7.0552
bisF330	bis + tpn + tr + tpo + sex + height	0.3823	4.9835	64.4578	6.9643
bisF360	bis + tpn + tr + tpo + sex	0.4167	5.2359	68.3735	7.1822
bisF390	bis + tpn + tpo + weight + tr	0.4386	5.2908	66.8675	7.1105
bisF420	bis + tpn + tpo + deltaTimeR + sex + height + lbm + deltaTimeP	0.4202	4.9848	73.1928	6.9738
bisF450	bis + tpn + tpo + sex + height	0.4458	5.004	71.988	6.971
bisF480	bis + tpn + tpo + sex + height + deltaTimeR + rt	0.4454	5.222	71.988	7.1499
bisF510	bis + tpn + tpo + sex + deltaTimeR	0.4217	5.1743	69.5783	6.7124
bisF540	bis + tpn + tpo + sex + height	0.408	5.0449	73.494	6.5877
bisF570	bis + tpn + tpo + sex + height	0.4486	5.1894	71.988	6.9725
bisF600	bis + tpn + tpo + sex + deltaTimeR + height	0.4522	5.2637	69.8795	7.0129

4.6.2 Modèle LL sans variables additionnelles

Le tableau suivant présente les résultats du calcul des quatre critères de performance pour le modèle LL sans les variables additionnelles. Les paramètres structuraux définis dans la section 2.5.2 ayant servi à la construction du modèle sont :

- $k_m^c = 20$ pour le nombre minimum de voisins dans le cadre des modèles locaux constants.
- $k_M^c = 100$ pour le nombre maximum de voisins dans le cadre des modèles locaux constants.
- $k_m^l = 50$ pour le nombre minimum de voisins dans le cadre des modèles locaux linéaires.
- $k_M^l = 150$ pour le nombre maximum de voisins dans le cadre des modèles locaux linéaires.

Le nombre de modèles à combiner a été fixé à 20.

On peut voir que le *bis* courant est de nouveau la variable qui à elle seule apporte le plus d'informations dans la sélection de variables. Cependant la sélection de variables a considérablement réduit le nombre de variables d'entrée car leur nombre maximum est de quatre à présent. Les résultats des calculs des critères nous indiquent que ce modèle est nettement plus précis que son homologue linéaire dont les performances ont été énumérées ci-avant. Cela est dû au fait qu'on a combiné un grand nombre de modèles (20 en l'occurrence) locaux, constants et linéaires autour du query-point, lorsqu'on a effectué le calcul de l'erreur leave-one-out et on sait que la combinaison de modèles a comme conséquence de réduire la variance de la prédiction. C'est le cas ici, car la MAE est nettement inférieure à celle du modèle linéaire.

rt	Variables sélectionnées	NMSE	MAE	P	sd(errorLoo)
bisF30	bis + deltaTimeR + deltaTimeP	0.0144	1.1371	92.7711	1.4056
bisF60	bis + rt + tpo	0.0194	1.2763	90.6627	1.6534
bisF90	bis + rt + age	0.0328	1.6144	88.5542	2.0543
bisF120	bis + deltaTimeP + tr	0.0387	1.8415	89.4578	2.2733
bisF150	bis + deltaTimeP + age	0.03	1.5593	91.2651	1.9915
bisF180	bis + rt + tr	0.0476	1.9755	86.747	2.4334
bisF210	bis + rt + deltaTimeP	0.0733	2.4055	87.9518	2.8834
bisF240	bis + rt + tr	0.0759	2.3649	86.1446	2.9177
bisF270	bis + rt + tr	0.0791	2.3943	84.6386	3.0434
bisF300	bis + deltaTimeR + tpn	0.0971	2.7142	87.6506	3.4423
bisF330	bis + deltaTimeR + deltaTimeP	0.0784	2.5012	88.253	3.1529
bisF360	bis + deltaTimeR + deltaTimeP + tpo	0.1007	2.8026	90.0602	3.5246
bisF390	bis + rt + height	0.1385	3.2292	87.3494	3.9833
bisF420	bis + rt + height	0.1475	3.2935	87.0482	4.1261
bisF450	bis + deltaTimeP + weight	0.1875	3.4764	88.5542	4.5146
bisF480	bis + deltaTimeR + tpo	0.1682	3.4899	87.3494	4.3911
bisF510	bis + rt + deltaTimeP	0.164	3.3818	86.4458	4.1844
bisF540	bis + rt + height	0.1616	3.3117	87.9518	4.1453
bisF570	bis + rt + weight	0.1866	3.5756	89.1566	4.4858
bisF600	bis + rt + deltaTimeP	0.1802	3.5177	84.3373	4.4266

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

4.6.3 Modèle linéaire avec variables additionnelles

Il est intéressant de noter que dans le cas du modèle linéaire avec variables additionnelles, au terme de la procédure de sélection de variables, 13 modèles parmi les 20 construits se basent sur les nouvelles variables pour réaliser la prédiction. On remarquera aussi une légère amélioration du critère P , tandis que les autres critères ne semblent pas montrer d'améliorations significatives comme le montre la section 4.6.5.

rt	Variables sélectionnées	NMSE	MAE	P	sd(errorLoo)
bisF30	bis + height + r + weight + deltaTimeR + tpn + tpo + minBis + pb60 + pb70 + lbm	0.2071	3.7204	64.759	5.4015
bisF60	bis + height + meanProp + r + deltaTimeR + tpn	0.2527	4.3321	59.6386	5.9731
bisF90	bis + sex + r + deltaTimeP + deltaTimeR + maxProp + pb70 + pb60 + age	0.287	4.5044	62.0482	6.1893
bisF120	bis + sex + r + deltaTimeR + tpo + tpn + timeMaxProp	0.3255	4.7982	62.3494	6.6147
bisF150	bis + sex + tpo + deltaTimeR + deltaTimeP	0.2998	4.7197	61.747	6.4108
bisF180	bis + sex + tpo	0.335	5.0501	62.6506	6.6238
bisF210	bis + sex + meanRemi	0.3447	5.0801	62.6506	6.5727
bisF240	bis + sex + tpn + r	0.325	4.7372	62.6506	6.1465
bisF270	bis + sex + tpn + tpo + tr	0.3252	4.5575	61.1446	6.1995
bisF300	bis + rt + deltaTimeP + sex	0.4072	5.1111	63.8554	7.0552
bisF330	bis + tpn + tr + tpo + sex + height	0.3823	4.9835	64.4578	6.9643
bisF360	bis + tpn + tr + tpo + sex	0.4167	5.2359	68.3735	7.1822
bisF390	bis + tpn + tpo + timeMinBis + lbm + height	0.4377	5.2625	70.1807	7.1032
bisF420	bis + tpn + tpo + r + deltaTimeR + sex + height + lbm + deltaTimeP	0.4178	4.9817	73.494	6.9542
bisF450	bis + tpn + tpo + sex + r + pb70 + pb80 + height + deltaTimeR	0.4406	4.9836	74.3976	6.9301
bisF480	bis + tpn + tpo + sex + r + deltaTimeR + height + pb70 + pb80 + rt	0.4434	5.255	71.6867	7.1344
bisF510	bis + tpn + tpo + sex + r + deltaTimeR	0.4191	5.2098	69.8795	6.691
bisF540	bis + tpn + tpo + sex + r + maxProp + age	0.4024	5.0277	72.2892	6.5417
bisF570	bis + tpn + tpo + sex + r + timeMinBis	0.4472	5.1881	72.8916	6.9619
bisF600	bis + tpn + tpo + sex + r + deltaTimeR + age	0.4511	5.2864	70.1807	7.0042

Interprétation de la valeur des coefficients pour la prédiction de *bisF450*

La commande *coefficients* de *R* nous permet pour un modèle linéaire de connaître rapidement la valeur des coefficients de pondération des variables d'entrée. Voici en guise

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

d'exemple le résultat de cette commande sur le modèle de prédiction de *bisF450* une fois l'affichage reformaté.

```
bisF450 ~ bis + tpn + tpo + sexe + r + pb70 + pb80 + height
```

```
(Intercept)    33.6830329776773
bis            0.668310959731682
tpn           -6.53401160856864
tpo            4.34300235066037
sexe           2.73341335019194
r             -2.24635385873144
pb70           1.74985888077363
pb80          -1.29851006199318
taille        -0.0995563652381111
```

Dans ce cas-ci, on peut interpréter les coefficients des variables les plus significatives comme ceci :

- *sexe* : si c'est un homme, son BIS aura tendance à être plus élevé.
- *tpn* : la variable *tpn* est la concentration de drogue injectée au patient. Le coefficient négatif nous dit bien que le BIS a tendance à chuter lorsqu'on injecte du Propofol.
- *tpo* : le coefficient est positif cette fois, mais moins puissant que celui du *tpn*. Cela signifie qu'indépendamment des autres variables, si $tpn = 4$ et $tpo = 3$ on va avoir $-6.5 \times 4 + 4.3 \times 3 \approx -14$. Il détecte bien qu'on a augmenté la concentration de Propofol et que le BIS devrait par conséquent diminuer.
- *bis* : le modèle semble encore attacher une grande importance à la valeur du BIS courant lors du changement de la concentration de Propofol. Dans le contexte la variable *bis* prend des valeurs allant généralement de 40 à 60. Ce coefficient ferait donc varier la valeur du bis prédit de 24 à 36, ce qui est relativement élevé face aux autres variables.
- *intercept* : l'intercept est le coefficient β_0 introduit dans l'annexe A.

Ceci illustre le fait que le modèle d'apprentissage réalise une bonne interprétation du contexte dans lequel il se prépare à réaliser la prédiction.

4.6.4 Modèle LL avec variables additionnelles

rt	Variables sélectionnées	NMSE	MAE	P	sd(errorLoo)
bisF30	bis + deltaTimeR + minBis	0.0141	1.0998	92.1687	1.4013
bisF60	bis + rt + timeMinBis	0.0191	1.2754	90.6627	1.6408
bisF90	bis + rt + timeMinBis + minBis	0.0314	1.5714	89.1566	2.0187
bisF120	bis + deltaTimeP + minBis	0.0364	1.7479	90.3614	2.2063
bisF150	bis + deltaTimeP + minBis	0.0282	1.5176	90.6627	1.9237
bisF180	bis + rt + timeMinBis	0.0453	1.8803	88.253	2.3756
bisF210	bis + rt + deltaTimeP	0.0733	2.4055	87.9518	2.8834
bisF240	bis + rt + minBis + pb60	0.0741	2.3225	82.8313	2.8831
bisF270	bis + timeMaxProp + timeMinBis + age + r	0.0744	2.2931	85.8434	2.9558
bisF300	bis + timeMaxProp + meanRemi	0.0904	2.5857	86.4458	3.3157
bisF330	bis + timeMaxProp + deltaTimeR	0.0772	2.4989	88.8554	3.125
bisF360	bis + timeMaxProp + age + minBis	0.1002	2.8073	88.5542	3.5125
bisF390	bis + rt + minBis	0.1365	3.2143	86.1446	3.9579
bisF420	bis + rt + minBis	0.144	3.2734	86.747	4.0766
bisF450	bis + timeMaxProp + age + pb70	0.1707	3.4501	89.1566	4.2969
bisF480	bis + deltaTimeR + maxProp	0.1679	3.4666	88.5542	4.3878
bisF510	bis + timeMaxProp + pb70	0.165	3.3055	84.9398	4.1965
bisF540	bis + rt + height	0.1616	3.3117	87.9518	4.1453
bisF570	bis + rt + weight	0.1866	3.5756	89.1566	4.4858
bisF600	bis + rt + deltaTimeP	0.1802	3.5177	84.3373	4.4266

4.6.5 Interprétation des résultats théoriques

Pour chaque variable de sortie $bisF30, bisB60, \dots, bisF600$ nous avons 4 modèles à comparer. Pour se faire un test de Wilcox [31] a été réalisé sur chacun des vecteurs $\hat{E}_{S_{Best}}^{loo}$ respectifs aux 4 modèles, afin de voir si la différence au niveau de la $NMSE$ est significative ou non. L'erreur l-o-o a toutefois été portée au carré.

Au niveau des p-values résultantes de la comparaison des erreurs l-o-o des modèles linéaires, on n'a pas suffisamment d'évidence pour affirmer que les deux modèles sont différents. Cela signifie qu'on ne peut affirmer que le modèle avec variables additionnelles soit meilleur que celui sans.

La même comparaison entre les modèles linéaires et LL a donné suffisamment d'évidence (la p-value inférieure à 5%) pour affirmer que la distribution des erreurs l-o-o était différente. Cela signifie qu'en se basant sur la $NMSE$ comme critère de comparaison de la qualité de prédiction des modèles, la classe de modèle LL est meilleure que la classe linéaire.

En effectuant le test de Wilcox une dernière fois entre les deux modèles LL (avec et sans variables additionnelles) les p-values nous indiquent que seuls les modèles de prédiction de $bisF120$ présentent une différence significative au niveau de leur distribution des erreurs l-o-o. Dans ce cas, toujours en se basant sur le critère $NMSE$, on peut affirmer que le modèle Lazy avec variables additionnelles présente de meilleures performances que son homologue n'utilisant pas les variables additionnelles.

CHAPITRE 4. AMÉLIORATIONS DES MODÈLES EXISTANTS

Au niveau des types de modèles, les modèles LL l'emportent haut la main sur tous les critères. A présent, intéressons-nous à mesurer l'apport en précision fourni par les variables additionnelles. Au niveau du modèle linéaire, on constate que la diminution de la *NMSE* est négligeable. Par contre, cela a sensiblement amélioré le critère *P* où dans le cas de la prédiction de *bisF390* on notera une amélioration de l'ordre de 3%.

Au terme du chapitre suivant, ces différents modèles de prédictions seront analysés face à un jeu de données issu d'un test en conditions réelles. Sur base d'un graphique comparatif, on pourra juger de la capacité de généralisation des quatre modèles.

Chapitre 5

BisPredict, un outil d'aide à la décision en anesthésie

Afin de pouvoir utiliser les prédictions fournies par les modèles présentés dans le chapitre précédent en conditions réelles, j'ai implémenté un outil, *BisPredict*, qui embarque un modèle de prédiction et qui permet de récupérer des données provenant des appareils médicaux. Ces données sont raffinées afin de les présenter en entrée du modèle qui effectuera les prédictions. Cela se traduira par l'affichage à l'écran d'une courbe correspondant aux valeurs de BIS prédites par le modèle. Pour communiquer avec les appareils médicaux, j'ai utilisé *ARAQ*, une librairie développée par Olivier Cailloux dans le cadre du projet *TANIA* [2], permettant de contrôler Itb95 [5].

Un exemple d'utilisation de *BisPredict* en temps réel est illustré sur la figure 5.1.

Ce chapitre a comme objectif de donner un aperçu général de l'implémentation et des fonctionnalités de *BisPredict*. Je parlerai ensuite de la validation statistique du modèle embarqué face aux données analysées lors des anesthésies. Je terminerai par faire une validation qualitative de l'outil *BisPredict* basée sur les remarques des anesthésistes par rapport aux prédictions réalisées par l'outil.

5.1 Présentation de l'outil

BisPredict est un outil dont la fonctionnalité principale est de fournir une aide à la décision à l'anesthésiste grâce aux prédictions que l'outil réalise. D'autres tâches sont cependant réalisées en arrière plan. Je vais commencer par décrire une liste exhaustive des fonctionnalités de *BisPredict*.

Au niveau des fonctionnalités :

- Récupération de données provenant des appareils médicaux : grâce à l'utilisation d'*ARAQ* comme librairie j'ai pu récupérer les données nécessaires à la construction des variables d'entrée du modèle.
- Transformation de ces données pour les rendre utilisables par le modèle de prédiction embarqué.
- Prédiction de l'évolution du BIS : manuelle et automatique. La prédiction manuelle illustrée sur la figure 5.2 simule l'injection d'une nouvelle concentration de Propofol

CHAPITRE 5. BISPREDICT, UN OUTIL D'AIDE À LA DÉCISION EN ANESTHÉSIE

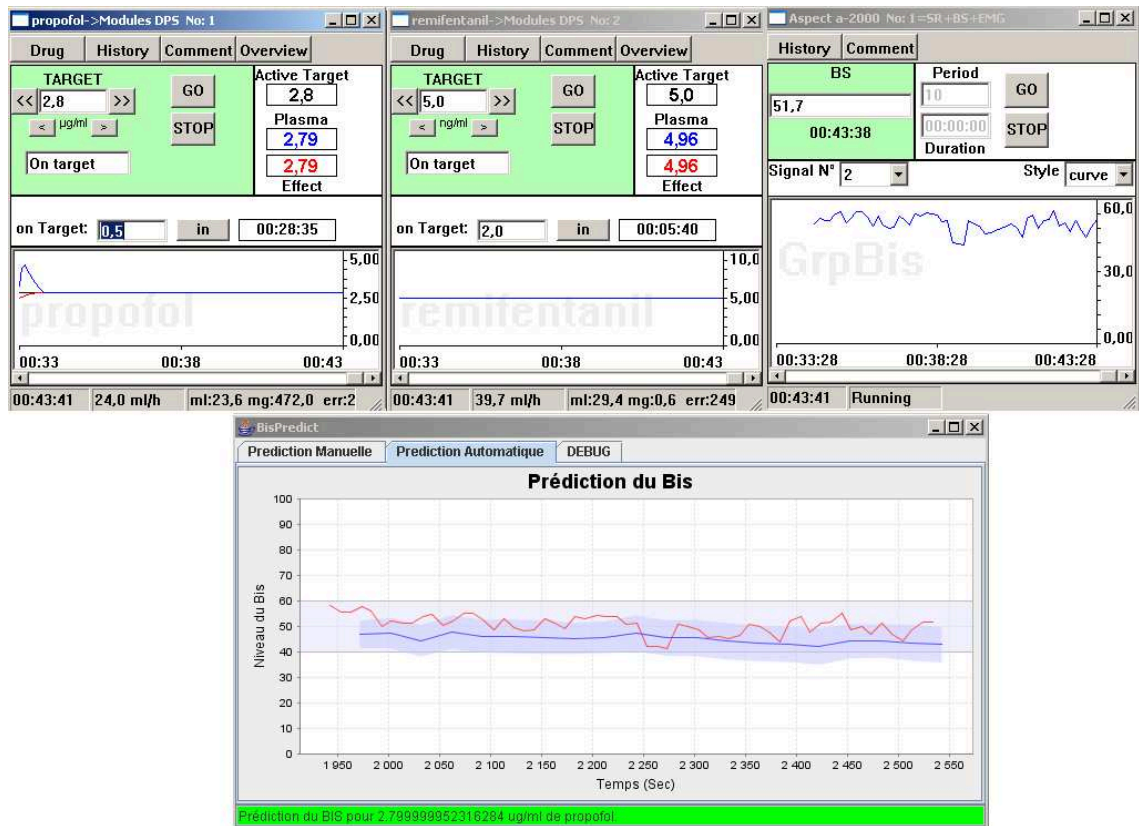


FIG. 5.1: Prédiction du BIS réalisée par BisPredict. La courbe rouge représente les valeurs prises par le BIS réel, tandis que la courbe bleue représente la prédiction du BIS dans la fenêtre temporelle de 10 minutes. La zone grisée autour de la prédiction représente l'intervalle de confiance sur base de l'écart type des erreurs 1-o-o commises par le modèle.

et affiche une prédiction de l'évolution du BIS pour cette concentration test. L'anesthésiste entre la concentration pour laquelle il désire réaliser une prédiction, appuie sur le bouton GO et la prédiction est affichée. La prédiction automatique illustrée sur la figure 5.1 consiste à effectuer une prédiction lors de chaque modification de la concentration de Propofol dans Itb95.

- Détection des outliers afin de mettre l'anesthésiste en garde sur l'imprécision de la prédiction réalisée sur un échantillon aberrant.
- Exportation des données dans un format structuré permettant d'effectuer une validation statistique du modèle embarqué.
- Une interface de debug intégrée qui permet de s'assurer du bon fonctionnement de l'outil.

Au niveau de la continuité du projet :

BisPredict embarque un modèle linéaire. Le modèle est implémenté en Abstract Data Type, c'est-à-dire qu'il est utilisé comme une boîte noire prenant des données structurées en entrée afin de produire un résultat sous forme de données structurées. BisPredict est donc utilisable pour d'autres types de modèles dès lors que le nouveau modèle implémenté respecte ce format d'entrée/sortie. Les données en entrée sont stockées dans

un vecteur contenant la valeur de chacune des variables d'entrée. La sortie est composée d'un vecteur qui contient la valeur de la prédiction toutes les 30 secondes et d'un autre vecteur qui contient l'intervalle de confiance pour chaque prédiction.

5.2 Implémentation

Cette section a pour but de donner une vision d'ensemble de l'implémentation de BisPredict. Je commencerai par introduire la solution qui a été choisie afin de communiquer avec les appareils médicaux. Ensuite, je détaillerai le fonctionnement interne du logiciel sur base d'un diagramme de packages. Pour terminer la section, je parlerai des variables construites à différents instants clés de l'exécution de BisPredict. J'aborderai aussi les différentes techniques utilisées afin de détecter les échantillons dits outliers qui, présentés en entrée du modèle, risquent de délivrer une prédiction aberrante.

Voici le contexte d'implémentation de BisPredict :

- Système d'exploitation cible : Microsoft Windows XP SP1.
- Langage de développement de BisPredict : Java
- Développement des modèles : R
- Autres logiciels et bibliothèques : Itb95, Araq, Swing et JFreeChart

5.2.1 Communication avec les appareils médicaux

Cette section montre comment BisPredict communique avec les appareils médicaux. Pour ce faire, j'ai utilisé ARAQ, une bibliothèque permettant de communiquer avec les appareils médicaux en interagissant avec Itb95.

Avant d'aller plus loin, il est important de voir comment on peut récupérer des données provenant d'Itb95.

Itb95 est en mesure de lire les informations renvoyées par les appareils médicaux et est aussi capable de commander les injecteurs. En plus, Itb95 dispose d'un serveur de communication permettant à un logiciel client de venir s'y connecter afin de communiquer avec lui. ARAQ est un logiciel capable d'interagir directement avec Itb95 et donc indirectement avec les appareils médicaux. L'utilisation d'ARAQ en tant que logiciel permet d'ajouter des fonctionnalités à Itb95 et d'utiliser une autre interface graphique que celle d'Itb95. ARAQ peut aussi se décliner sous la forme d'une bibliothèque et dans ce cas permet à un logiciel-pilote utilisant la bibliothèque de communiquer avec les appareils médicaux. Dans notre cas, BisPredict est ce logiciel pilote. Grâce à l'utilisation d'ARAQ comme bibliothèque, BisPredict peut donc communiquer indirectement avec les appareils médicaux. Dans notre cas, il ne sera pas nécessaire de commander Itb95. On se contentera de collecter les données qu'il envoie.

Les données collectées par BisPredict proviennent de deux sources. La première vient des appareils médicaux qui vont nous fournir les informations sur le niveau de BIS actuel. La deuxième source est le logiciel Itb95 qui nous permettra de récupérer les données concernant les niveaux de drogues injectées au patient. Les données concernant les niveaux de drogues sont directement accessibles sur Itb95 car c'est ce dernier qui pilote les injecteurs en fonction des concentrations cibles de drogues à atteindre. ARAQ

lit cette information sur le serveur de communication d'Itb95 et la met à disposition de BisPredict. L'information concernant le BIS devra d'abord être envoyée du moniteur de BIS jusqu'à Itb95. Puis suivra le même trajet que les informations concernant les niveaux de drogues. Il est important de noter qu'il est possible de capter de nombreux autres signaux provenant d'Itb95. Sans être exhaustif, on pourra citer les débits des différentes perfusions ou encore l'index de qualité du signal BIS (*sqi*).

Par la suite, je considérerai ARAQ et Itb95 comme une seule et même entité nommée *ARAQ-Itb* me permettant de communiquer directement avec les appareils médicaux.

5.2.2 Les différentes phases traversées par BisPredict

Maintenant qu'on sait comment communiquer avec les appareils médicaux, on va pouvoir collecter des informations qui vont servir à alimenter le modèle. Le prochain paragraphe décrit les différentes phases que traverse BisPredict afin de créer l'ensemble des variables d'entrée du modèle.

- Initialisation Araq : création d'un communicateur permettant de se mettre à l'écoute d'évènements provenant des appareils médicaux. De tels évènements sont par exemple un changement du BIS ou encore lors des changements des targets de Propofol ou de Remifentanil.
- La phase 1 commence lors du premier changement de la concentration cible de Propofol. Elle correspond à la phase d'induction de l'anesthésie. Pendant 10 minutes, on va collecter les données *runtime*, *tpn*, *tpo* et le *bis* à chaque modification du niveau de BIS ou de la concentration de drogue. Par la suite, tous les runtimes enregistrés seront relatifs à l'instant du premier changement de la concentration de Propofol.
- A la fin de la phase 1, on va construire les variables additionnelles sur base du tableau construit pendant les 10 minutes et en récupérant des données statiques concernant le patient (âge, poids, sexe...).
- Entrée dans la phase 2. C'est ici que les prédictions sont utilisées. On est toujours à l'écoute des mêmes évènements. Un vecteur contenant les variables d'entrée du modèle est à jour en permanence. Dès qu'un évènement se produit le vecteur est mis à jour. Deux types de prédictions peuvent être réalisées :
 - Prédiction manuelle illustrée sur la figure 5.2 : Copie du vecteur d'entrée et modification des variables adéquates (*runtime*, *deltaTimeP*, *deltaTimeR*, *tpn*, *tpo*,...). On ne doit pas modifier le vecteur d'entrée.
 - Prédiction automatique illustrée sur la figure 5.1 : lors d'un changement de Propofol, comme lors de chaque évènement, le vecteur de variables d'entrée du modèle est mis à jour. Il fait appel à la méthode `makePredictionRT(variablesEntrée)` qui nous renvoie le vecteur contenant les prédictions pour les 10 minutes suivant le changement de Propofol. La prédiction globale est ensuite affichée sur le panel de prédiction automatique.

Pendant toute la durée de son exécution, BisPredict va être à l'écoute d'évènements déclenchés par ARAQ-Itb. Comme je l'ai mentionné précédemment, ARAQ-Itb crée de nombreux évènements mais dans notre cas, seuls les modifications du niveau de BIS et le changement des concentrations cibles (target) de Propofol et de Remifentanil nous intéressent. Un évènement de type BIS est déclenché toutes les dix secondes, tandis que les évènements de type target se produisent lorsque l'anesthésiste change la concentration

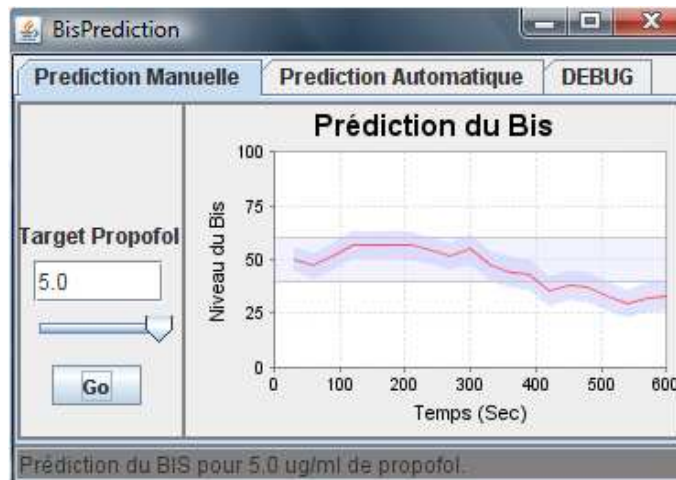


FIG. 5.2: Illustration de l'interface de prédiction manuelle. L'anesthésiste entre la concentration de Propofol pour laquelle il désire faire une prédiction et clique sur GO afin de la déclencher.

d'un des deux agents anesthésiques. Parmi les évènements de type target, on distinguera les targets de Propofol et les targets de Remifentanyl.

5.2.3 Diagramme des packages

Le but ici n'est pas de rentrer dans les détails de l'implémentation, mais de donner un aperçu des relations entre les différents packages de l'implémentation. J'expliquerai néanmoins les responsabilités de chaque package ainsi que les relations qui les associent.

Les explications ci-dessous se basent sur la figure 5.3 qui illustre les relations entre chaque package. Je vais d'abord expliquer le rôle de chaque package.

- `mainApp` : il s'agit du chef d'orchestre. C'est lui qui coordonne le travail. Il va commencer par créer l'interface graphique, initialiser une connexion avec Itb95 grâce aux méthodes implémentées dans ARAQ et puis va lancer la phase de construction des variables additionnelles et attendre que celle-ci soit terminée pour passer dans la phase de prédictions.
- `mainGui` : regroupe toutes les classes implémentant l'interface graphique. Il s'agit d'un panel comprenant trois onglets. Le premier concerne l'onglet prédiction manuelle. Il contient un panel destiné au graphique de la prédiction et un autre panel permettant à l'anesthésiste de simuler un changement de Propofol et de réaliser une prédiction. Le second onglet est destiné à la prédiction automatique, il ne dispose que d'un panel pour le graphique de la prédiction. Sur ce panel, contrairement à celui de la prédiction manuelle, la courbe du BIS réel est superposée à la courbe de la prédiction (voir figure 5.1). Le dernier onglet fait office de debug. Un dernier panel commun à tous les onglets sert pour l'affichage des messages d'informations. Il est en bas de la fenêtre de BisPredict.
- `itbConnexion` : ce package se charge de créer une connexion avec Itb95 au travers des méthodes implémentées dans ARAQ. Il instancie un communicateur qui nous permet de nous mettre à l'écoute d'Itb95.
- `collectDataPhase` : il implémente tout ce qui est nécessaire à la construction des

variables additionnelles. Grâce au communicateur créé par `itbConnexion`, il commence par se mettre à l'écoute d'évènements provenant d'`Itb95`. Ces évènements sont de deux types, `BIS_INDEX` et `TARGET` et se déclenchent respectivement toutes les 10 secondes afin de donner le niveau du BIS et lorsqu'un target de Propofol ou de Remifentanyl a été changé via `Itb95`. Des méthodes liées au communicateur permettent de récupérer les valeurs du BIS et des concentrations de drogues. Lors du premier changement du target de Propofol, un compte à rebours de 10 minutes est déclenché. Il s'agit de la durée pendant laquelle les données servant à la construction des variables additionnelles vont être collectées. Au terme de ces 10 minutes, le calcul des variables additionnelles est effectué. Les variables collectées ainsi que les calculs effectués sur ces données ont été expliqués dans le chapitre précédent à la section 4.4. Pour terminer, `collectDataPhase` signale au chef d'orchestre `mainApp` que son travail est terminé et qu'il peut reprendre la main.

- `predictionPhase` : il est initialisé par `mainApp` une fois que `collectDataPhase` a terminé son travail. `mainApp` lui fournit le vecteur contenant les variables additionnelles. `predictionPhase` se remet à l'écoute des mêmes évènements que dans `collectDataPhase`. Ici, on va tenir à jour un vecteur de variables qui vont servir d'entrée au modèle. A chaque évènement le vecteur sera mis à jour. L'arrivée d'un évènement de type target de Propofol déclenchera une prédiction qui est affichée sur l'onglet central, celui réservé aux prédictions automatiques. En plus d'écouter les évènements provenant des outils médicaux, `predictionPhase` s'est mis à l'écoute du bouton GO de l'onglet de prédiction manuelle. Cela permet de déclencher une prédiction sur l'onglet de gauche correspondant aux prédictions manuelles. Le modèle de prédiction est implémenté dans ce package. Il s'agit d'une boîte noire qui prend le vecteur de variables d'entrée et qui retourne un vecteur correspondant aux prédictions de BIS toutes les 30 secondes à partir de l'instant du dernier changement de Propofol jusque 10 minutes plus tard.

5.2.4 Construction des variables

Dans `BisPredict` on distingue deux phases. La première consiste à collecter durant la phase d'induction de l'anesthésie des informations servant à construire les variables additionnelles. Ces variables additionnelles viendront s'ajouter aux variables mises à jour en temps réel lors de la seconde phase de l'anesthésie (la phase de maintien). J'explique ci-après quelles informations sont collectées lors de chacune des deux phases et quelles variables elles servent à construire.

Pour résumer, on va construire trois types de variables :

- Les variables statiques, à savoir l'âge, le sexe, la taille, le poids et le `lbm`.
- Les variables additionnelles. Elles sont construites au terme de la phase d'induction de l'anesthésie. Elles sont au nombre de 10 et ont été détaillées dans la section 4.2.
- Les variables dynamiques. Elles sont au nombre de 7 et nécessitent d'être mises à jour lors de l'apparition d'un évènement lors de la phase de maintien. Ces variables ont quant à elles été décrites dans la section 4.1.

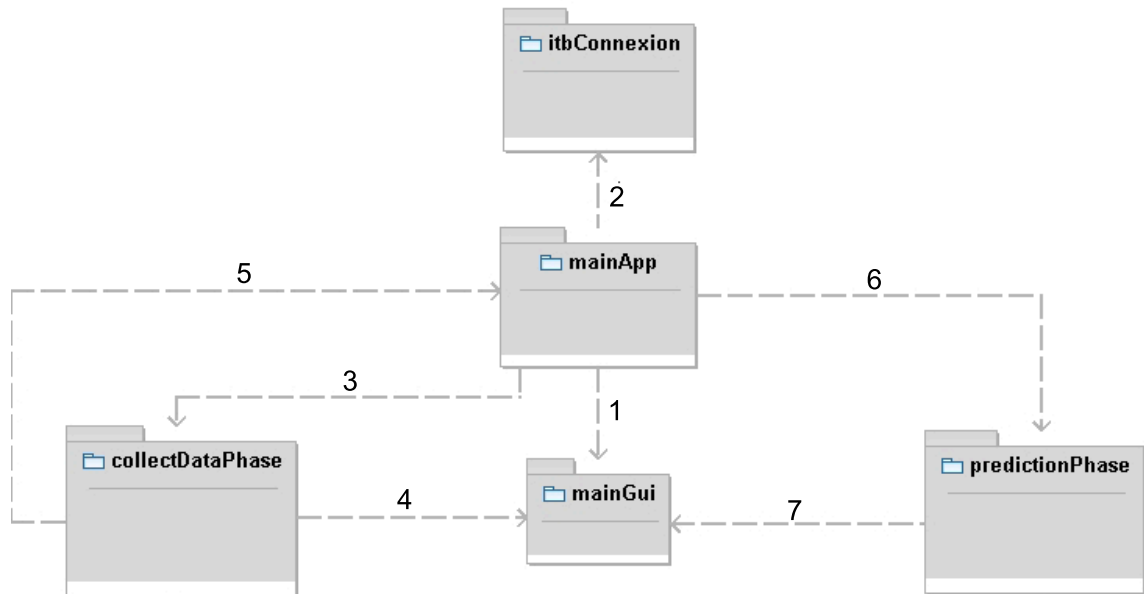


FIG. 5.3: Relations entre les différents packages. Si une flèche relie A à B, l'interprétation à faire est la suivante : A fait appel à B. Les numéros en vis-à-vis des flèches indiquent l'ordre dans lequel on accède aux packages.

Pendant la phase d'induction

Durant cette phase on va construire un tableau contenant quatre informations : le *runtime*, le *t_{pn}*, le *tr* et la valeur du *bis* correspondant au *runtime*. Une entrée est ajoutée au tableau lors d'un changement du BIS, ou lors des changements concentrations de Propofol ou de Remifentanyl. Le tableau 5.1 donne un exemple du contenu du tableau au terme de la phase d'induction de l'anesthésie.

Sur base des données contenues dans le tableau 5.1, voici les valeurs prises par les variables additionnelles (voir section 4.2) :

- $Pb80 : 3$
- $Pb70 : 3$
- $Pb60 : 3$
- $meanRemi : \frac{1}{600}(29 \times 2 + 571 \times 7) = 6.75$
- $meanProp : \frac{1}{600}(600 \times 3) = 3$
- $maxProp : 3$
- $timeMaxProp : 0$
- $minBis : 47$
- $TimeMinBis : 107$
- $r : \frac{(100 - minBis)}{timeMaxProp - timeMinBis} = -0.49$

A ces variables calculées, on ajoutera les variables statiques suivantes correspondant aux informations concernant le patient, à savoir l'âge, la taille, le poids, le sexe et lbm.

Une fois que ces variables sont construites au terme de la phase 1, elles seront utilisées

Runtime	TPN	TR	BIS
0	3	2	95
7	3	2	93
17	3	2	90
27	3	2	85
29	3	7	87
37	3	7	85
47	3	7	85
⋮	⋮	⋮	⋮
107	3	7	47
⋮	⋮	⋮	⋮
600	3	7	54

TAB. 5.1: Exemple fictif de collecte des données lors de la phase d'induction.

par le modèle dans la phase 2 où aucune autre modification ne leur sera apportée.

Pendant la phase de maintien

Ici il convient de garder les variables d'entrée du modèle à jour en permanence, car l'anesthésiste peut à tout moment demander une prédiction manuelle. Les variables d'entrée sont stockées dans un vecteur. Afin de maintenir le vecteur d'entrée à jour, il faudra effectuer des modifications lors de l'apparition des évènements suivants :

- Modification du niveau de BIS : mise à jour du BIS et du *runtime* dans le vecteur.
- Modification d'un target de Propofol : mise à jour de *runtime*, *t_{pn}*, *t_{po}*, *deltaTimeP* et de *deltaTimeR*.
- Modification d'un target de Remifentanyl : mise à jour de *runtime*, *tr* et d'une variable temporaire *timeLastTr* servant à calculer le *deltaTimeR* lors d'un changement du target de Propofol.

Grâce à ces opérations exécutées lors de l'apparition d'un évènement, le vecteur des variables d'entrée du modèle est à jour constamment. Les données sont donc en permanence utilisables par le modèle.

5.2.5 Détection des outliers

Au moment où l'anesthésiste va demander à BisPredict de réaliser une prédiction, BisPredict va analyser le contexte de la prédiction afin d'éventuellement mettre l'anesthésiste en garde sur la défaillance de la prédiction face aux données en entrée du modèle. Il se peut que la prédiction soit effectuée dans un contexte étranger à celui de l'ensemble d'apprentissage sur lequel l'apprentissage a été effectué.

La première technique ayant été testée a considéré chaque variable indépendamment les unes des autres et a calculé les quantiles à 5% et 95% sur la distribution de fréquences de chacune des variables, de sorte à ne conserver que 90% des informations. Si le modèle venait à vouloir effectuer une prédiction pour un échantillon se trouvant en dehors

CHAPITRE 5. BISPREDICT, UN OUTIL D'AIDE À LA DÉCISION EN ANESTHÉSIE

Nom	Quantile 5%	Quantile 95%	Moyenne
bis	24.90833	64.33333	44.40291
tpn	1.2	3.5	2.087819
tpo	1.3	3.65	2.352394
deltaTimeP	48.75	4139.75	1133.213
tr	2	8	4.710293
deltaTimeR	42.75	3252.75	948.4521
sexe	0	1	0.481383
taille	155	185	169.1649
poids	51	98	71.92287
lbn	38.92502	68.05853	52.69366
runtime	818.50	14283.25	5412.043
pb80	0.95	4.50	2.698245
pb70	1	5	2.810771
pb60	1	5	2.874229
meanProp	0.500000	3.358333	1.546064
maxProp	1.4	5.0	3.174867
minBis	24.725	84.200	48.96144
timeMinBis	322	597	523.0718
timeMaxProp	53	577	311.4187
R	0.02332952	0.86350561	0.2250466

TAB. 5.2: Ce tableau montre les quantiles de la distribution de chaque variable pour 5% et 95%.

des limites fixées par les quantiles, l'anesthésiste sera averti que le contexte n'est pas favorable à une prédiction de qualité car le modèle n'a pas eu l'occasion de s'entraîner suffisamment sur de tels échantillons. Le résultat de la prédiction dans ce contexte a de plus fortes chances de ne pas refléter la réalité. Le résultat des ces tests est présenté dans le tableau 5.2. Un exemple d'utilisation des quantiles est donné sur la distribution de fréquences de la variable BIS sur la figure 5.4.

L'autre technique utilisée et implémentée dans BisPredict repose sur la notion de *distance de Mahalanobis* [32]. Il s'agit ici de calculer la distance d'un échantillon par rapport au centre de masse de l'ensemble d'apprentissage en tenant compte des dépendances entre chaque variable de l'ensemble d'apprentissage. Pour ce faire on va utiliser la matrice de covariance Σ .

La distance de Mahalanobis d'un échantillon se calcule comme ceci :

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (5.1)$$

où $x = (bis, tpn, tpo, \dots, r)$ est le vecteur contenant les variables d'entrée et $\mu = (\overline{bis}, \overline{tpn}, \overline{tpo}, \dots, \overline{r})$ le vecteur contenant les moyennes de chaque variables d'entrée de l'ensemble d'apprentissage, c'est le centre de masse. On remarquera que lorsque Σ est la matrice identité, le calcul de $D_M(\cdot)$ peut être ramené au calcul de la distance euclidienne entre le point x et le centre de masse de l'ensemble d'apprentissage μ .

Lors de chaque prédiction, BisPredict va calculer la distance de Mahalanobis du vecteur d'entrée du modèle et va avertir l'anesthésiste si l'échantillon est trop éloigné

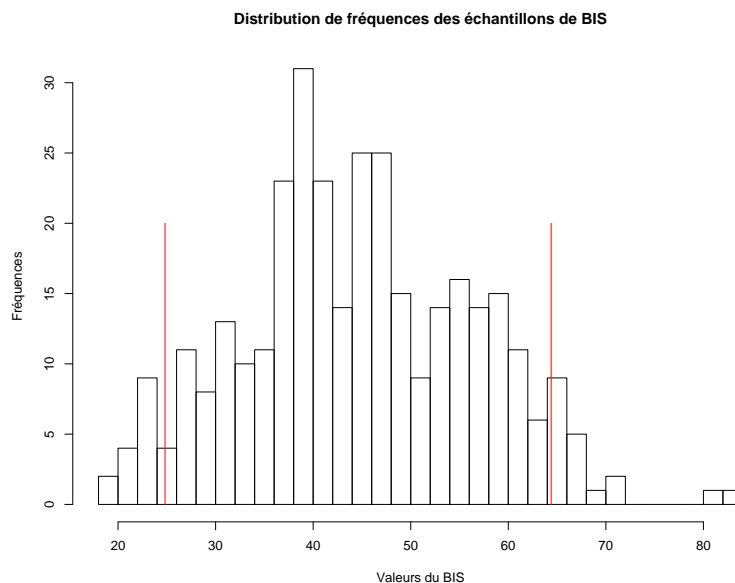


FIG. 5.4: Représentation des quantiles à 5% et 95% de la variable BIS de l'ensemble d'apprentissage.

des échantillons sur lesquels le modèle s'est entraîné. Une distance seuil a été fixée arbitrairement à 12.4 selon le quantile à 95% sur la distribution de fréquence des distances de Mahalanobis des échantillons de l'ensemble d'apprentissage D_N . Cette distribution des distances est illustrée sur la figure 5.5.

5.3 Analyse des risques

Il s'agit ici d'une version réadaptée et restreinte à l'utilisation de BisPredict de l'analyse de risques effectuée par Olivier Cailloux, chercheur responsable du développement d'ARAQ, concernant la librairie ARAQ. Je n'aborderai ici que les risques logiciels. Cependant l'analyse des risques au niveau médical est entrain d'être finalisée par le même chercheur.

Comme je l'avais mentionné avant, ARAQ et Itb95 sont groupés et forment une seule et même entité appelée *ARAQ-Itb*. On peut se servir d'ARAQ-Itb de deux manières différentes.

- Comme librairie : ARAQ-Itb est inclus dans un logiciel désirant récupérer des informations en provenance des appareils médicaux. Ici c'est le logiciel-pilote qui disposant d'une interface graphique utilise les fonctionnalités d'ARAQ-Itb en vue de communiquer avec les appareils. C'est dans ce contexte que je me sers d'ARAQ-Itb.

- Comme logiciel en tant que tel. Dans ce cas, il est utilisé directement par l'anesthésiste via son interface graphique, pour contrôler les injecteurs et/ou observer les moniteurs.

Dans le cadre de l'utilisation d'ARAQ-Itb en tant que librairie, la figure 5.6 illustre les relations entre les différentes entités. Le schéma présente trois couches aux fonctionnalités ascendantes. La couche de droite représente les appareils médicaux. Cette couche permet à l'anesthésiste de contrôler manuellement les injecteurs et de leur fournir des informations

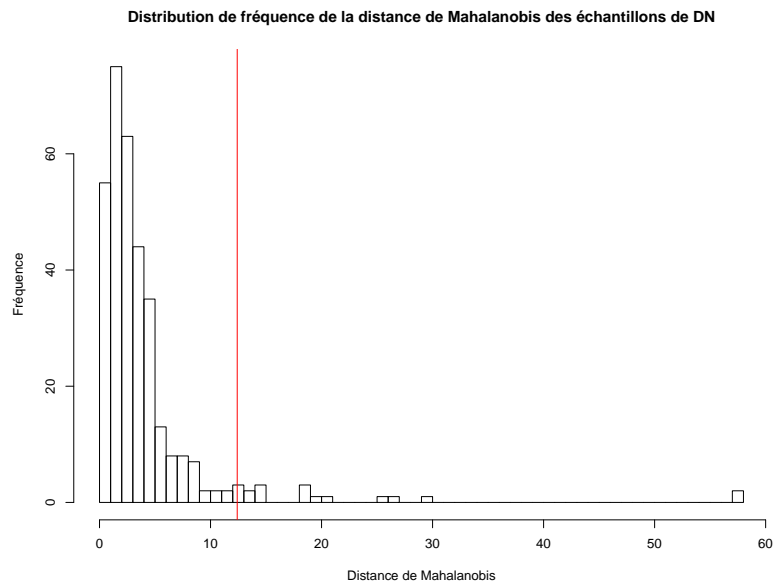


FIG. 5.5: 95% des échantillons présents dans D_N ont une distance de Mahalanobis inférieure à 12.40284.

médicales concernant l'état du patient. La seconde couche constituée d'ARAQ-Itb peut être subdivisée en deux sous-couches. Itb95 est un logiciel implémentant un protocole de communication avec les appareils médicaux. Il permet de lire les informations envoyées par les appareils mais aussi de donner des ordres aux injecteurs. La sous-couche ARAQ implémente un protocole de communication avec Itb95. ARAQ fait appel aux fonctionnalités d'Itb95 et constitue une sur-couche d'Itb95, permettant l'ajout de fonctionnalités, l'utilisation d'une interface graphique différente de celle d'Itb95, ou, lorsqu'il est utilisé comme une librairie, une facilité d'utilisation accrue. BisPredict se situe dans la dernière couche. Il va communiquer avec les appareils médicaux grâce à l'entité ARAQ-Itb.

5.3.1 Tests à réaliser

Au vu du contexte d'utilisation de BisPredict, il a fallu effectuer un large panel de tests afin de s'assurer que le fonctionnement de BisPredict n'interférait en rien avec les appareils médicaux. Il convient d'effectuer les tests décrits ci-après lors de l'installation de BisPredict sur un nouveau chariot ou lorsqu'une mise à jour de la librairie ARAQ est effectuée.

- Tests d'injection : vérifier que le comportement des injecteurs est bien celui attendu par rapport aux ordres donnés par l'anesthésiste. Ce test doit être fait pour toutes les fonctionnalités de BisPredict. Dans ce cas-ci les tests ont été réalisés pendant les deux phases et n'ont pas donné signe de dysfonctionnement.
- Validation de la communication avec les appareils : il faut s'assurer que les données provenant des appareils sont les mêmes que celles reçues par BisPredict. Les tests ont été validés grâce à l'interface de debug. Pour chaque modification d'un target de Propofol/Remifentanyl, les valeurs ont été comparées. Le BIS a aussi été testé. Le

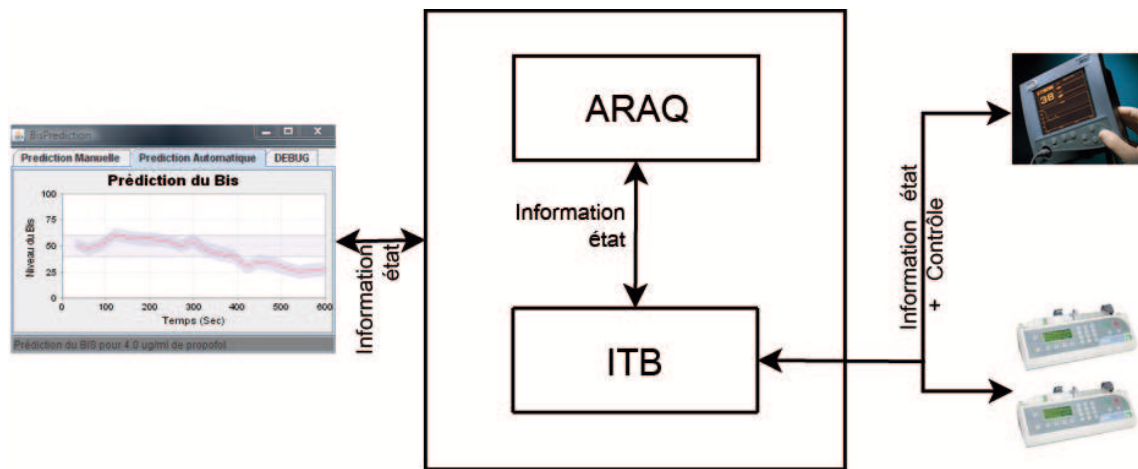


FIG. 5.6: Illustration de l'utilisation d'ARAQ comme une librairie. Il est utilisé conjointement à BisPredict, ce dernier fournissant une interface homme-machine et interagissant avec ARAQ-Itb pour la communication avec les appareils médicaux.

résultat des tests n'a pas non plus révélé de dysfonctionnements.

- Validation de l'affichage : avant l'utilisation de BisPredict, il faut s'assurer que l'utilisateur interprète les données provenant des appareils médicaux rapportées par le logiciel-pilote de la même façon qu'il les interprète sur l'interface homme-machine des appareils. Par exemple la courbe de BIS réel affichée lors de la prédiction automatique est un répliat de l'évolution du BIS affichée sur le moniteur de BIS. Ou encore que la courbe du BIS prédit ne reflète pas forcément la véritable évolution du BIS réel.

Les tests énoncés précédemment ont pour but de s'assurer que BisPredict n'interfère pas avec les signaux transmis aux appareils médicaux. Cependant, d'autres tests ont été effectués afin de s'assurer que le raffinement des données a été correctement effectué. Il a ainsi fallu vérifier que la construction des variables additionnelles, au terme de la phase 1 et des autres variables pendant la phase 2, a été correctement effectuée. Ici encore, la fenêtre de debug s'est révélée d'une aide précieuse.

5.3.2 Hypothèses de travail

Afin de minimiser les risques liés à l'utilisation de BisPredict et d'ARAQ-Itb il convient de respecter un ensemble d'hypothèses de travail. Ces hypothèses sont :

- Utilisation uniquement en bloc opératoire : les situations où le patient se trouve sans anesthésiste à ses côtés pendant une durée prolongée sont à proscrire absolument.
- Utilisation par un anesthésiste expérimenté : il va de soi qu'en cas de défaillance générale du système informatique, l'anesthésiste doit être capable de reprendre le contrôle de l'anesthésie de manière manuelle.
- Les données doivent être dupliquées : BisPredict affiche l'évolution du BIS réel à l'écran. Cependant cette information provient de couches inférieures. Il se peut qu'en remontant les couches, l'intégrité de l'information ait été compromise. C'est pour cette raison qu'on ne peut utiliser BisPredict que dans le cas où une source de

données *fiabes* affiche les données du patient.

- BisPredict ne doit être relié à un injecteur que si l'injecteur possède une interface homme-machine fiable, ce qui permet à l'anesthésiste de reprendre le contrôle de l'injecteur manuellement en cas de dysfonctionnement du système informatique.

5.3.3 Risques liés à BisPredict

Hormis les risques liés aux dysfonctionnements logiciels, un dernier type de risque est à mettre en évidence. BisPredict est un logiciel dont le développement est toujours en phase expérimentale. Il fournit une aide à l'utilisateur par les prédictions de l'état de conscience qu'il réalise. A ce stade, les prédictions délivrées par le logiciel ne peuvent pas encore être considérées complètement fiables. Pour cette raison, il est important de mettre l'anesthésiste en garde contre les erreurs de jugement qu'il pourrait commettre sous l'influence de la prédiction de l'évolution de l'état de conscience du patient.

5.4 Déploiement

Cette section a pour objectif de détailler les différentes phases du déploiement de BisPredict sur les chariots d'anesthésie de l'hôpital Erasme. J'ai consacré plusieurs journées à effectuer de nombreux tests afin que BisPredict s'intègre parfaitement avec ARAQ-Itb.

5.4.1 Installation

L'installation manuelle étant assez fastidieuse due à la nécessité de modifier certains fichiers de configuration d'Itb95 ainsi que plusieurs variables d'environnement et afin d'automatiser ce processus, j'ai réalisé un script permettant d'installer de façon automatique tous les composants nécessaires au fonctionnement de BisPredict. Malheureusement, il faut toujours éditer un fichier de configuration crypté dans Itb95. Le processus d'installation est détaillé en annexe B.

5.4.2 Cas Test

Pour s'assurer de la bonne cohabitation entre BisPredict et ARAQ-Itb, j'ai effectué de nombreux tests. Je résume sommairement chacun des tests réalisés ainsi que les résultats que j'ai obtenus.

Le déploiement de BisPredict s'est déroulé en plusieurs phases.

- Installation, configuration manuelle d'ARAQ-Itb, tests communication avec appareils médicaux (Propofol et Remifentanyl). A ce stade, BisPredict ne récupérait pas encore les signaux de BIS. Les tests étaient concluants, aucun problème n'a été rencontré.
- Installation, configuration manuelle d'ARAQ-Itb, tests avec appareils médicaux (Propofol, Remifentanyl et BIS). Une mise à jour d'ARAQ a été réalisée, j'ai donc dû modifier mon code source afin de prendre en compte les modifications effectuées. Les tests de communication ont été réeffectués à nouveau. La construction des différentes variables a aussi été testée. Aucun problème n'a été remarqué.

- Installation, configuration manuelle d'ARAQ-Itb, tests avec appareils médicaux (Propofol, Remifentanil et BIS). Un test-set définitif était encore à faire avant d'être sûr du bon fonctionnement de BisPredict. Il s'est avéré que cette journée de tests s'est soldée par un échec. Pour la première fois j'ai rencontré des problèmes de communication avec les appareils médicaux. Pour communiquer avec les logiciels client, Itb95 utilise des fichiers partagés. Dans certains cas, l'en-tête de ces fichiers était manquant. Cela a permis à Olivier Cailloux de modifier ARAQ en simulant la présence des en-têtes lorsque ceux-ci sont absents. Aucun test concernant BisPredict n'a pu être exécuté ce jour. Cependant les tests ont permis de reproduire le problème de manière quasi systématique. Les en-têtes sont absents lorsqu'Itb95 lance automatiquement les outils de contrôle des appareils médicaux lors d'un *startSession*.
- Installation + configuration manuelle d'ARAQ-Itb + tests avec appareils médicaux (Propofol + Remifentanil + BIS)
 - La vérification de la correction du problème des en-têtes manquants a été un succès.
 - Test-set effectué : les débits affichés dans Itb95 sont les mêmes que ceux indiqués sur les pousse-seringues. Les données concernant le BIS ainsi que les concentrations de Propofol et de Remifentanil sont correctement transmises à BisPredict. La construction des variables lors des deux phases s'est faite correctement.
 - Un script d'automatisation de l'installation a été créé. Cependant il faut encore modifier un fichier de configuration via Itb95. Mais il réduit considérablement le temps de déploiement. Voir annexe B pour le détail.
- BisPredict est opérationnel. Aucun problème n'a été remarqué lors de cette dernière phase de tests.

5.5 Validation du modèle face aux données collectées en conditions réelles

Après que le logiciel BisPredict ait été consciencieusement testé et que l'analyse des risques ait été présentée et acceptée par l'anesthésiste en charge de réaliser les tests, BisPredict a été utilisé en conditions réelles pendant 3 anesthésies. Les données collectées durant ces opérations m'ont permis de confronter les prédictions du BIS réalisées par les quatre modèles construits dans le chapitre précédent aux valeurs réelles du BIS pendant l'opération.

A chaque changement du target de Propofol, le vecteur contenant les variables d'entrée du modèle ainsi que la prédiction retournée par le modèle ont été sauvés dans un fichier. A partir de cet instant, les changements du BIS réel et le runtime ont été sauvés dans un autre fichier. Cela m'a permis de réaliser un script R qui calcule l'estimation de l'erreur de généralisation et qui affiche le graphique superposant la courbe du BIS réel sur la courbe de BIS prédite.

Grâce au fichier contenant les vecteurs d'entrée ayant servis à réaliser les prédictions, j'ai également pu tester les performances des trois autres modèles.

Dans cette section, je vais réaliser la validation statistique des quatre modèles ainsi qu'une validation qualitative du modèle linéaire avec variables additionnelles qui a été

embarqué par BisPredict lors des tests en conditions réelles.

5.5.1 validation statistique

Lors des tests en conditions réelles, BisPredict a collecté des données qui m’ont permis de tester les autres modèles comme s’ils avaient été embarqués dans BisPredict.

Les graphiques illustrant la comparaison des prédictions réalisées par les quatre modèles sont présentés dans l’annexe C.

Afin de comparer les prédictions réalisées par les modèles aux valeurs du BIS réel observé durant l’anesthésie, je vais dans le cas de Λ^{Lin} utiliser le critère suivant :

$$R_{ts}(\Lambda^{Lin}) = \frac{1}{N_{ts}} \sum_{i=1}^{N_{ts}} \sqrt{(BIS_i - \widehat{BIS}_i)^2} \quad (5.2)$$

qui est une mesure de l’erreur commise par Λ^{Lin} lorsque BIS_i est la valeur du BIS réel observé pendant l’anesthésie et \widehat{BIS}_i la valeur du bis prédit par Λ^{Lin} . Ce critère se décline de la même manière aux trois autres modèles.

Le modèle prédit une valeur de BIS toutes les 30 secondes. On sait que la valeur du BIS est corrélée aux concentrations de drogues injectées dans le patient. Cependant, d’autres facteurs influencent cette valeur. L’utilisation du bistouri électrique influence fortement les mesures du BIS de par le bruit qu’il induit sur le canal. C’est pour cette raison que le BIS_i ayant servi à la validation empirique a été calculé sur un intervalle de temps [t-10sec, t+10 sec] en prenant la moyenne des valeurs du BIS observé dans cet intervalle.

En tenant compte de ces considérations, le tableau 5.3 illustre le résultat des calculs de R_{ts} appliqué à chacun des quatre modèles.

Modèle	R_{ts}
Λ^{Lin}	7.03
Λ^{Lazy}	6.75
Λ^{LinAdd}	7.03
$\Lambda^{LazyAdd}$	6.84

TAB. 5.3: Mesure de l’erreur de prédiction commise par les modèles en conditions réelles.

Au vu du tableau 5.3 les calculs de R_{ts} semblent confirmer les résultats théoriques dans le sens où les modèles LL sont plus performants que les deux modèles linéaires. On pourrait interpréter ce résultat comme le fait que le problème de prédiction du BIS est un problème non linéaire.

5.5.2 Validation qualitative de BisPredict

Cette section a pour objectif d’évaluer le bon fonctionnement de BisPredict comme logiciel embarqué en salle d’opération ainsi que la pertinence des informations fournies par le logiciel à l’anesthésiste tout au long des trois anesthésies. En plus, grâce à l’interaction que j’ai eue avec l’anesthésiste Djamal Ghoundiwal et le Docteur Luc Barvais qui

CHAPITRE 5. BISPREDICT, UN OUTIL D'AIDE À LA DÉCISION EN ANESTHÉSIE

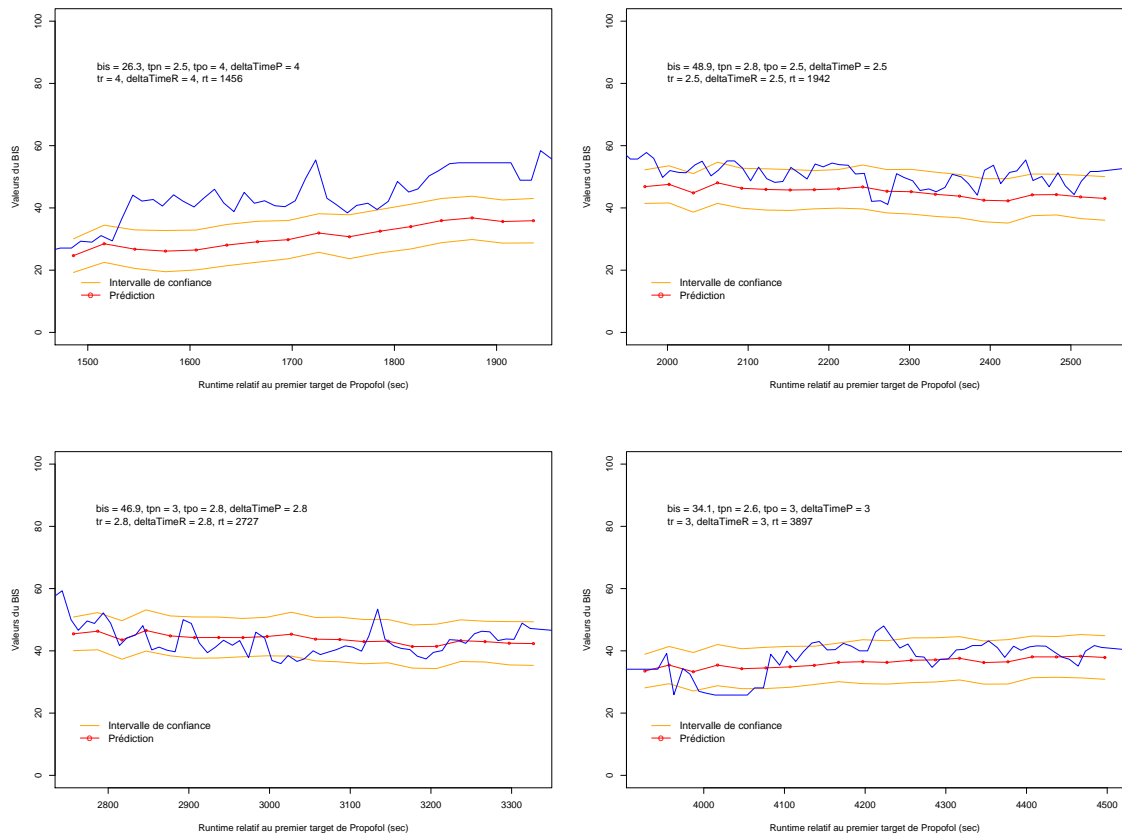


FIG. 5.7: Partie 1/2. Graphique comparatif, tiré d'un test en conditions réelles, du BIS réel et du BIS prédit par le modèle embarqué dans BisPredict en fonction du temps. Chaque graphique correspond à une prédiction réalisée lors d'un changement du target de Propofol. La courbe rouge représente la prédiction, tandis que les courbes orange délimitent l'intervalle de confiance par rapport à l'estimation de l'erreur leave-one-out de la prédiction. Le BIS réel est représenté par la courbe bleue.

m'ont encadré lors de mes expériences en bloc opératoire, j'ai pu obtenir des informations concernant l'aide fournie par les logiciels informatiques embarqués en salle d'opération. J'ai retenu leurs impressions face aux prédictions réalisées par BisPredict. Ces impressions et remarques figurent aussi dans la suite de cette section.

Cohabitation de BisPredict avec Itb95

J'ai eu l'occasion de tester BisPredict pendant 3 anesthésies. Le résultat de l'utilisation de BisPredict lors de la première anesthésie n'a pas abouti à cause d'une erreur dans BisPredict. Les données ont été correctement collectées et analysées pendant la phase d'induction, cependant une erreur est survenue lors de la première prédiction manuelle. Une exception concernant le format d'encodage de la concentration a été capturée. C'est une virgule au lieu d'un point, dans le nombre réel représentant la concentration de Propofol, qui a déclenché l'exception. Le traitement de l'exception a mis fin proprement

CHAPITRE 5. BISPREDICT, UN OUTIL D'AIDE À LA DÉCISION EN ANESTHÉSIE

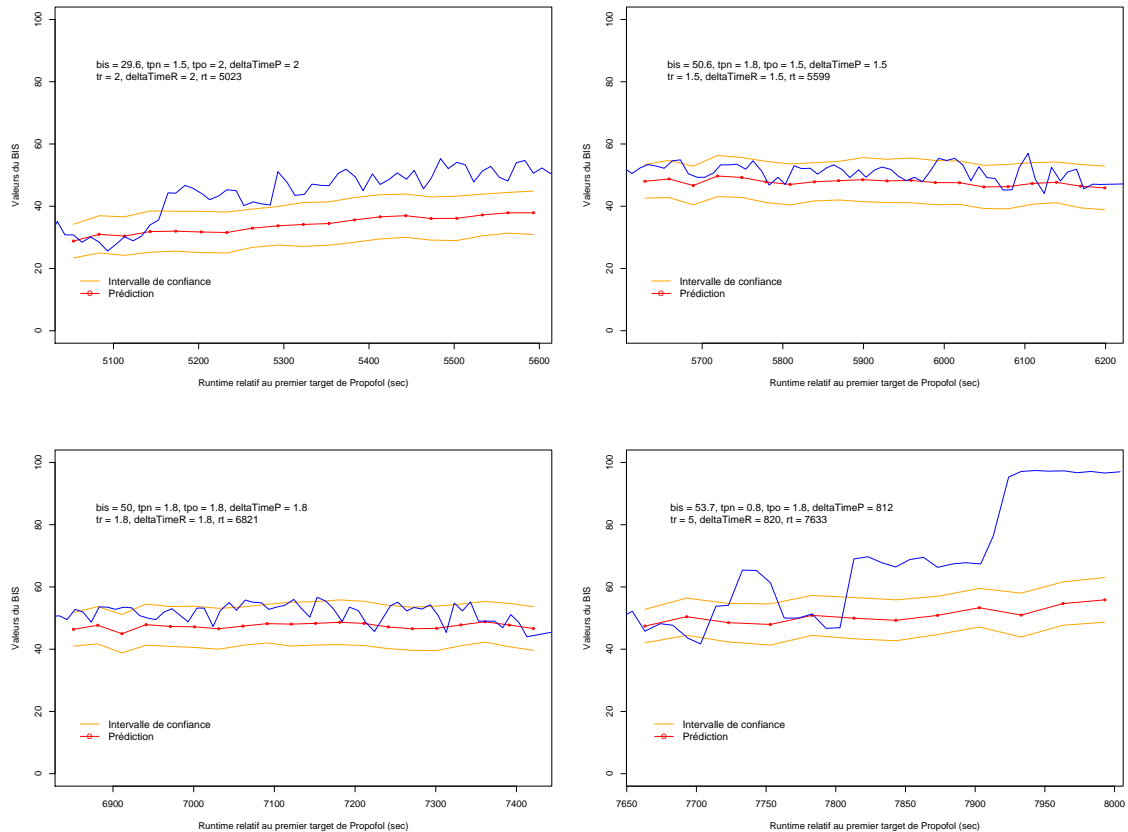


FIG. 5.8: Partie 2/2. Graphique comparatif, tiré d'un test en conditions réelles, du BIS réel et du BIS prédit par le modèle embarqué dans BisPredict en fonction du temps. Chaque graphique correspond à une prédiction réalisée lors d'un changement du target de Propofol. La courbe rouge représente la prédiction, tandis que les courbes orange délimitent l'intervalle de confiance par rapport à l'estimation de l'erreur leave-one-out de la prédiction. Le BIS réel est représenté par la courbe bleue. Le dernier graphique (en bas à droite) représente la phase de réveil. La remontée vertigineuse du BIS entre les runtimes 7900 et 7950 est liée au fait que l'anesthésiste a interagi de manière physique avec le patient. Ce phénomène représente encore une difficulté quant au fait de prédire avec exactitude le moment de réveil.

CHAPITRE 5. BISPREDICT, UN OUTIL D'AIDE À LA DÉCISION EN ANESTHÉSIE

à la communication de BisPredict avec les appareils médicaux. L'anesthésiste a été averti de l'exception. Cet incident n'a eu aucun effet néfaste sur le fonctionnement des appareils médicaux ni sur celui d'Itb95. La solution qui a été apportée à ce problème est de corriger la virgule en un point. Si toutefois aucun format n'est détectable, on ne ferme plus l'application, et on ne réalisera pas de prédiction pour la donnée entrée par l'utilisateur. Un message d'erreur s'affichera dans le panel des messages.

Durant les deux autres anesthésies, BisPredict et Itb95 ont fonctionné en parfaite harmonie. Aucun dysfonctionnement n'a été remarqué dans aucun fichier de log (aussi bien au niveau d'ARAQ que de BisPredict).

Le premier objectif des tests en conditions réelles, à savoir s'assurer du fait de la bonne cohabitation de BisPredict et d'Itb95, a donc été atteint en sachant que les hypothèses d'utilisation de BisPredict et d'ARAQ ont été respectées.

Pertinence des prédictions lors de l'anesthésie

Comme l'a montré la figure 5.1 les prédictions se sont montrées assez proches de la réalité durant la phase de maintien. Pour de petites variations de la concentration de Propofol, l'allure de la prédiction se rapprochait assez bien de l'évolution du BIS réel. Cependant, lorsque la variation de concentration de Propofol tendait à augmenter, la précision du modèle laissait à désirer. La remontée ou la descente du BIS prédit avait le même sens que le BIS réel, mais la variation n'était pas suffisamment agressive. Ce manque d'agressivité est illustré par le premier graphique de la figure 5.7 et le premier de la figure 5.8.

Outre le fait de voir la prédiction de l'évolution du BIS, l'anesthésiste souhaiterait pouvoir prédire l'instant de réveil du patient. Pour cela, il coupe la perfusion de Propofol. La concentration de Propofol passe par exemple de $2\mu\text{g/ml}$ à $0\mu\text{g/ml}$ pour un BIS courant de 40. Il s'est avéré dans les deux cas testés que la prédiction indiquait que le BIS prendrait des valeurs avoisinant 60, cela 10 minutes après l'arrêt de la perfusion. En réalité, il n'a fallu qu'environ trois à quatre minutes avant que le patient ne se réveille avec un BIS avoisinant 90. Le réveil est illustré sur le dernier graphique de la figure 5.8.

Ce manque d'agressivité du modèle est lié aux informations contenues dans l'ensemble d'apprentissage. En effet, la majorité des échantillons constituant l'ensemble d'apprentissage a été créé durant la phase de maintien où le BIS varie entre 40 et 60. Le modèle n'a donc pas été suffisamment entraîné sur des échantillons représentant le contexte du réveil par exemple. La figure 5.4 montre la distribution de fréquence de la variable BIS au sein de l'ensemble d'apprentissage. On peut dire qu'il n'y a aucun voire très peu d'échantillons où le patient a un BIS supérieur à 80. A nouveau ceci est une évidence expliquant le manque de performance du modèle à prédire dans la phase de réveil.

Concernant la trop faible réactivité du modèle face aux gros changements de concentration de Propofol, on peut voir sur la figure 5.9 montrant la distribution de fréquence des variations de Propofol qu'il y a très peu d'échantillons représentant des variations supérieures à $1\mu\text{g/ml}$ en valeur absolue.

Compte tenu des informations présentes dans l'ensemble d'apprentissage, il est donc normal que le modèle ait été aussi peu performant lors de la phase de réveil et lorsque

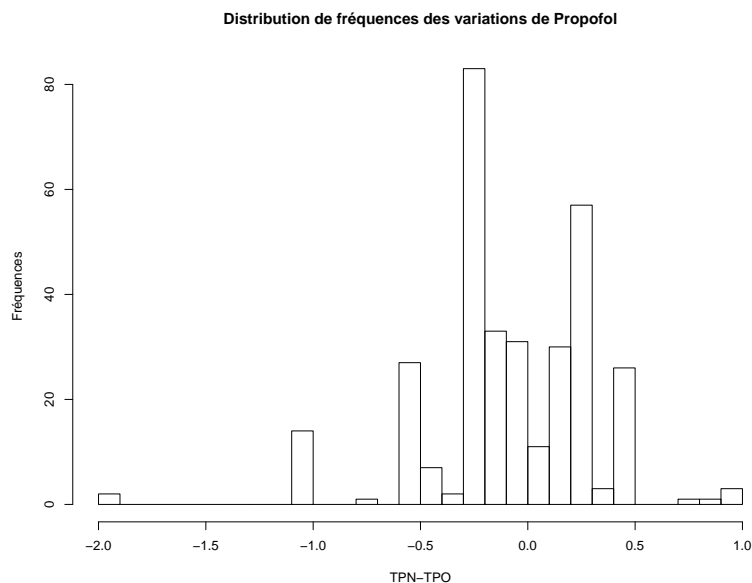


FIG. 5.9: Graphique illustrant la distribution de fréquences des variations de Propofol. On peut voir qu'il n'y a aucun échantillon où la variation de Propofol est strictement supérieure à 1.

l'anesthésiste a fait varier la concentration de Propofol de manière conséquente. Pour pallier à cette carence, il faudrait regarnir l'ensemble d'apprentissage avec des échantillons caractéristiques de ces évènements mais ils sont par nature plus rares.

Autres remarques

Une dérivation de l'utilisation de BisPredict est la mise en évidence d'une concentration de Propofol à injecter afin que le BIS réel atteigne un niveau souhaité. Ainsi, après avoir soumis plusieurs concentrations de Propofol fictives à BisPredict, l'anesthésiste, via l'interface de prédictions manuelles, sera en mesure de choisir la concentration pour laquelle le modèle a prédit un BIS suffisamment proche du niveau souhaité par l'anesthésiste.

A terme, il serait intéressant d'ajouter une série de fonctionnalités à BisPredict. On pourrait par exemple concevoir un système qui avertit l'anesthésiste lorsque le BIS du patient sort de l'intervalle de confiance de la prédiction.

Une autre remarque a été émise quant au fait de l'affichage des valeurs du signal BIS lorsque l'index de qualité de ce signal était trop faible. Dans ce cas, il serait souhaitable de ne pas afficher ces valeurs ou encore d'avertir l'anesthésiste par un message adéquat.

Implémenter la superposition du BIS réel sur l'onglet de prédiction manuelle a aussi été suggéré.

Les modèles ont été entraînés sur un ensemble d'apprentissage mélangeant toutes sortes de chirurgies. Or, certains types de chirurgie nécessitent de maintenir le patient dans un état de conscience plus bas que la normale. Dans ce contexte il pourrait être pertinent de créer des datasets propres à chaque type de chirurgie.

Une dernière remarque a aussi été émise sur le fait que l'analgésie joue elle aussi un rôle important sur l'état de conscience. A ce titre il serait intéressant d'effectuer une prédiction automatique lors d'un changement du target de Remifentanyl. Toujours dans le contexte de prédiction, il a été demandé que BisPredict effectue des prédictions automatiques à intervalles réguliers, de sorte qu'il y ait une prévision permanente de l'état de conscience sur l'écran de BisPredict.

5.6 Conclusion des tests

Pour conclure ce chapitre, je peux dire que les résultats obtenus sont encourageants. Au niveau de son fonctionnement d'un point de vue logiciel, BisPredict s'est très bien comporté en conditions réelles et aucun dysfonctionnement avec les appareils médicaux n'a été à déplorer.

La collecte des données en arrière plan a permis de réaliser une étude plus approfondie des différents modèles de prédictions construits dans le chapitre précédent. Ce système de collecte des données pourrait être utilisé afin d'étoffer l'ensemble d'apprentissage dans les phases où il manque cruellement d'informations.

Le second point important est que les prédictions effectuées par BisPredict se sont montrées assez proches de la réalité durant la phase de maintien de l'anesthésie. On a cependant remarqué que les prédictions avaient tendance à sous-estimer la réalité.

Des lacunes importantes apparaissent lors des prédictions pendant la phase de réveil et lors des changements abruptes de la concentration de Propofol. Ceci est dû au manque d'échantillons correspondant à ces phases dans l'ensemble d'apprentissage.

Chapitre 6

Conclusions et perspectives

J'ai réalisé mon mémoire en collaboration avec le service d'anesthésie/réanimation de l'hôpital Erasme. Mon étude a porté sur la conception de modèles de prédictions concernant la profondeur de l'anesthésie suite à une modification de la concentration de Propofol. Afin de réaliser ces prédictions, j'ai, sur base d'un ensemble d'apprentissage, entraîné quatre modèles de prédictions. A partir de modèles déjà développés par Olivier Caelen dans le cadre d'un projet First intitulé *Data mining prédictif en anesthésie intraveineuse informatisée en vue de certification* j'ai étudié le gain obtenu en qualité de prédiction par l'ajout de 10 nouvelles variables d'entrées aux modèles. Ces variables reflètent la manière dont le patient réagit aux drogues de manière générale.

J'ai ensuite étudié les performances de modèles de type linéaire et Lazy en tenant compte de 10 nouvelles variables additionnelles. Il s'est avéré que les modèles Lazy ont de manière théorique présenté de meilleures performances que les modèles de type linéaire. Ceci pourrait être un argument en faveur du fait que la relation qui associe le BIS aux variations de Propofol n'est pas linéaire. De plus, le processus de sélection de variables a clairement mis en évidence l'utilité de ces nouvelles variables dans leur apport d'informations pour la prédiction.

Afin de tester les performances de ces modèles en conditions réelles, j'ai développé un logiciel, BisPredict, qui a fourni aux anesthésistes en charge une aide à la décision pendant 3 opérations. Le premier objectif de ces tests en conditions réelles a été de montrer que BisPredict était stable et n'interférait pas avec les appareils médicaux. Cet objectif a clairement été atteint car aucun dysfonctionnement au niveau des appareils médicaux n'a été à déplorer.

Le second objectif était d'évaluer la pertinence des prédictions délivrées par le modèle (linéaire avec variables additionnelles) embarqué dans BisPredict. Lors des tests en conditions réelles, il s'est avéré que les prédictions en per-opératoire se sont montrées assez précises bien que le modèle tendait à sous-estimer constamment la valeur réelle du BIS. Malheureusement le modèle a montré ses limites lorsque de grandes variations des concentrations de Propofol étaient effectuées. La prédiction de l'instant de réveil à l'arrêt des perfusions a elle aussi été défailante. Ces dysfonctionnements du modèle de prédiction sont dus au fait d'un manque d'échantillons d'apprentissage caractérisant ces évènements.

Afin de se donner une idée des résultats obtenus par les différents modèles, l'annexe C

CHAPITRE 6. CONCLUSIONS ET PERSPECTIVES

reprend les graphiques comparant les prévisions des différents modèles face à un cas test issu d'une opération pendant laquelle BisPredict a été utilisé.

Au vu des résultats obtenus lors des tests en conditions réelles, je pense qu'il est intéressant de continuer les recherches dans la voie ouverte par mon mémoire. Une manière de poursuivre ce travail serait de continuer de garnir l'ensemble d'apprentissage en insistant sur des échantillons collectés dans les contextes où le modèle est actuellement peu performant. Une méthode pour y parvenir serait d'étudier des techniques d'apprentissage par renforcement tel que le *recursive least-squares* [8]. Je pense également qu'il serait opportun de créer plusieurs ensembles d'apprentissage correspondant à des types de chirurgie distincts car la profondeur de l'anesthésie ainsi que les concentrations de drogues peuvent considérablement varier d'un type d'opération à l'autre.

Annexe A

Développements de calculs

Régression linéaire multivariée

Considérons un phénomène où on suppose qu'il existe une relation linéaire entre les variables d'entrée $x \in \mathbb{R}^n$ et la variable de sortie $y \in \mathbb{R}$ de telle sorte que

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + w \quad (\text{A.1})$$

où w est une variable aléatoire de moyenne nulle et de variance constante σ_w^2 et les β_i , $i = 0, \dots, n$ sont les *coefficients de régression*.

Par la suite je manipulerai les expressions sous forme matricielle. Ainsi la formule A.1 devient :

$$y = x^T \beta + w \quad (\text{A.2})$$

où

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \quad (\text{A.3})$$

Considérons à présent que nous avons observé N réalisations $\{\langle x_i, y_i \rangle : i = 1, \dots, N\}$ où $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ du processus stochastique A.1. La formule A.2 peut se réécrire sous la forme suivante :

$$y = X\beta + w \quad (\text{A.4})$$

où

ANNEXE A. DÉVELOPPEMENTS DE CALCULS

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \dots & x_{Nn} \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \quad w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} \quad (\text{A.5})$$

Le but de la méthode des *least-squares* est de trouver une estimation $\hat{\beta}$ des coefficients de régression β afin de minimiser le risque empirique R_{emp} :

$$\frac{\partial}{\partial \hat{\beta}} [(y - X\hat{\beta})^T (y - X\hat{\beta})] = 0 \quad (\text{A.6})$$

où

$$(y - X\hat{\beta})^T (y - X\hat{\beta}) = \sum_{i=1}^n (y_i - x_i^T \hat{\beta})^2 = R_{emp}(\hat{\beta}) \quad (\text{A.7})$$

Calcul de la minimisation du risque empirique afin de trouver le $\hat{\beta}$ optimal :

$$\begin{aligned} \frac{\partial}{\partial \hat{\beta}_i} R_{emp}(\hat{\beta}) &= \frac{\partial}{\partial \hat{\beta}_i} [(y^T - \hat{\beta}^T X^T)(y - X\hat{\beta})] \\ &= 2(\hat{\beta}^T X^T - y^T)x_i \\ &\text{\small } x_i \text{ est la } i^{\text{eme}} \text{ colonne de } X \end{aligned}$$

$$\begin{aligned} \left(\frac{\partial}{\partial \hat{\beta}} R_{emp}(\hat{\beta}) \right)^T &= \left(\frac{\partial}{\partial \hat{\beta}_1} R_{emp}(\hat{\beta}), \frac{\partial}{\partial \hat{\beta}_2} R_{emp}(\hat{\beta}), \dots, \frac{\partial}{\partial \hat{\beta}_n} R_{emp}(\hat{\beta}) \right) \\ &= 2(\hat{\beta}^T X^T - y^T)X \end{aligned}$$

Il reste à annuler la dérivée de l'erreur empirique afin de trouver le $\hat{\beta}$ optimal :

$$\begin{aligned} \left(\frac{\partial}{\partial \hat{\beta}} R_{emp}(\hat{\beta}) \right)^T &= 0 \\ 2(\hat{\beta}^T X^T - y^T)X &= 0 \\ \hat{\beta}^T X^T - y^T &= 0 \\ X\hat{\beta} - y &= 0 \\ X\hat{\beta} &= y \\ X^T X\hat{\beta} &= X^T y \\ (X^T X)^{-1} X\hat{\beta} &= (X^T X)^{-1} X^T y \\ \hat{\beta} &= (X^T X)^{-1} X^T y \end{aligned}$$

ANNEXE A. DÉVELOPPEMENTS DE CALCULS

Le résultat de l'annulation de la dérivée de l'erreur empirique

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (\text{A.8})$$

est la solution du problème d'optimisation des paramètres $\hat{\beta}$ par la méthode least-squares. $\hat{\beta}$ contient les paramètres optimaux du modèle linéaire pour un D_N donné.

Décomposition Biais/Variance du MSE

$$\begin{aligned} \text{MSE}(x) &= E_{D_N, y} [(y - h(x, \alpha_N))^2] \\ &= E_y [(y - E_y[y|x])^2] + E [(h(x, \alpha_N) - E_y[y|x])^2] \\ &= \sigma_w^2 + E_{D_N} [(h(x, \alpha_N) - E_y[y|x])^2] \\ &= \sigma_w^2 && \text{bruit} \\ &+ (E_{D_N}[h(x, \alpha_N)] - E_y[y|x])^2 && \text{biais au carre} \\ &+ E_{D_N} [(h(x, \alpha_N) - E_{D_N}[h(x, \alpha_N)])^2] && \text{variance} \end{aligned}$$

Annexe B

Manuel de déploiement de BisPredict

Cette annexe décrit la procédure d'installation d'ARAQ et de BisPredict.

Déploiement manuel

Configuration du serveur de communication d'Itb95

Pour commencer, il faut dire à Itb95 de démarrer son serveur de communication (ItbLogServer) lorsqu'il démarre. Pour ce faire, il faut modifier le fichier *itb95.rc* qui se trouve dans le répertoire racine d'Itb95.

Lancer Itb95 et choisir le programme CryptedFile editor. L'utilisateur est Administrator. On ouvre le fichier en tant qu'Administrator (par défaut) ensuite on ajoute les deux lignes suivantes avant la balise END de fin de fichier :

```
ExportState "true"  
ExportDir "C:\PROGRA~1\BisPredict\workDir"
```

Attention, ici il faut utiliser les formats de chemins courts. Sauver le fichier en utilisant le nom Administrator (par défaut).

Installation d'ARAQ

Créer un répertoire dans un emplacement quelconque dans lequel on copiera les fichiers contenus dans le répertoire de la librairie ARAQ.

A présent il va falloir ajouter à la variable d'environnement *path* le chemin vers le répertoire *native* de la librairie. Après quoi il faut créer une nouvelle variable d'environnement qu'on nommera *toolbox.server.path* qui pointera sur *native/itb_req/fmServer.exe* qui est le logiciel permettant de communiquer avec ItbLogServer.

Maintenant il faut copier les fichiers de configurations d'ARAQ présents dans le dossier *araqAppData* dans le répertoire

```
C:\Users\%USERNAME%\AppData\Araq
```

ANNEXE B. MANUEL DE DÉPLOIEMENT DE BISPREDICT

Ensuite il faut modifier le fichier ARAQ.properties dans le répertoire créé précédemment afin de lui indiquer où se trouve le *workDir* d'Itb95 ainsi que chemin vers le *fmServer.exe* d'ARAQ. Pour ce faire, il faut modifier les deux entrées suivantes :

```
topicsComServer.workPath = C:\\Program Files\\BisPredict\\workDir
topicsComServer.dependantLibsPath
    =C:\\Program Files\\BisPredict\\Araq\\lib\\native\\itb_req
```

Script d'installation automatique d'ARAQ et BisPredict

```
rem - Création des répertoires qui vont accueillir BisPredict et ARAQ.
rem - Le workDir d'Itb95 est dans le répertoire BisPredict.
```

```
mkdir C:\\Program Files\\BisPredict
mkdir C:\\Program Files\\BisPredict\\workDir
mkdir C:\\Program Files\\BisPredict\\Araq
```

```
rem -----
```

```
rem - Copie des programmes dans leur répertoire d'installation
```

```
xcopy /S Araq C:\\Program Files\\BisPredict\\Araq
xcopy /S BisPredict C:\\Program Files\\BisPredict
```

```
rem -----
```

```
rem - Installation des fichiers de configurations d'Araq.
rem - Ils ont été pré-configurés de sorte à ne rien devoir y modifier.
```

```
mkdir C:\\Users\\%USERNAME%\\AppData\\Araq
xcopy /S araqAppData C:\\Users\\%USERNAME%\\AppData\\Araq
```

```
rem -----
```

```
rem - Modification des variables d'environnement.
rem - MAJ de path : afin de donner l'emplacement de TopicsComClientWrapper.dll
rem - Création de toolbox.server.path
```

```
set path=%path%;C:\\Program Files\\BisPredict\\Araq\\lib\\native
set toolbox.server.path=C:\\PROGRA~1\\BisPredict\\Araq\\lib\\native\\itb_req\\fmServer.exe
```

```
rem -----
```

Après quoi il reste à modifier le fichier *itb95.rc* se trouvant dans le répertoire racine d'Itb95. Pour ce faire, il faut lancer Itb95 et choisir le programme CryptedFile editor. L'utilisateur est Administrator et le Password est *****. On ouvre le fichier en tant qu'Administrator (par défaut) ensuite on ajoute les deux lignes suivantes avant la balise END

ANNEXE B. MANUEL DE DÉPLOIEMENT DE BISPREDICT

de fin de fichier :

```
ExportState "true"  
ExportDir "C:\Program Files\BisPredict\workDir"
```

Sauver le fichier en utilisant le nom Administrator (par défaut).

Exécution de BisPredict

Lancer Itb95, choisir le type de chirurgie, entrer les données du patient et attendre que les fenêtres de monitoring soient ouvertes. Exécuter BisPredict.

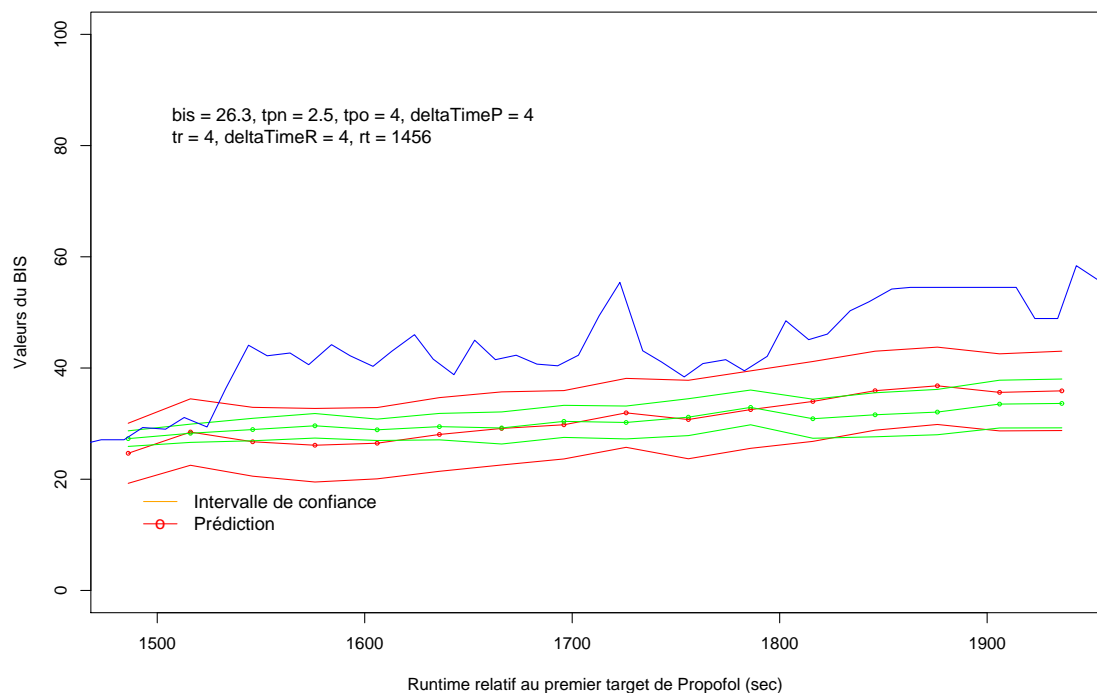
En pratique on pourra exécuter BisPredict dès que la console ItbLogServer aura été démarrée automatiquement par Itb.

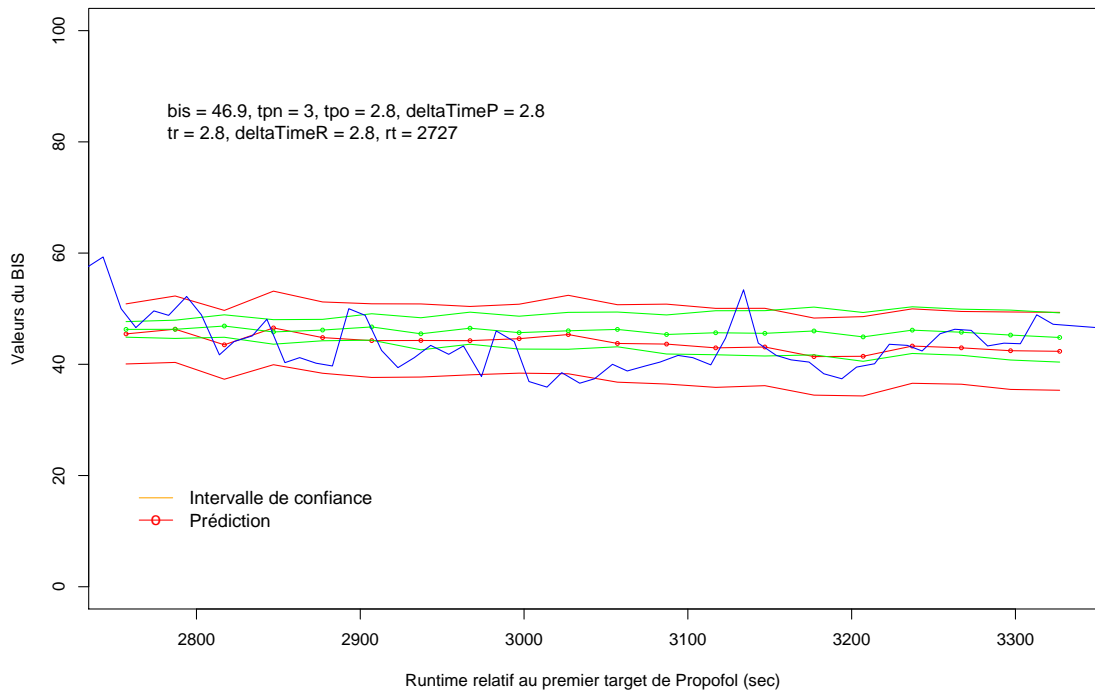
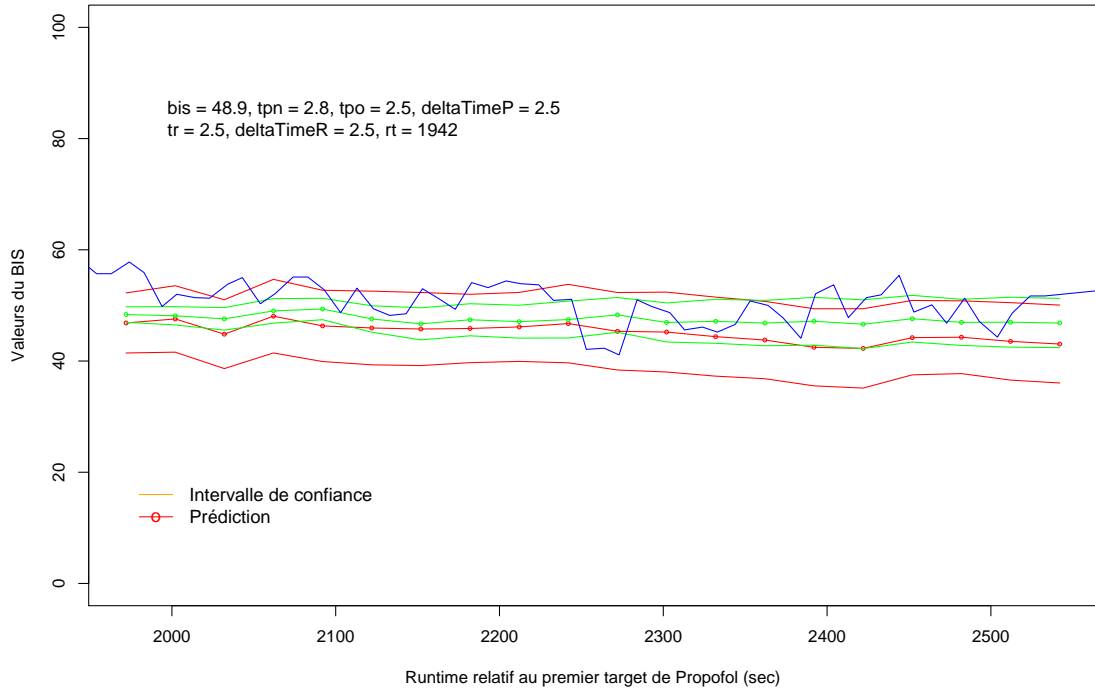
Annexe C

Graphiques comparatifs des prédictions des différents modèles

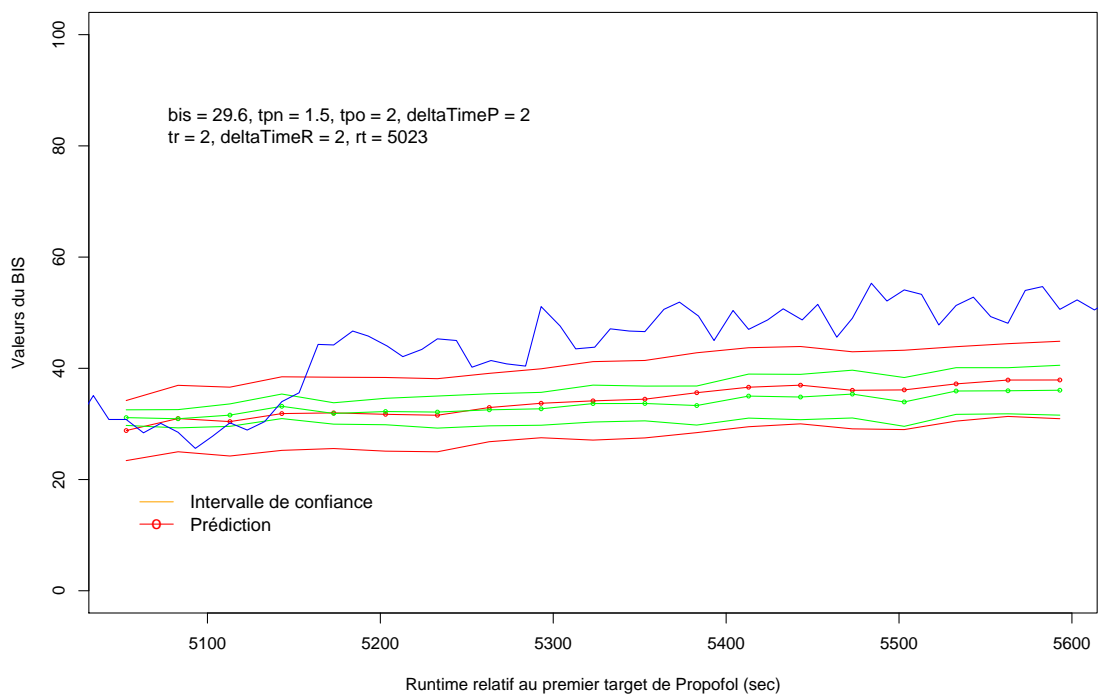
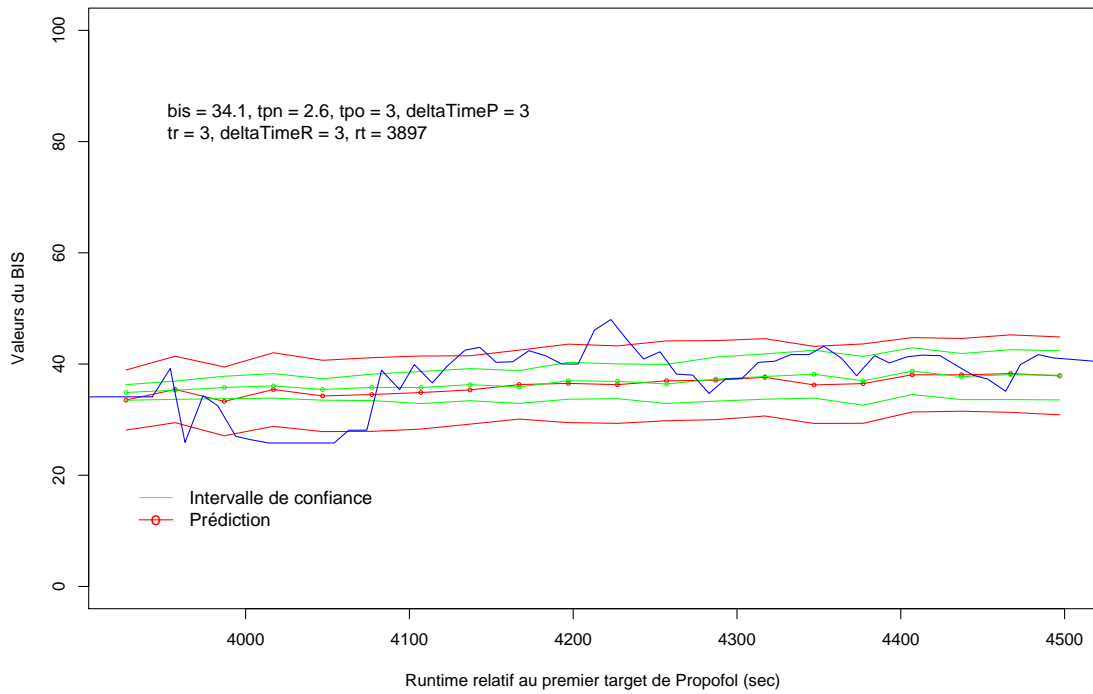
Dans cette annexe, je présente un graphique comparatif des prédictions réalisées par le modèle linéaire avec variables aléatoires (courbes rouges) ainsi que le modèle Lazy avec variables aléatoires (courbes vertes). Par souci de lisibilité j'ai préféré ne pas ajouter les deux autres modèles car leurs prédictions sont similaires à celles des modèles présentés ici.

Ces graphiques sont issus d'une des trois anesthésies. Chaque graphique correspond à une prédiction. Le contexte de la prédiction est renseigné par la valeur qu'avaient les variables d'entrée au moment de la prédiction.

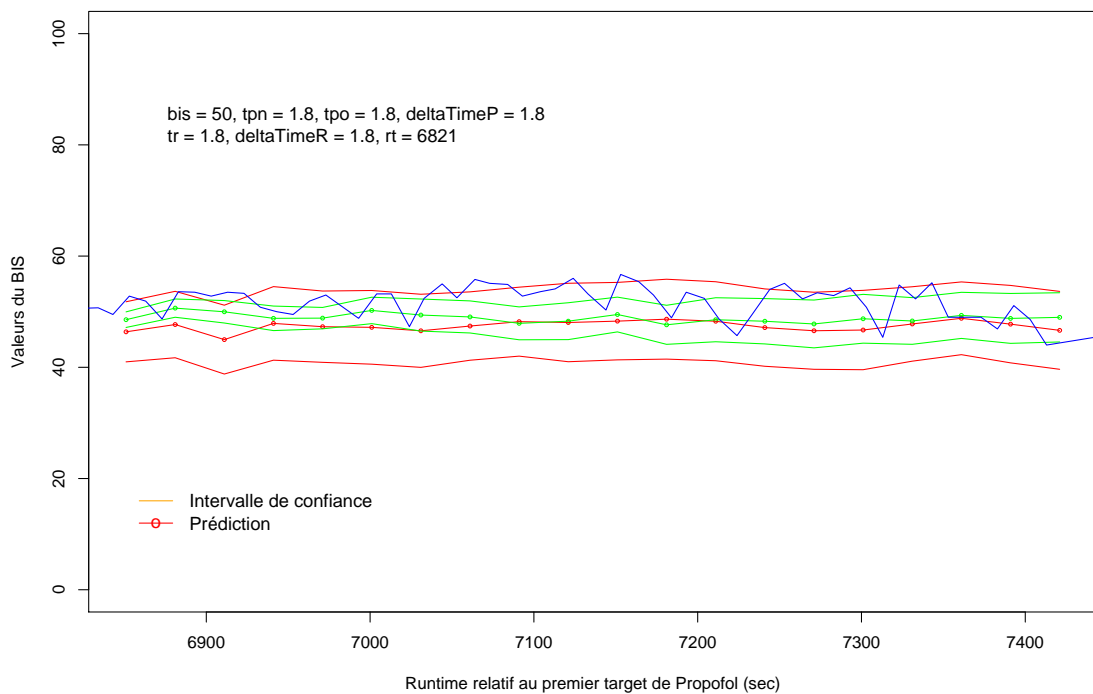
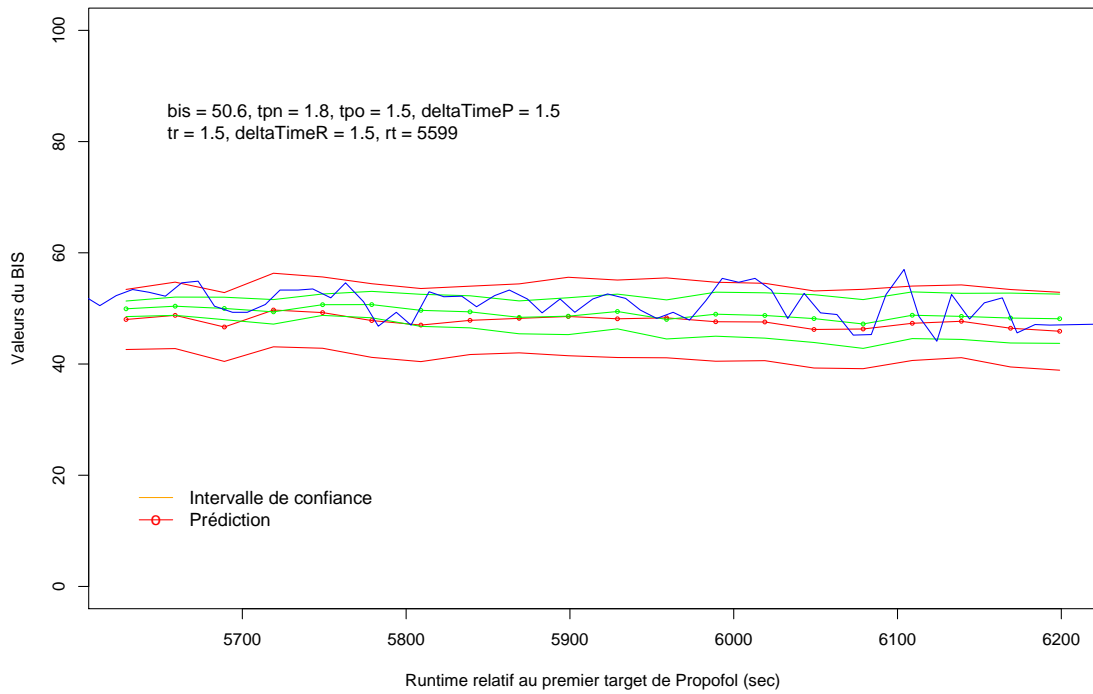




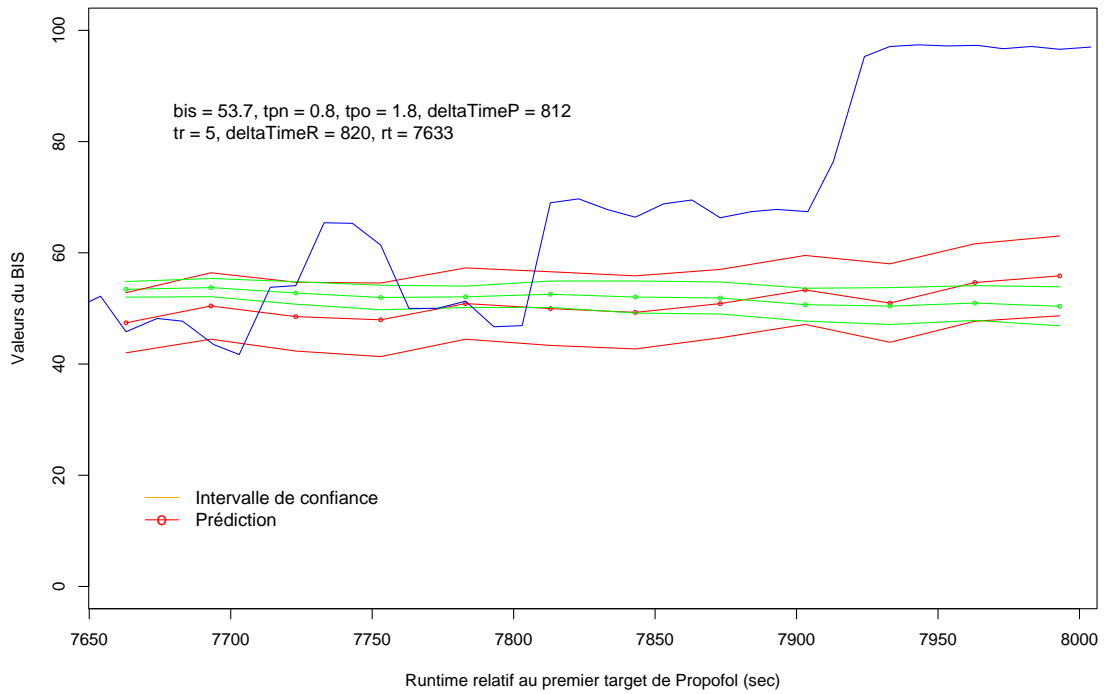
ANNEXE C. GRAPHIQUES COMPARATIFS DES PRÉDICTIONS DES DIFFÉRENTS MODÈLES



ANNEXE C. GRAPHIQUES COMPARATIFS DES PRÉDICTIONS DES DIFFÉRENTS MODÈLES



ANNEXE C. GRAPHIQUES COMPARATIFS DES PRÉDICTIONS DES DIFFÉRENTS MODÈLES



Bibliographie

- [1] V. Billard. Surveillance de la profondeur de l'anesthésie. *Conférences d'actualisation*, pages 17–32, 1997.
- [2] Région Wallonne. Tania propose, l'anesthésiste dispose. In *Revue ATHENA*, number 224, Octobre 2006.
- [3] Olivier Caelen, Gianluca Bontempi, and Luc Barvais. Machine learning techniques to build a decision support tool in anesthesia. 2007. MLG - Université Libre de Bruxelles.
- [4] Norbert Roewer and Holger Thiel. *Atlas de poche d'anesthésie*. Flammarion, 2003.
- [5] F. Cantraine and E. Coussaert. The first object oriented monitor for intravenous anesthesia. *The Journal of Clinical Monitoring and Computing*, 16 :3-10, 16(1) :3–10(8), January 2000.
- [6] Alpaydin and Ethem. *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004.
- [7] Gianluca Bontempi. *Local Learning Techniques for Modeling, Prediction and Control*. PhD thesis, Université Libre de Bruxelles, December 1999.
- [8] Mauro Birattari, Gianluca Bontempi, and Hugues Bersini. Lazy learning meets the recursive least squares algorithm. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 375–381. MIT Press, 1999.
- [9] Liu N, Chazot T, Genty A, Landais A, Restoux A, McGee K, Laloë PA, Trillat B, Barvais L, and Fischler M. Titration of propofol for anesthetic induction and maintenance guided by the bispectral index : closed-loop versus manual control : a prospective, randomized, multicenter study. *Anesthesiology*, 104(4) :686–695, April 2006.
- [10] Gentilini A., Frei C.W., Glattfedler A.H., Morari M., Sieber T.J., Wymann R., Schnider T.W., and A.M. Zbinden. Multitasked closed-loop control in anesthesia. volume 20, pages 39–53, January/February 2001.
- [11] James M. Bailey, Wassim M. Haddad, and Tomohisa Hayakawa. Closed-loop control in clinical pharmacology : Paradigms, benefits, and challenges. volume 3, pages 2268–2277, July 2, 2004.
- [12] Olivier Caelen, Gianluca Bontempi, Eddy Coussaert, Luc Barvais, and Francois Clement. Machine learning techniques to enable closed-loop control in anesthesia. In *CBMS '06 : Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 696–701. IEEE Computer Society, 2006.
- [13] Gianluca Bontempi. *Statistical foundations of machine learning*. 2008. Cours dispensé par l'auteur à l'Université Libre de Bruxelles - Département d'Informatique.

BIBLIOGRAPHIE

- [14] F. Blayo and M. Verleysen. *Les réseaux de neurones artificiels*. Presses Universitaires de France, 1996.
- [15] J. Ghosh and A. Nag. An overview of radial basis function networks. pages 1–36. Physica-Verlag GmbH, 2001.
- [16] J.H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*,19, 1991.
- [17] R. H. Myers. *Classical and Modern Regression with Applications*. PWS-KENT Publishing Company, Boston, MA, second edition, 1994.
- [18] J. S. U. Hjorth. Computer intensive statistical methods. *Chapman and Hall*, 1994.
- [19] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.
- [20] Kohavi Ron and John George H. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2) :273–324, 1997.
- [21] C. M. Bishop. *Neural networks for pattern recognition*. Oxford UK : Oxford University Press, 1994.
- [22] D. M. Allen. *The relationship between variables and data augmentation and a method of prediction*, pages 125–127. Technometrics, 1974.
- [23] Alan T Hope and Roman Rosipal. Measuring depth of anesthesia using electroencephalogram entropy rates.
- [24] D. Longrois, M. Hirshci, E. Junke, and C. Meistelman. Monitorage de la profondeur de l’anesthésie. *plo*, 2001.
- [25] Zikov T., Bibian S.and Dumont G.A.and Huzmezan M., and Ries C.R. Quantifying cortical activity during general anesthesia using wavelet analysis. volume 53, pages 617–632, April 2006.
- [26] F. Servin. Anesthésie intraveineuse à objectif de concentration. *Conférences d’actualisation 1998*, p. 35-48., 1998.
- [27] M. Tod F. Lagneau and J. Marty. Applications cliniques de la pharmacologie des agents anesthésiques intraveineux (hypnotiques et morphiniques). *Conférences d’actualisation 2003*, p. 181-202, 2003.
- [28] Schnider TW, Minto CF, Shafer SL, and al. The influence of age on propofol pharmacodynamics. volume 90, pages 1502–1516. Lippincott, Hagerstown, 1999.
- [29] Minto CF, Schnider TW, Egan TD, and al. Influence of age and gender on the pharmacokinetics and pharmacodynamics of remifentanyl. volume 86, pages 10–23, 1997.
- [30] Gianluca Bontempi, Mauro Birattari, and Hugues Bersini. Lazy learning for local modeling and control design. *International Journal of Control*. Accepted, 72 :643–658, 1997.
- [31] Sidney Siegel and N. John Castellan. *Nonparametric Statistics for the Behavioral Sciences*. New York : McGraw-Hill, 1956.
- [32] P. Filzmoser. A multivariate outlier detection method. volume 1, pages 18–22. Belarusian State University, Minsk, 2004.