



HAL
open science

Contribution à la reconnaissance d'activités à partir d'un objet connecté

Romain Auber

► **To cite this version:**

Romain Auber. Contribution à la reconnaissance d'activités à partir d'un objet connecté. Automatique / Robotique. Normandie Université, 2019. Français. NNT : 2019NORMC242 . tel-02520542v2

HAL Id: tel-02520542

<https://theses.hal.science/tel-02520542v2>

Submitted on 27 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité AUTOMATIQUE, SIGNAL, PRODUCTIQUE, ROBOTIQUE

Préparée au sein de l'Université de Caen Normandie

Contribution à la reconnaissance d'activités à partir d'un objet connecté

**Présentée et soutenue par
Romain AUBER**

**Thèse soutenue publiquement le 10/12/2019
devant le jury composé de**

M. EMMANUEL GODOY	Professeur des universités, CentraleSupélec	Rapporteur du jury
M. PIERRE MELCHIOR	Professeur des universités, IMS-Bordeaux	Rapporteur du jury
Mme ESTELLE BLOND	Ingénieur, Bodycap	Membre du jury
M. ALAIN OUSTALOUP	Professeur émérite, Institut national polytechnique Bordeaux	Président du jury
M. MATHIEU POULIQUEN	Maître de conférences, Université Caen Normandie	Membre du jury
Mme NEZHA TRIGEASSOU MAAMRI	Maître de conférences HDR, Université de Poitiers	Membre du jury

Thèse dirigée par MOHAMMED M'SAAD et STEPHANE FLAMENT, Groupe de recherche en informatique, image, automatique et instrumentation



UNIVERSITÉ
CAEN
NORMANDIE



Remerciements

Ce manuscrit est le fruit d'un travail de recherche de près de trois ans. Aussi, je souhaite adresser mes remerciements aux personnes qui ont contribué à ce travail et qui m'ont soutenu et encouragé.

Mes remerciements vont d'abord à tous ceux qui ont participé au financement de cette thèse, à savoir la société Bodycap avec le soutien de l'ANRT.

Je souhaite également remercier mes directeurs de thèse, Monsieur Mohammed M'Saad et Monsieur Stéphane Flament, professeurs à l'ENSICAEN, d'avoir accepté de diriger ce travail. Je tiens aussi à remercier les membres du jury d'avoir accepté de juger et d'examiner ce travail.

Je remercie tout autant Monsieur Sébastien Moussay, Président de la société Bodycap, de m'avoir donné l'opportunité de travailler sur cette thèse au sein de son entreprise.

Je tiens également à remercier Monsieur Mathieu Poulinquen, Maître de Conférences à l'Université de CAEN, pour m'avoir guidé, conseillé et soutenu durant cette thèse. Je remercie aussi les membres du laboratoire d'automatique de Caen qui m'ont accompagné durant la thèse, en particulier Monsieur Eric PIGEON, Monsieur Olivier Gehan, Monsieur Thomas Menard et Monsieur Miloud Frikel.

Un grand merci également à tous les employés de Bodycap pour leur accueil, leur aide et leurs conseils. C'est un plaisir de travailler avec vous.

Merci à ma femme Anaïs qui a su me motiver durant les trois années de thèse. Merci pour ton soutien et tes encouragements.

Merci aux doctorants du laboratoire qui m'ont accompagné durant la thèse. Je remercie ainsi Clément, Tristan et Abdelhak.

Merci, enfin, à toute ma famille, mes proches et amis qui m'ont encouragé et soutenu tout au long de ce travail de thèse.

Table des matières

Acknowledgements	ii
1 Introduction Générale	1
1.1 Contexte et motivation	1
1.2 Organisation de la thèse	3
2 État de l’art sur la reconnaissance d’activité à partir de données d’un accéléromètre	7
2.1 Introduction	7
2.1.1 Généralité	7
2.1.2 Méthodologie générale pour la reconnaissance d’activités	8
2.2 Les données brutes	10
2.2.1 Les données accélérométriques	10
2.2.2 Fréquence	13
2.3 Placement des capteurs	14
2.4 Le pré-traitement des données	17
2.4.1 Filtrage	17
2.4.2 Fenêtrage	18
2.5 Extraction et sélection de paramètres de classification	19
2.5.1 Liste des paramètres	21
2.5.1.1 Moyenne, variance et écart type	21
2.5.1.2 Médiane, percentile, maximum, minimum et éventail	25
2.5.1.3 Corrélation	26
2.5.1.4 Autres paramètres dans le domaine temporel	26
2.5.1.5 La Transformation de Fourier discrète (TFD)	27
2.5.1.6 Transformée en ondelettes discrète	28
2.5.1.7 Modèle auto-régressif (AR)	30
2.5.2 Sélection des paramètres	31

2.5.3	Normalisation des paramètres	35
2.6	Classification	35
2.6.1	Apprentissage par arbre de décision	38
2.6.2	Les forêts aléatoires	39
2.6.3	Naïve Bayésienne	39
2.6.4	Analyse discriminante	40
2.6.5	K plus proche voisins	41
2.6.6	Séparateur à Vaste Marge (SVM)	42
2.6.7	Réseaux de neurones	44
2.6.8	Combinaison d'algorithmes de classification	46
2.7	Validation	47
2.7.1	Données de validation	47
2.7.2	Indicateurs de performance	47
2.8	Conclusion	49
3	L'environnement d'eTact	51
3.1	Introduction	51
3.2	Présentation d'eTact	51
3.3	Les contraintes	53
3.3.1	Contrainte matériel	53
3.3.2	Positionnement	54
3.4	Les différents approches proposées	54
3.5	Les données d'apprentissages	56
3.6	Conclusion	58
4	Reconnaissance d'activités avec eTact	61
4.1	Introduction	61
4.2	Choix des paramètres	62
4.2.1	<i>Ensemble_Param_A</i> : Ensemble de paramètres choisis calculés sur une plateforme externe	62
4.2.2	<i>Ensemble_Param_B</i> : Ensembles de paramètres calculés sur eTact	64
4.2.3	Normalisation	65
4.3	Réduction de dimension des ensembles de paramètre	65
4.3.1	Algorithme de réduction de dimension utilisé	65
4.3.1.1	Sélection de paramètres avec l'algorithme ReliefF	66

4.3.1.2	Sélection de paramètres avec l'algorithme Sequential Forward Selection	66
4.3.2	Réduction de l'ensemble <i>Ensemble_Param_A</i>	69
4.3.3	Réduction de l'ensemble <i>Ensemble_Param_B</i>	69
4.4	Algorithmes d'apprentissage	71
4.5	Impact de la taille de la fenêtre glissante T_w	71
4.6	Impact de la fréquence d'échantillonnage de l'accéléromètre	72
4.7	Comparaison des différents approches	74
4.7.1	Résultats de l'ensemble de paramètres <i>Ensemble_Param_A</i>	74
4.7.2	Résultats de l'ensemble de paramètres <i>Ensemble_Param_B</i>	76
4.7.3	Autonomie d'eTact en fonction de l'approche utilisée	77
4.8	Conclusion	79
5	Identification d'un modèle autorégressif à partir de données binaires ou quantifiées	81
5.1	Introduction	81
5.2	Formulation du problème	82
5.3	Estimation d'un modèle AR pour un seuil égal à 0	84
5.4	Estimation d'un modèle AR pour un seuil différent de 0	85
5.4.1	Algorithme proposé	85
5.4.2	Résultats de convergence	89
5.4.3	Analyse de la variance	91
5.4.4	Résultats numériques	95
5.5	Estimation d'un modèle AR en temps réel	100
5.5.1	Algorithme temps réel proposé	100
5.5.2	Résultats de convergence	101
5.5.3	Résultats numériques	104
5.6	Extension de l'algorithme pour des données quantifiées	104
5.6.1	Formulation du problème	104
5.6.2	Première approche	106
5.6.3	Seconde approche	106
5.6.4	Résultats numériques	108
5.7	Utilisation d'une série temporelle non linéaire	109
5.7.1	Formulation du problème	109
5.7.2	Algorithme proposé	109

5.8	Application de l'identification d'un modèle AR sur des données réelles binaires d'un accéléromètre	111
5.8.1	Contexte	111
5.8.2	Algorithme proposé	112
5.8.3	Résultats numériques	112
5.9	Conclusion	115
6	Reconnaissance d'activités avec des données binaires	117
6.1	Introduction	117
6.2	Choix des paramètres	118
6.2.1	Normalisation	119
6.3	Réduction de dimension de l'ensemble de paramètres	119
6.4	Impact du seuil T_w	120
6.5	Impact du seuil C_b	120
6.6	Comparaison des résultats de classification	121
6.6.1	Performance des algorithmes de classification	121
6.6.2	Autonomie d'eTact	123
6.7	Conclusion	124
7	Identification d'un modèle commutant linéaire à partir de données binaires	127
7.1	Introduction	127
7.2	Formulation du problème	128
7.3	Algorithme d'identification	129
7.3.1	Principe de l'algorithme	129
7.3.2	Description de l'algorithme proposé	132
7.3.2.1	Estimation de l'ensemble \mathcal{B}	132
7.3.2.2	Identification des sous-systèmes	137
7.3.2.3	Résumé de l'algorithme	137
7.4	Exemple numérique	139
7.4.1	Premier exemple	139
7.4.2	Deuxième exemple	139
7.5	Conclusion	140
8	Conclusion générale	143
8.1	Conclusion	143
8.2	Perspectives	145

Bibliographie

147

Résumé

156

Table des figures

2.1	Méthodologie générale pour la reconnaissance d'activités	9
2.2	Exemple de données d'un accéléromètre pendant la marche	11
2.3	Décomposition du signal s avec la transformée en ondelettes	29
2.4	Procédure pour la sélection des paramètres	32
2.5	Exemple d'un arbre de décision	38
2.6	Exemple d'un réseaux de neurones avec deux couches cachées	45
3.1	eTact	52
3.2	Architecture d'eTact	53
4.1	Poids des différents paramètres de l'ensemble <i>Ensemble_Param_A</i> avec l'utilisation de l'algorithme ReliefF pour $T_w = 1s$	69
4.2	Justesse de la classification en fonction du nombre de paramètres avec l'algorithme SFS pour l'algorithme de classification des K plus proches voisins avec $T_w = 2s$ et le capteur positionné au torse	70
4.3	Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné au torse pour l'ensemble de paramètres <i>Ensemble_Param_B</i> avec $f = 50Hz$	72
4.4	Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné au poignet pour l'ensemble de paramètres <i>Ensemble_Param_B</i> avec $f = 50Hz$	73
4.5	Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné à la cheville pour l'ensemble de paramètres <i>Ensemble_Param_B</i> avec $f = 50Hz$	73
4.6	Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné à la taille pour l'ensemble de paramètres <i>Ensemble_Param_B</i> avec $f = 50Hz$	74

4.7	Justesse de la classification des k plus proches voisins en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné à la taille pour l'ensemble de paramètres <i>Ensemble_Param_B</i> pour $f = 25\text{Hz}$ et $f = 50\text{Hz}$	75
5.1	Représentation graphique $P_C(i)$, $Q_C(i)$ et $R_C(i)$ pour $C = 2$.	87
5.2	$P'_C(\bar{\rho}_y(i))$ en fonction de $\bar{\rho}_y(i)$ pour les seuils $C = 0\sigma_y$, $C = 0.5\sigma_y$, $C = 1\sigma_y$ et $C = 1.5\sigma_y$	96
5.3	Erreur de l'estimation des paramètres du modèle AR en fonction du seuil C de l'algorithme avec le seuil $C = 0$ et l'algorithme avec le seuil $C \neq 0$ pour $N = 10000$	97
5.4	Erreur de l'estimation des paramètres du modèle AR en fonction du seuil C avec un nombre de données $N=1000$, $N=5000$, $N=10000$ et $N=50000$	97
5.5	Erreur de l'estimation des paramètres du modèle AR en fonction du nombre de données N pour les seuils $C = 0.05\sigma_y$, $C = 0.5\sigma_y$, $C = 0.75\sigma_y$ and $C = \sigma_y$	98
5.6	Erreur de l'estimation des paramètres du modèle AR en fonction du seuil C avec l'algorithme utilisant la proportion P_C , la Proportion Q_C et la matrice M	99
5.7	La valeur expérimentale de $\sum_{i \in [1, n]} \sigma_{\hat{\theta}(i)}^2 N$ en fonction du seuil C pour $N = 10000$ et $N = 40000$	100
5.8	évolution de $\hat{\rho}_y(i)_t$ en fonction de t	104
5.9	Évolution de $\hat{\theta}_t$ en fonction de t	105
5.10	Représentation graphique des différentes proportions $P_{j,k}(i)$ dans le cas de l'utilisation de deux seuils $C_1 = -2$ et $C_2 = 2$	107
5.11	Moyenne des erreurs de l'estimation des paramètres du modèle AR en fonction du nombre de données N avec l'algorithme avec un seuil et l'algorithme avec quatre seuils	108
5.12	Représentation de la proportion $T_C(\rho_y(i))$ (représentée en hachuré)	110
5.13	La justesse de la classification à partir des paramètre calculée avec l'algorithme avec un seuil $C = 0$ en fonction de la taille de fenêtre T_w pour le degré du modèle AR $p = 2$, $p = 5$ et $p = 8$.	113
5.14	La justesse de la classification à partir des paramètre calculé avec l'algorithme avec un seuil $C = 0$ en fonction du degré du modèle AR p pour une taille de fenêtre $T_w = 2s$, $T_w = 5s$ et $T_w = 8s$	114

5.15	La justesse de la classification à partir des paramètre calculé avec l'algorithme avec un seuil $C \neq 0$ en fonction de la taille de fenêtre T_w pour un degré du modèle AR $p = 10$ et pour un seuil $C = 0,4g$, $C = 0,8g$ et $C = 0,12g$	114
5.16	La justesse de la classification à partir des paramètre calculé avec l'algorithme avec un seuil $C \neq 0$ en fonction du seuil C pour un degré du modèle AR $p = 10$ et une taille de fenêtre $T_w = 2s$, $T_w = 7s$ et $T_w = 12s$	115
6.1	Justesse de la classification (Forêts aléatoire) en fonction du temps T_w pour les différentes position du capteur avec $f = 50Hz$ et $C_b = 0$	120
6.2	Justesse de la classification (arbre de décision) en fonction du seuil C_b pour les différentes positions du capteur avec $f = 50Hz$	121
6.3	L'autonomie de la mémoire en fonction de l'autonomie de la batterie pour les différentes approches avec $f = 50Hz$	124
7.1	Définition des ensemble \mathcal{O}_1^+ and \mathcal{O}_1^- avec trois sous-systèmes pour un exemple de deux dimensions ($n + 1 = 2$)	131
7.2	Définition de $\mathcal{O}^+ = \cap_j \mathcal{O}_j^+$ et de $\mathcal{O}^- = \cap_j \mathcal{O}_j^-$ pour un exemple de deux dimensions ($n + 1 = 2$) et de deux sous-systèmes.	132
7.3	Représentation graphique des ensembles $\hat{\mathcal{O}}_N^+$ et $\hat{\mathcal{O}}_N^-$ avec $\delta = 1$	134
7.4	Représentation graphique de l'ensemble $\hat{\mathcal{O}}_N^{+/-}$ avec $\delta = 1$	135
7.5	Représentation graphique des ensembles $\hat{\mathcal{B}}_N^+$ et $\hat{\mathcal{B}}_N^-$ avec $\delta = 1$. $\hat{\mathcal{B}}_N$ est l'union des ensembles $\hat{\mathcal{B}}_N^+$ et $\hat{\mathcal{B}}_N^-$	136
7.6	La moyenne de l'erreur $mean \left(\sum_{i=1}^3 \ \theta_i - \hat{\theta}_i\ _2 \right)$ en fonction du rapport signal sur bruit.	141

Liste des tableaux

2.1	Type d'équipement utilisé pour la collecte de données	11
2.2	Fréquence d'échantillonnage utilisé pour la reconnaissance d'activité	15
2.3	placement du capteur	16
2.4	Longueur de fenêtres utilisées dans la reconnaissance d'activité	20
2.5	paramètres utilisés pour la reconnaissance d'activité	22
2.6	paramètres utilisés pour la reconnaissance d'activité	23
2.7	Classification supervisée utilisée pour la reconnaissance d'activité	37
3.1	Récapitulatif des différentes approches	56
3.2	Description des différents types d'activité de la base de données PAMAP2	58
3.3	Description des différents types d'activité de la base de données USC-HAD	58
3.4	Temps d'acquisition des différentes activités pour chaque emplacement du capteur de la base de données	59
4.1	pseudo-code de l'algorithme ReliefF	67
4.2	pseudo-code de l'algorithme Sequential Forward Selection	68
4.3	Justesse des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres <i>Ensemble_Param_A</i> avec $T_w = 5s$	75
4.4	F1score des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres <i>Ensemble_Param_A</i> avec $T_w = 5s$	76
4.5	Justesse des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres <i>Ensemble_Param_B</i> avec $T_w = 5s$	76
4.6	F1score des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres <i>Ensemble_Param_B</i> avec $T_w = 5s$	77
4.7	Autonomie mémoire eTact en fonction des différentes approches en utilisant l'algorithme des forêts aléatoires	78

4.8	Autonomie de la batterie d'eTact en fonction des différentes approches en utilisant l'algorithme des forêts aléatoires	79
5.1	Résumé de l'algorithme pour un seuil $C = 0$	86
5.2	Résumé de l'algorithme pour un seuil $C \neq 0$	90
5.3	Résumé de l'algorithme en temps réel	101
6.1	Justesse des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres <i>Ensemble_Param_C</i> avec $T_w = 5s$ et $C_b = 0.15g$	122
6.2	F1score des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres <i>Ensemble_Param_C</i> avec $T_w = 5s$ et $C_b = 0.15g$	122
6.3	Autonomie mémoire eTact en fonction des différentes approches	123
6.4	Autonomie de la batterie d'eTact en fonction des différentes approches	123
7.1	Résumer de l'algorithme en deux étapes	138
7.2	Paramètres estimés du premier exemple	140
7.3	Paramètres estimés du deuxième exemple	142

Liste des symboles

\mathbb{R}	Ensemble des nombres réels
\mathbb{R}^n	Espace vectoriel de dimension n dans l'ensemble des réels
$\max x$	la valeur maximal de x
$\min x$	la valeur minimal de x
μ_x	Moyenne de série temporelle $\{x_t\}$
σ_x	Variance de série temporelle $\{x_t\}$
$\{\bar{z}_t\}$	La séquence complémentaire de $\{z_t\}$ définie par $\bar{z}_t = 1 - z_t$
$E(x)$	La partie entière de x
I_n	Matrice identité de dimensions $n \times n$
t	variables temporelle

Acronymes

AR	AutoRég ressif
BLE	Bluethoot Low Energy
FN	Faux N égatifs
FP	Faux P ositifs
IMU	Inertial Measurement Units
GPS	Global Positionong System
TAT	Time de Above Threshold
TFD	Transformée de Fourier Discrète
MCU	MicroControleur Unit
TFD	Transformée de Fourier Discrète
ROC	Receiver Operating Characteristic
USB	Universal Serial Bus
VN	Vrais N égatifs
VP	Vrais P ositifs
SFS	Sequential Foward Selection
SVM	Séparateur à Vaste Marge
SNR	Signal-to-Noise Ratio
ZCM	Zero Crossing Mode

Chapitre 1

Introduction Générale

1.1 Contexte et motivation

De nos jours, les capteurs sont de plus en plus présents dans notre quotidien et les données de ces capteurs peuvent être collectées pour être analysées et nous fournir de nombreuses informations sur nos comportements. Par exemple, de plus en plus de personnes sont en possession d'un smartphone et ces appareils contiennent différents capteurs tel qu'un accéléromètre, gyroscope, thermomètre, microphone, baromètre, magnétomètre ou GPS. Les montres connectées peuvent aussi être équipées d'un capteur qui permet de mesurer par exemple la fréquence cardiaque. Il est ainsi possible de collecter un nombre impressionnant de données sur les utilisateurs à partir de ces produits de la vie courante.

Les smartphones et les montres connectés ne sont pas les seuls appareils disponibles pour collecter des données sur des personnes. De plus de plus d'objets connectés, embarquant des capteurs pour le suivi des personnes, sont disponibles sur le marché. Ces objets connectés sont développés dans le but de collecter des types de données ciblées. En conséquence, l'hardware de ces objets connectés est développé de manière à avoir des données de bonne qualité et facile à récupérer. Le cadre d'utilisation de ces objets connectés permet de collecter des données de meilleure qualité car, si besoin, la position de ces objets peut être fixée par avance, à la différence de la position du smartphone qui, elle, diffère selon l'activité de l'utilisateur.

On peut citer en exemple les objets connectés développés par Bodycap. Le premier

produit est e-Celsius. Il s'agit d'une pilule ingérable qui embarque un capteur de température pour monitorer la température corporelle d'une personne. Les données du capteur de température sont envoyées en temps réel par un protocole sans fil avec un temps d'échantillonnage déterminé par l'utilisateur. Ce produit sert, entre autre, à améliorer les performances des sportifs ou à surveiller la température d'un patient post-ambulatoire pour détecter une possible infection. Le deuxième produit développé par Bodycap est eTact. Cet objet connecté est un patch qui embarque un accéléromètre et un capteur de température ce qui permet ainsi de quantifier l'activité d'une personne. De plus, il peut transmettre aussi les données collectées par un protocole de communication sans fils. Des algorithmes d'actimétrie peuvent être embarqués afin d'évaluer, par exemple, la qualité de sommeil d'une personne ou de monitorer l'activité de personnes fragiles (des personnes âgées ou personnes atteintes d'un cancer).

Une des conséquences de cette abondance de données est l'intérêt suscité ces dernières années par la reconnaissance d'activités. Elle s'applique à de nombreux domaines d'application et particulièrement dans le domaine de la santé. La reconnaissance d'activités peut par exemple permettre de suivre l'activité de personnes pour détecter un changement d'activité pouvant signifier un problème de santé. Un autre exemple est l'observation de l'augmentation de l'activité, signe d'un bon rétablissement post-opératoire. La reconnaissance d'activités peut aussi être utilisée dans le domaine du sport pour mesurer les performances des sportifs et améliorer les séances d'entraînement.

La société Bodycap souhaite aller plus loin avec eTact et souhaite développer des algorithmes pour faire de la reconnaissance d'activités avec cet objet embarqué. Bien qu'il existe de nombreuses solutions dans la littérature, le développement d'un algorithme de reconnaissance d'activités pour un objet connecté n'est pas un problème simple. L'utilisation d'eTact induit des contraintes qui ont un impact sur l'algorithme. Parmi ces contraintes, la durée de vie du patch et les performances de l'algorithme sont les deux variables importantes qui doivent être prises en considération. Pour réduire la consommation, il est possible d'envoyer les données par un protocole sans fil à partir d'eTact vers un appareil externe afin de réduire la charge de calcul sur le patch. Néanmoins, le nombre de données envoyées par le protocole sans ne doit pas être trop important car cela consomme de l'énergie. Pour ce qui est des performances, elles vont dépendre de l'algorithme et du choix de ses paramètres de synthèse. Un autre aspect important est le positionnement du patch. De ce positionnement va dépendre la qualité des informations

relevées et la nature des activités qui pourront être reconnues. Le but de cette thèse est ainsi d'explorer différentes solutions pour faire de la reconnaissance d'activités à partir d'eTact.

1.2 Organisation de la thèse

Ce manuscrit de thèse se compose de huit chapitres, dont l'introduction et la conclusion.

Le deuxième chapitre s'organise autour d'un état de l'art sur la reconnaissance d'activités. Le but de ce chapitre est de recenser les différentes solutions pour faire de la reconnaissance d'activités à partir de données d'un accéléromètre. Les différentes solutions se présentent dans la majorité des cas autour de l'utilisation d'algorithmes de classification et requièrent un pré-traitement des données. Ce pré-traitement consiste en la division des données de l'accéléromètre en fenêtres d'analyse. Une activité est associée à chaque fenêtre et un ensemble de paramètres est calculé pour ensuite être utilisé pour la classification. Il existe un grand nombre de paramètres possibles à calculer. Les paramètres les plus utilisés dans la littérature sont présentés dans ce chapitre. Pour finir, les différentes métriques pour mesurer les performances des algorithmes d'apprentissage sont présentées.

Le troisième chapitre se concentre sur la présentation du dispositif eTact. L'utilisation de ce dispositif implique des contraintes liées au hardware telles que la fréquence d'échantillonnage, la durée de vie de la batterie ou la capacité mémoire d'eTact. Ces différentes contraintes sont présentées dans ce chapitre pour être prises en compte pour l'élaboration de l'algorithme de reconnaissance d'activités. De plus, eTact a la possibilité d'envoyer des données vers un système externe pour externaliser la charge de calcul. À partir de cette capacité, quatre différentes approches sont présentées pour le développement d'un algorithme de reconnaissance d'activités. Ces différentes approches varient selon le support utilisé pour le calcul des paramètres et la mise en oeuvre des algorithmes de classification supervisée. Pour l'approche 1, eTact stocke, puis envoie directement les données brutes vers un appareil externe sur lequel le calcul des paramètres et l'algorithme de classification sont exécutés. Pour l'approche 2, le calcul des paramètres est effectué directement sur Etact, puis les paramètres sont envoyés sur un appareil externe où l'algorithme de classification est exécuté. Pour l'approche 3, l'appareil externe n'est

pas utilisé. Le calcul des paramètres et la mise en oeuvre de l'algorithme de classification sont directement réalisés sur eTact. À l'inverse des trois premières approches qui utilisent directement les données brutes de l'accéléromètre, l'approche 4 a pour particularité de binariser les données avant de les utiliser pour la reconnaissance d'activités. Les données brutes binarisées sont ensuite envoyées vers un appareil externe pour le calcul des paramètres et la mise en oeuvre de l'algorithme de classification comme pour la première approche. Pour finir, l'utilisation d'algorithmes de classification supervisées requiert une base de données d'apprentissage. Les bases de données utilisées dans cette thèse sont présentées dans ce chapitre.

Le quatrième chapitre traite le problème de la reconnaissance d'activités à partir des données brutes d'eTact. Les trois approches qui utilisent directement les données brutes de l'accéléromètre sans les binariser sont étudiées dans ce chapitre. Dans un premier temps, deux ensembles de paramètres sont présentés. Le premier ensemble correspond à des paramètres qui peuvent nécessiter un grand nombre de calculs et qui sont calculés sur une plateforme externe. Le deuxième ensemble correspond aux paramètres qui peuvent être calculés directement sur eTact et qui nécessitent peu de calcul. Ensuite, deux algorithmes de sélection de paramètres sont appliqués pour réduire la dimension de ces ensembles. À partir de ces ensembles réduits, plusieurs algorithmes de classification supervisée sont utilisés pour faire de la reconnaissance d'activités. Les différentes approches sont finalement comparées.

Le cinquième chapitre se consacre à l'identification d'un modèle autorégressif à partir de données binaires. La dernière approche pour la reconnaissance d'activités a en effet pour particularité d'utiliser des données binaires et le but de ce chapitre est d'étudier les solutions pour estimer les paramètres d'un modèle autorégressif à partir d'un signal binaire, pour ensuite utiliser ces paramètres pour la classification. Dans la littérature, une solution existe pour l'estimation des paramètres d'un modèle autorégressif dans le cas où le signal est binarisé avec un seuil nul. Cette solution est tout d'abord présentée. Ensuite, nous proposons un nouvel algorithme pour l'estimation d'un modèle autorégressif dans le cas où le seuil est non nul. Une analyse de la consistance, de la variance et des résultats de simulation est proposée. Ensuite, un algorithme adapté au temps réel est proposé. Pour continuer, l'algorithme est étendu pour des données quantifiées. Pour finir, l'algorithme est testé sur des données de l'accéléromètre.

Le sixième chapitre se concentre sur la quatrième approche pour faire de la reconnaissance d'activités. La particularité de cette approche est d'utiliser seulement les données binarisées de l'accéléromètre pour faire de la reconnaissance d'activités. Dans un premier temps, une liste de paramètres utilisables pour la classification dans le cas de données binaires est présentée. Ensuite, l'ensemble de paramètres est réduit à l'aide d'algorithmes de réduction de dimensions. Pour finir, plusieurs algorithmes de classification sont comparés pour faire de la reconnaissance d'activités à partir de l'ensemble de paramètres réduit. Les résultats de la reconnaissance d'activités de cette approche sont comparés avec ceux des premières approches étudiées dans le quatrième chapitre.

Le septième chapitre traite le problème d'identification d'un système commutant avec des données binaires. Les résultats du chapitre 4 montrent qu'il est possible de faire de l'identification à partir de données binaires. Un algorithme pour l'identification de ce type de système est proposé. Cet algorithme est ensuite appliqué sur des exemples numériques.

Pour finir, une conclusion générale est apportée à ce manuscrit et des perspectives pour des travaux futurs sont apportées.

Chapitre 2

État de l'art sur la reconnaissance d'activité à partir de données d'un accéléromètre

2.1 Introduction

2.1.1 Généralité

La reconnaissance d'activités est un sujet de recherche qui suscite beaucoup d'intérêt ces dernières années. Une des motivations importantes de la reconnaissance d'activités est le suivi médical des personnes, suivi pouvant être utilisé dans la médecine ambulatoire. La reconnaissance d'activités peut aussi permettre la détection d'anomalies liée à la santé ou prévenir des chutes. Elle peut aussi être utilisée à des fins plus légères, par exemple pour le suivi des activités physiques quotidiennes via une estimation de la dépense calorique ou le suivi des activités sportives (durée d'une course, distance parcourues, etc.).

Il existe plusieurs solutions pour analyser l'activité d'une personne à partir d'un ou plusieurs capteurs, les références suivantes font état de cette diversité : ATTAL et al., 2015, LARA et LABRADOR, 2013, JANIDARMIAN et al., 2017 et PREECE et al., 2009b. L'ensemble de ces revues est utilisé comme base de cette bibliographie. Une première possibilité consiste à faire de la reconnaissance d'activités à partir d'images ou de vidéo. Une revue des différentes techniques utilisées dans le cas de l'utilisation d'images ou de vidéos peut être trouvée dans SAIF, TEHSEEN et KAUSAR, 2018. Cependant, l'utilisation de données visuelles a deux défauts majeurs : d'une part, la personne doit rester dans le champ

de la caméra et, d'autre part, certains utilisateurs peuvent être réticents à être filmés dans leurs vies privées. Une autre possibilité est donc d'utiliser des capteurs portés par l'utilisateur. A ce titre, en raison de sa démocratisation, de sa discrétion et de sa puissance de calcul, le smartphone peut être utilisé pour collecter énormément de données sur les individus et ces données peuvent être utilisées pour de la reconnaissance d'activités. Le smartphone contient en effet différents capteurs tels que des accéléromètres, des gyroscopes, magnétomètres, GPS, microphones et autres. Les revues SHOAIIB et al., 2015 et SAEEDI et EL-SHEIMY, 2015 font un point sur la reconnaissance d'activités via un smartphone en prenant en compte la consommation des ressources du téléphone.

Outre le smartphone, des dispositifs peuvent être spécifiquement développés pour la reconnaissance d'activités. Ainsi, dans JAIN et KANHANGAD, 2018 un gyroscope est utilisé en combinaison avec un accéléromètre. Dans GU et al., 2015, les données d'un baromètre sont utilisées en plus des données d'un accéléromètre. Il existe aussi des dispositifs qui intègrent directement un accéléromètre, un gyroscope et un magnétomètre (CHAPRON et al., 2018). Dans d'autres cas, des signaux physiologiques peuvent être utilisés : la fréquence cardiaque, fréquence respiratoire et la saturation en oxygène sont utilisés dans CASTRO et al., 2017. Dans le cas des maisons connectées, des données complémentaires peuvent être fournies par des capteurs binaires qui indiquent l'emplacement de l'utilisateur dans la maison (CERON, LOPEZ et ESKOFIER, 2018).

Il a été choisi dans cette bibliographie de prendre uniquement en compte les données fournies par un accéléromètre. L'accéléromètre est en effet le capteur porté le plus utilisé dans la reconnaissance d'activités. L'utilisation de ce capteur permet de pouvoir récolter des données de n'importe quel endroit et il permet de ne plus être dépendant de l'environnement de l'utilisateur. Le but de ce paragraphe est de faire une revue des différentes méthodes utilisées dans la littérature pour faire de la reconnaissance d'activités, ceci sur la base de données accélérométriques.

2.1.2 Méthodologie générale pour la reconnaissance d'activités

Comme décrit dans ATTAL et al., 2015, LARA et LABRADOR, 2013 et PREECE et al., 2009b, la reconnaissance d'activités suit une série d'étapes, série décrite sur la figure 2.1 et que nous proposons de détailler dans les sections suivantes.

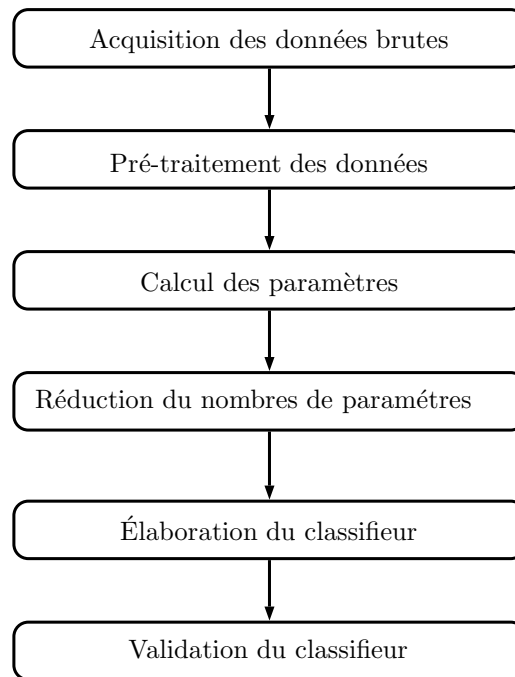


FIGURE 2.1 – Méthodologie générale pour la reconnaissance d’activités

- La première étape est la collecte d’un grand nombre de données des différentes activités ciblées. Les types de données utilisés dans la reconnaissance d’activités sont décrits dans la section 2.2.
- Ensuite, un pré-traitement des données est effectué sur les données. La technique la plus utilisée est le fenêtrage des données, fenêtrage consistant à découper le signal en différents tronçons. Le pré-traitement des données est décrit dans la section 2.4.
- Par la suite, différents paramètres sont calculés sur chaque fenêtre pour ensuite être utilisés par des algorithmes d’apprentissages. Dans la section 2.5 sont exposés les paramètres les plus souvent utilisés dans la littérature.
- La section 2.6 est dédiée aux algorithmes d’apprentissages les plus utilisés pour la reconnaissance d’activités.
- Pour finir, les approches pour l’évaluation et la comparaison des performances des algorithmes sont présentées dans la section 2.7.

2.2 Les données brutes

2.2.1 Les données accélérométriques

Les données brutes sont en général collectées par un accéléromètre se trouvant soit sur un smartphone, soit sur un objet connecté dédié. L'avantage d'utiliser un smartphone est de collecter les données d'un utilisateur de manière discrète à l'aide d'une application, sans rajouter un matériel supplémentaire. A noter cependant que l'utilisateur risque de modifier la position de son smartphone assez souvent, l'utilisation d'une plateforme dédiée a pour avantage de pouvoir positionner et fixer le capteur à un endroit précis du corps. Les différents placements du capteurs sont décrits dans la section 2.3 et le type d'équipement utilisé dans les études est décrit dans le tableau 2.1.

Dans le cas de l'utilisation d'objets connectés, un seul ou plusieurs accéléromètres peuvent être utilisés. L'avantage d'utiliser plusieurs capteurs est d'avoir plus d'information si les capteurs sont placés à différents endroits. Dans CLELAND et al., 2013, sept capteurs sont placés à des endroits différents du corps et les 63 combinaisons possibles entre ces capteurs sont étudiées. Avec un seul capteur, la précision de la reconnaissance d'activités est de 96.60% en moyenne. Le meilleur résultat est obtenu avec la combinaison des données de trois capteurs qui donne une précision de 97.66% en moyenne.

L'accéléromètre fourni 3 données différentes :

- a_x qui représente l'accélération suivant l'axe x.
- a_y qui représente l'accélération suivant l'axe y.
- a_z qui représente l'accélération suivant l'axe z.

Un exemple de données fournit par accéléromètre est présenté dans la figure 2.2.

Dans le cas, où le capteur a une position et une orientation fixes, les données de chaque axe peuvent être utilisées directement pour faire de la reconnaissance d'activités. Si le capteur n'a pas de position et d'orientation fixes, les valeurs des accélérations a_x , a_y et a_z sont différentes en fonction de l'orientation du capteur. Il est alors possible d'utiliser la norme de l'accélération définie par :

$$a_n = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2.1)$$

TABLE 2.1 – Type d'équipement utilisé pour la collecte de données

Type d'équipement	publication(s)
Smartphone	BAYAT, POMPLUN et TRAN, 2014 CHEN et SHEN, 2017 GU et al., 2015 JAIN et KANHANGAD, 2018 KIM, CHO et KIM, 2012 NI, ZHANG et LI, 2018 PRASERTSUNG et HORANONT, 2016 RAHIM et al., 2018 SAN-SEGUNDO et al., 2018 VYAS, WALSE et DHARASKAR, 2017 WANG et al., 2018
Objets connectés	ATALLAH et al., 2011 CHAPRON et al., 2018 DAVILA, CRETU et ZAREMBA, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GUPTA et DALLAS, 2014 GHAZALI et al., 2018 HE et JIN, 2008 HE et JIN, 2009 KHAN et al., 2010 KHAN, LEE et KIM, 2008 LEE et al., 2011 PREECE et al., 2009a MEHRANG, PIETILA et KORHONEN, 2018 YANG, WANG et CHEN, 2008 ZHANG et SAWCHUK, 2011b

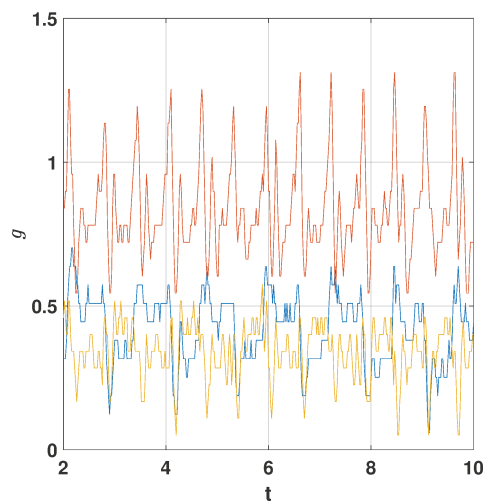


FIGURE 2.2 – Exemple de données d'un accéléromètre pendant la marche

L'avantage de travailler avec la norme est que le capteur n'a pas besoin d'avoir une position fixe. En pratique, c'est un cas qu'on rencontre fréquemment. Par exemple, si un accéléromètre d'un smartphone est utilisé, le téléphone est souvent situé dans la poche, son orientation est donc non déterminée. Dans ce cas, il est préférable donc d'utiliser la norme plutôt que directement l'accélération suivant les axes de l'accéléromètre. Cependant, l'utilisation de la norme a pour défaut de perdre des informations qui peuvent être déterminantes dans la reconnaissance d'activités. Une solution pour pouvoir quand même utiliser les valeurs des axes dans le cas où le capteur n'est pas fixe est de déterminer l'orientation du capteur. Si l'orientation est connue, un changement de repère permet d'utiliser l'accélération suivant les axes. Une méthode simple est de calculer l'orientation du capteur à l'aide de l'accélération terrestre, mais cette méthode requiert que la composante dynamique soit filtrée. Une autre solution est d'utiliser d'autres capteurs comme un magnétomètre ou un gyroscope. Par exemple, YURTMAN, BARSHAN et FIDAN, 2018 propose une méthode qui permet de d'obtenir l'accélération d'un capteur suivant 3 axes indépendamment de son orientation et à l'aide d'un magnétomètre.

Chaque accélération peut être décomposée en une composante statique due à la gravité et une composante dynamique due au mouvement de l'utilisateur :

$$a_i = a_{ig} + a_{id} \quad (2.2)$$

où a_i peut être a_x , a_y , a_z ou a_n , où a_{ig} est la composante statique du signal a_i et a_{id} sa composante dynamique.

La composante statique est utilisée en général pour déterminer l'orientation et la partie dynamique est utilisée pour déterminer le mouvement de l'utilisateur. Un seul signal ou l'ensemble des signaux peut être utilisé comme entrée pour la reconnaissance d'activités. Par exemple, l'accélération, la norme, la composante statique et la composante dynamique sont utilisées pour faire de la reconnaissance d'activités dans CASALE, PUJOL et RADEVA, 2011.

Il est possible de calculer les angles de rotation permettant de définir l'orientation de l'accéléromètre dans le repère du capteur à l'aide des accélérations des trois axes de la

manière suivante :

$$yaw = atan\left(\frac{a_z}{\sqrt{a_x^2 + a_y^2}}\right) \quad (2.3)$$

$$pitch = atan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (2.4)$$

$$roll = atan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \quad (2.5)$$

Si le capteur à une orientation et une position fixe sur l'utilisateur, il est donc possible de déterminer la position de l'utilisateur à l'aide de ces trois angles de rotation et peut permettre notamment de déterminer si la personne est allongée ou debout. Les angles de rotation sont utilisés pour faire de la reconnaissance d'activités dans DAVILA, CRETU et ZAREMBA, 2017, KIM, CHO et KIM, 2012, LEE et al., 2011, KHAN, LEE et KIM, 2008 et NI, ZHANG et LI, 2018.

Si le capteur peut tourner seulement sur un axe, on peut utiliser les normes calculées sur un plan comme suit :

$$a_{xy} = \sqrt{a_x^2 + a_y^2} \quad (2.6)$$

$$a_{yz} = \sqrt{a_y^2 + a_z^2} \quad (2.7)$$

$$a_{xz} = \sqrt{a_x^2 + a_z^2} \quad (2.8)$$

La norme a_{xz} est par exemple utilisée dans GUPTA et DALLAS, 2014. Cette norme est utilisée en raison de l'emplacement particulier du capteur : l'accéléromètre est situé sur la ceinture de l'utilisateur. Le capteur peut tourner librement autour de la taille de l'utilisateur, mais l'axe y reste parallèle au tronc.

2.2.2 Fréquence

La fréquence d'échantillonnage est un paramètre important qui aura un impact sur le temps de calcul et la précision de la reconnaissance d'activités. Une fréquence d'échantillonnage élevée va permettre de gagner en précision, mais va nécessiter un temps de calcul plus important. La fréquence d'échantillonnage doit aussi être choisie en fonction du spectre des mouvements à reconnaître. D'après BOUTEN et al., 1997, une fréquence

d'échantillonnage de 20 Hz est suffisante pour pouvoir capturer les mouvements quotidiens d'une personne. Dans ANTONSSON et MANN, 1985, il est montré que le spectre des activités est comprise entre 0 et 20 Hz et la majeure partie de l'énergie du signal se situe en dessous de 3 Hz. Le tableau 2.2 énumère les différentes fréquences d'échantillonnage utilisées dans la littérature. La fréquence d'échantillonnage de l'accéléromètre est souvent comprise entre 10 et 100 Hz.

2.3 Placement des capteurs

Le placement des capteurs est une composante importante à prendre en compte dans la reconnaissance d'activités. Il n'existe pas aujourd'hui de consensus sur un placement optimal et plusieurs positionnements sont proposés dans la littérature. Dans le cas d'un objet connecté dédié, le positionnement du capteur est souvent motivé par la nature de l'activité suivie. Une revue des différents placements du capteur est présentée dans le tableau 2.3. En général, la position du capteur n'est pas fixe pour les smartphones et peut varier selon l'activité de l'utilisateur. Par exemple, quatre types de poses du smartphone sont prises en compte dans WANG et al., 2018 : le smartphone dans la poche, le smartphone à côté de la tête durant un appel, le smartphone qui suit le mouvement de la main durant un mouvement comme la marche et le smartphone dans la main, devant l'utilisateur, durant son utilisation. Dans certains cas, le smartphone doit avoir un emplacement précis. Par exemple, le smartphone doit être positionné au niveau de la taille à l'aide d'une ceinture dans PRASERTSUNG et HORANONT, 2016 et le smartphone est positionné au niveau du dos dans MITCHELL, MONAGHAN et O'CONNOR, 2013.

Il existe des études comparatives pour étudier l'impact du positionnement du capteur. Dans ATALLAH et al., 2011, une étude est faite sur le meilleur positionnement pour suivre les activités de la vie quotidienne (le couché, la cuisine, manger ou boire, la cuisine, la lecture, s'habiller, la marche, la course, le cyclisme). Pour déterminer la meilleure position des capteurs comportant un accéléromètre sont placés sur sept parties différentes du corps (pied, genoux, taille, torse, bras, poignet et tête) sur neuf personnes différentes. Ensuite, un ensemble de paramètres est calculé pour chaque position. Des paramètres sont ensuite sélectionnés pour être utilisés pour faire de la reconnaissance d'activités avec deux algorithmes de classification (les k plus proches voisins et la classification naïve bayésienne). Les résultats de cette étude montrent que les paramètres qui permettent de mieux discriminer les mouvements varient selon le placement du capteur. La précision

TABLE 2.2 – Fréquence d'échantillonnage utilisé pour la reconnaissance d'activité

Fréquence d'échantillonnage (Hz)	publication(s)
10 Hz	FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 KHAN, LEE et KIM, 2008
15 Hz	PANNURAT et al., 2017
16 Hz	MITCHELL, MONAGHAN et O'CONNOR, 2013
20 Hz	CHEN et SHEN, 2017 KHAN et al., 2010 LEE et al., 2011
25 Hz	MEHRANG, PIETILA et KORHONEN, 2018 MITCHELL, MONAGHAN et O'CONNOR, 2013
30 Hz	DAVILA, CRETU et ZAREMBA, 2017
32 Hz	GU et al., 2015
50 Hz	ATALLAH et al., 2011 CHAPRON et al., 2018ERMES, PARKKA et CLUITMANS, 2008 JAIN et KANHANGAD, 2018 RAHIM et al., 2018 SAN-SEGUNDO et al., 2018 WANG et al., 2011a WANG et al., 2018
51,2 Hz	CLELAND et al., 2013
52 Hz	CASALE, PUJOL et RADEVA, 2011
68 Hz	PREECE et al., 2009a
100 Hz	BAYAT, POMPLUN et TRAN, 2014 DUTTA et al., 2018 HE et JIN, 2008 HE et JIN, 2009 PRASERTSUNG et HO- RANONT, 2016 YANG, WANG et CHEN, 2008 ZHANG et SAWCHUK, 2011b
102.4 Hz	NI, ZHANG et LI, 2018
126 Hz	GUPTA et DALLAS, 2014

TABLE 2.3 – placement du capteur

positionnement	publication(s)
poignet	ATALLAH et al., 2011 CLELAND et al., 2013 DAVILA, CRETU et ZAREMBA, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GU et al., 2015 MEHRANG, PIETILA et KORHONEN, 2018 PANNURAT et al., 2017 SAN-SEGUNDO et al., 2018 YANG, WANG et CHEN, 2008
cheville	FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 PREECE et al., 2009a
bras	ATALLAH et al., 2011 DAVILA, CRETU et ZAREMBA, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 PANNURAT et al., 2017
cuisse	CLELAND et al., 2013 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 PANNURAT et al., 2017
taille	ATALLAH et al., 2011 CLELAND et al., 2013 DAVILA, CRETU et ZAREMBA, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GUPTA et DALLAS, 2014 HE et JIN, 2008 HE et JIN, 2009 KIM, CHO et KIM, 2012 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 PRA-SERTSUNG et HORANONT, 2016 PREECE et al., 2009a WANG et al., 2011a ZHANG et SAWCHUK, 2011b
torse	ATALLAH et al., 2011 CASALE, PUJOL et RADEVA, 2011 CLELAND et al., 2013 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GHAZALI et al., 2018 KHAN et al., 2010 KHAN, LEE et KIM, 2008 LEE et al., 2011 PANNURAT et al., 2017
dos	CLELAND et al., 2013 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017
genoux	ATALLAH et al., 2011 DAVILA, CRETU et ZAREMBA, 2017 ERMES, PARKKA et CLUITMANS, 2008 PREECE et al., 2009a
tête	ATALLAH et al., 2011 MITCHELL, MONAGHAN et O'CONNOR, 2013 PANNURAT et al., 2017
pied	ATALLAH et al., 2011 CLELAND et al., 2013 DAVILA, CRETU et ZAREMBA, 2017

de la classification varient aussi en fonction du placement du capteur. Par exemple, les meilleurs résultats pour la classification de la marche ont été observés avec un capteur positionné au niveau de la poitrine ou du poignet.

Une étude similaire est faite dans CLELAND et al., 2013. Dans cette étude, des capteurs sont placés à six endroits différents (poitrine, pied, bas du dos, hanche, cuisse et poignet) et des données sont récoltées sur huit personnes différentes. Les données brutes de sept différentes activités (le repos, la course, monter un escalier, descendre un escalier, la position debout et la marche) sont collectées. Sur le même principe que l'étude précédente, un ensemble de paramètres est calculé à partir de chaque capteur pour être utilisé par quatre algorithmes de classification différents (apprentissage par arbres de décisions, Naïve bayésienne, réseaux de neurones et machines à vecteurs de support). Cette étude montre que l'algorithme des machines à vecteurs de support donne les meilleurs résultats de reconnaissance d'activités sur tous les emplacements à l'exception du pied. La meilleure précision est obtenue avec le capteur fixé à la hanche et les plus mauvais résultats sont observés quand le capteur est fixé au genou ou au pied.

Une dernière étude, plus récente, proposé dans PANNURAT et al., 2017 traite ce sujet dans le cas où les capteurs sont placés à la tête, au bras, au poignet, à la cheville, au torse, à la cuisse, sur le côté et sur le devant de la taille. Les activités étudiées sont la position assise, couché sur le dos, couché sur le côté gauche, couché sur le côté droit, couché sur le ventre, la position debout et la marche. Le but cette étude est de déterminer la position optimale pour faire de la reconnaissance d'activités. 19 paramètres différents sont étudiés et 8 algorithmes différents de classification sont utilisés. Le meilleur résultat est obtenu avec le capteur positionné sur le côté de la taille sur deux jeux de données : un jeu de données obtenu avec des personnes jeunes et un jeu de données obtenu avec des personnes âgées.

2.4 Le pré-traitement des données

2.4.1 Filtrage

Les signaux fournis par les capteurs peuvent être bruités et un filtre passe-bas doit souvent être appliqué pour éliminer le bruit. Comme vu auparavant, dans certains cas, il est choisi de filtrer la gravité de d'accélération en appliquant un filtre passe-haut. Par

exemple, la gravité est filtrée à l'aide d'un filtre de Butterworth dans PRASERTSUNG et HORANONT, 2016 et un filtre passe bande est utilisé dans DAVILA, CRETU et ZAREMBA, 2017. Une étude est proposée dans WANG et al., 2011b pour déterminer le filtre qui permet d'améliorer le mieux les résultats de classification. Plusieurs types de filtres sont mis en oeuvre et comparés comme le filtre médian, le filtre de Butterworth et le filtre de Kalman.

2.4.2 Fenêtrage

Une étape de fenêtrage est souvent utilisée dans le domaine de la reconnaissance d'activités. Cette étape correspond à la segmentation du signal en fenêtre. Une activité et un vecteur de paramètres sont ensuite attribués à chaque fenêtre pour être utilisés par des algorithmes de classification. Un état de l'art sur les différentes techniques de fenêtrages est proposé dans BANOS et al., 2014. Il existe trois différents types de fenêtrage : le fenêtrage défini par l'activité, le fenêtrage défini par événement et le fenêtrage par fenêtre glissante.

Le fenêtrage défini par l'activité consiste à déterminer les fenêtres en détectant les changements d'activités. Ensuite, un algorithme de reconnaissance d'activités est appliqué pour déterminer le type d'activité sur la fenêtre. Plusieurs techniques sont possibles pour déterminer le changement d'activité. Ceci peut être réalisé par une analyse fréquentielle ou comme dans SEKINE et al., 1998 par une analyse basée sur la transformée en ondelettes.

Le fenêtrage par événement a pour principe qu'un événement extérieur (par exemple, un chronomètre ou un individu) indique le début et la fin d'une fenêtre. Par exemple, des interrupteurs fixés sur un pied permettent de déterminer les fenêtres dans JASIEWICZ et al., 2006. Cette méthode est utilisée presque exclusivement en laboratoire car il est difficile de la mettre en oeuvre en situation réelle.

La fenêtrage glissant est la technique la plus utilisée dans la littérature pour la reconnaissance d'activités, car elle ne nécessite pas de pré-traitement de données et elle est facile d'utilisation en temps réel. Cette technique consiste à découper le signal à analyser en une suite de fenêtres de taille fixe. Un recouvrement peut être utilisé, il peut varier de 10% à 90%. Un recouvrement de 50% est par exemple utilisé dans HE et JIN, 2008.

Ce fenêtrage est particulièrement utilisé pour la reconnaissance d'activités périodiques (par exemple la marche ou la course) et pour la reconnaissance d'activités statiques (par exemple le repos). Les différentes tailles de fenêtres utilisées dans la littérature sont listées dans le tableau 2.4. La taille de la fenêtre a un impact sur la précision de la reconnaissance d'activités. Ainsi, dans BANOS et al., 2014 une analyse de l'impact de taille est réalisée. Trois ensembles de paramètres sont calculés sur des fenêtres allant de 0.25 secondes à 7 secondes. Le premier ensemble est composé de la moyenne, le second ensemble est composé de la moyenne et de l'écart-type et le dernier ensemble de paramètre est composé de la moyenne, l'écart-type, le maximum, le minimum et le nombre de fois que le signal traverse la moyenne. Ensuite, quatre algorithmes de classification (arbre de décisions, k plus proche voisins, classification naïve bayésienne et K-moyennes) sont appliqués sur les trois différents jeux de paramètres pour déterminer la longueur de fenêtre qui donne des meilleurs résultats. Cette étude montre que les fenêtres de longueurs entre 0.25 secondes et 2 secondes donnent les meilleures performances pour la reconnaissance d'activités et qu'il n'est donc pas nécessaire d'utiliser de large fenêtres pour effectuer de la reconnaissance d'activités. Dans une autre étude MEHRANG, PIETILA et KORHONEN, 2018, l'algorithme des forêts aléatoires est utilisé pour différencier cinq activités (le cyclisme haute et basse intensité, le repos, la position assise, et l'activité ménagère). Le but de cette étude est de trouver la taille optimale de la fenêtre glissante entre des fenêtres allant de 1 seconde à 21 secondes avec 90% de recouvrement. Les meilleurs résultats sont obtenus avec une fenêtre de 13 secondes. Dans WANG et al., 2018, les données des capteurs d'un smartphone sont utilisées pour faire de la reconnaissance d'activités. Huit mouvements sont étudiés : la marche, le repos, la descente d'escalier, la montée d'escalier, la montée d'ascenseur, la descente d'ascenseur, la montée d'escalateur et la descente d'escalateur. Cinq différents algorithmes sont comparés (arbre de décision, machine à vecteurs de support, k plus proches voisins, Naïve bayésienne et adaptive boosting). La fenêtre optimale pour reconnaître les mouvements de cette étude se situe entre 2.5 et 3.5 secondes.

2.5 Extraction et sélection de paramètres de classification

Les données de l'accéléromètre peuvent être directement utilisées par l'algorithme de classification. Par exemple, les données des 3 axes de l'accéléromètre sont utilisées directement avec l'algorithme de classification du modèle de Markov caché dans WANG et al., 2011a. Cependant, dans la majorité des cas, un ensemble de paramètres est calculé

TABLE 2.4 – Longueur de fenêtres utilisées dans la reconnaissance d'activité

taille de la fenêtrés	publication(s)
1s	CHEN et SHEN, 2017 CASALE, PUJOL et RADEVA, 2011 PANNURAT et al., 2017
1.28s	BAYAT, POMPLUN et TRAN, 2014 PRASERTSUNG et HORANONT, 2016
2s	CHAPRON et al., 2018 GU et al., 2015 ZHANG et SAWCHUK, 2011b
2.56s	GHAZALI et al., 2018 JAIN et KANHANGAD, 2018 NI, ZHANG et LI, 2018 RAHIM et al., 2018
3s	SAN-SEGUNDO et al., 2018
3.2s	KHAN et al., 2010
5s	ATALLAH et al., 2011
5,12s	HE et JIN, 2008 HE et JIN, 2009 YANG, WANG et CHEN, 2008
6s	CHAPRON et al., 2018 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017
10s	LEE et al., 2011 CLELAND et al., 2013 DUTTA et al., 2018
13s	MEHRANG, PIETILA et KORHONEN, 2018

à partir des données d'une fenêtre. Cet ensemble de paramètres va constituer un vecteur qui va être ensuite utilisé pour la classification. L'objectif est ici de sélectionner parmi ces paramètres ceux qui permettent de mieux discriminer les différentes activités.

Il est possible d'utiliser un grand nombre de paramètres ou combinaisons de paramètres, cependant, il est préférable d'en utiliser le moins possible car si on utilise des variables inappropriées ou redondantes, on augmente le nombre de calculs et on réduit les performances des algorithmes de classification. Il existe un très grand nombre de paramètres possibles. Les articles suivants énumèrent les paramètres souvent utilisés en reconnaissance d'activités : ATTAL et al., 2015, FIGO et al., 2010 et PREECE et al., 2009b. Ces paramètres peuvent être calculés à partir de l'accélération de chaque axe a_x , a_y et a_z , de la norme a_n , des angles de rotation ou des normes définies sur un plan a_{xy} , a_{yz} et a_{xz} . Les paramètres les plus utilisés sont présentés dans la suite de cette section, ils sont répertoriés dans le tableau 2.5 et 2.6. Dans la suite, s désigne le signal sur lequel sont calculés ces paramètres (il peut s'agir de a_x , a_y , etc.), N est le nombre d'échantillons disponibles sur la fenêtre et s_t est la valeur de l'échantillon du signal s à l'instant t .

2.5.1 Liste des paramètres

2.5.1.1 Moyenne, variance et écart type

Un des paramètres les plus simples et plus intuitifs est la moyenne. Le calcul de ce paramètre nécessite très peu de calculs. Il permet aussi de lisser le signal et d'atténuer le bruit. C'est un des paramètres les plus souvent utilisés dans la littérature. La moyenne est définie de la manière suivante :

$$\mu = \frac{1}{N} \sum_{t=1}^N s(t) \quad (2.9)$$

La variance est aussi une variable souvent utilisée pour la reconnaissance d'activités. Cette variable sert à caractériser la dispersion des données. L'écart-type correspond à la racine carrée de la variance et donne les mêmes informations que la variance. La variance ou l'écart-type seront moins intéressants à utiliser si le signal contient de nombreuses valeurs aberrantes, celles-ci ayant un impact important sur le calcul de ces paramètres.

TABLE 2.5 – paramètres utilisés pour la reconnaissance d'activité

Paramètres	Publication(s)
Moyenne	ATALLAH et al., 2011 BAYAT, POMPLUN et TRAN, 2014 CASALE, PUJOL et RADEVA, 2011 CHAPRON et al., 2018 CHEN et SHEN, 2017 CLELAND et al., 2013 DUTTA et al., 2018 ERMES, PARKKA et CLUITMANS, 2008 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GUPTA et DALLAS, 2014 GHAZALI et al., 2018 KIM, CHO et KIM, 2012 KHAN et al., 2010 LEE et al., 2011 MEHRANG, PIETILA et KORHONEN, 2018 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 PRASERTSUNG et HORANONT, 2016 PREECE et al., 2009a WANG et al., 2018
Variance	ATALLAH et al., 2011 BAYAT, POMPLUN et TRAN, 2014 ERMES, PARKKA et CLUITMANS, 2008 GU et al., 2015 GUPTA et DALLAS, 2014 PRASERTSUNG et HORANONT, 2016 WANG et al., 2018
Écart type	BAYAT, POMPLUN et TRAN, 2014 CASALE, PUJOL et RADEVA, 2011 CHAPRON et al., 2018 CHEN et SHEN, 2017 CLELAND et al., 2013 DUTTA et al., 2018 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GHAZALI et al., 2018 KHAN et al., 2010 KIM, CHO et KIM, 2012 LEE et al., 2011 MEHRANG, PIETILA et KORHONEN, 2018 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 PRASERTSUNG et HORANONT, 2016 PREECE et al., 2009a RAHIM et al., 2018 WANG et al., 2018
Moyenne quadratique	ATALLAH et al., 2011 BAYAT, POMPLUN et TRAN, 2014 CASALE, PUJOL et RADEVA, 2011 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GU et al., 2015 NI, ZHANG et LI, 2018 PRASERTSUNG et HORANONT, 2016
Zone d'amplitude du signal (SMA)	FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 KHAN et al., 2010 KHAN, LEE et KIM, 2008 KIM, CHO et KIM, 2012 LEE et al., 2011 MEHRANG, PIETILA et KORHONEN, 2018 NI, ZHANG et LI, 2018 YANG, WANG et CHEN, 2008
Entropie	ATALLAH et al., 2011 ERMES, PARKKA et CLUITMANS, 2008 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GUPTA et DALLAS, 2014 KHAN et al., 2010 LEE et al., 2011 NI, ZHANG et LI, 2018 PREECE et al., 2009a RAHIM et al., 2018
temps entre deux pics	BAYAT, POMPLUN et TRAN, 2014 PRASERTSUNG et HORANONT, 2016
nombre de pic	BAYAT, POMPLUN et TRAN, 2014 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 PRASERTSUNG et HORANONT, 2016
Zero crossing mode	CHAPRON et al., 2018

TABLE 2.6 – paramètres utilisés pour la reconnaissance d’activité

Corrélation	BAYAT, POMPLUN et TRAN, 2014 CASALE, PUJOL et RADEVA, 2011 CHAPRON et al., 2018 CHEN et SHEN, 2017 CLELAND et al., 2013 KHAN et al., 2010 KIM, CHO et KIM, 2012 LEE et al., 2011 MEHRANG, PIETILA et KORHONEN, 2018 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 PRASERTSUNG et HORANONT, 2016 PREECE et al., 2009a
Éventail	ATALLAH et al., 2011 BAYAT, POMPLUN et TRAN, 2014 GU et al., 2015 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 PRASERTSUNG et HORANONT, 2016
Écart absolu moyen	GUPTA et DALLAS, 2014 WANG et al., 2018
Minimum	CHEN et SHEN, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GHAZALI et al., 2018 PANNURAT et al., 2017 PRASERTSUNG et HORANONT, 2016 RAHIM et al., 2018
Maximum	CHEN et SHEN, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GU et al., 2015 GHAZALI et al., 2018 PANNURAT et al., 2017 PRASERTSUNG et HORANONT, 2016 RAHIM et al., 2018
Médiane	MEHRANG, PIETILA et KORHONEN, 2018 PREECE et al., 2009a WANG et al., 2018
Percentile	CHEN et SHEN, 2017, DUTTA et al., 2018, WANG et al., 2018
Énergie	ATALLAH et al., 2011 CHAPRON et al., 2018, CLELAND et al., 2013 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 NI, ZHANG et LI, 2018 PRASERTSUNG et HORANONT, 2016 PREECE et al., 2009a YANG, WANG et CHEN, 2008
Asymétrie	ATALLAH et al., 2011 CASALE, PUJOL et RADEVA, 2011 CHAPRON et al., 2018 CLELAND et al., 2013 DUTTA et al., 2018 GU et al., 2015 GUPTA et DALLAS, 2014 KIM, CHO et KIM, 2012 NI, ZHANG et LI, 2018 RAHIM et al., 2018
Kurtosis	ATALLAH et al., 2011 CASALE, PUJOL et RADEVA, 2011 CHAPRON et al., 2018 CLELAND et al., 2013 DUTTA et al., 2018 NI, ZHANG et LI, 2018 RAHIM et al., 2018
Modèle AR	HE et JIN, 2008 KHAN et al., 2010 KHAN, LEE et KIM, 2008 KIM, CHO et KIM, 2012 LEE et al., 2011
Transformée en cosinus discrète	HE et JIN, 2009
Transformée en ondelette	CASALE, PUJOL et RADEVA, 2011 CHEN et SHEN, 2017 DUTTA et al., 2018 MITCHELL, MONAGHAN et O’CONNOR, 2013 PREECE et al., 2009a
Transformé de Fourier	ATALLAH et al., 2011 CHAPRON et al., 2018 CHEN et SHEN, 2017 DUTTA et al., 2018 ERMES, PARKKA et CLUITMANS, 2008 GUPTA et DALLAS, 2014 MEHRANG, PIETILA et KORHONEN, 2018 NI, ZHANG et LI, 2018 PREECE et al., 2009a WANG et al., 2018

La variance est définie de la manière suivante :

$$\sigma^2 = \frac{1}{N} \sum_{t=1}^N (s(t) - \mu_s)^2 \quad (2.10)$$

À partir de l'écart type calculé sur chacun des axes, il est possible de calculer la magnitude de l'écart type (utilisé dans PANNURAT et al., 2017). Elle est définie par :

$$MAG_\sigma = \sqrt{\sigma_{a_x}^2 + \sigma_{a_y}^2 + \sigma_{a_z}^2} \quad (2.11)$$

où $\sigma_{a_x}^2$, $\sigma_{a_y}^2$ et $\sigma_{a_z}^2$ sont les variances suivant les axes x, y et z de l'accéléromètre.

La moyenne des valeurs absolues des écarts est un autre paramètre qui est utilisé pour mesurer la dispersion (utilisée dans LARA et LABRADOR, 2013). Ce paramètre est défini par :

$$MAD = \frac{1}{N} \sum_{t=1}^N |s(t) - \mu_s| \quad (2.12)$$

Dans GUPTA et DALLAS, 2014, une fenêtre de temps de 6 secondes est utilisée. Cette fenêtre est ensuite découpée en sous-fenêtres de 0,5 s sans recouvrement. Avec ces sous-fenêtres, il est proposé de calculer la tendance moyenne comme suit :

$$\mu T = \sum_{k=2}^j (|\mu_k - \mu_{k-1}|) \quad (2.13)$$

où j est le nombre de sous-fenêtres dans la fenêtre de base, μ_k est la moyenne de l'k-ème sous-fenêtre fille.

À partir de ces sous-fenêtres, d'autres paramètres sont calculés dans GUPTA et DALLAS, 2014 comme la différence des moyennes :

$$\mu D = \sum_{k=1}^j (|\mu - \mu_k|) \quad (2.14)$$

où μ est la moyenne de la fenêtre mère.

Ces mêmes paramètres peuvent être calculés relativement à la variance :

$$\sigma T^2 = \sum_{k=2}^j (|\sigma_k^2 - \sigma_{k-1}^2|) \quad (2.15)$$

$$\sigma D^2 = \sum_{k=1}^j (|\sigma^2 - \sigma_k^2|) \quad (2.16)$$

2.5.1.2 Médiane, percentile, maximum, minimum et éventail

La médiane est la valeur qui permet de séparer un ensemble d'échantillons en deux parties égales : une partie composée de tous les échantillons avec les valeurs les plus élevées et une autre partie composée de tous les échantillons avec les valeurs les moins élevées.

Le percentile est une mesure qui indique la valeur où on peut trouver un pourcentage des observations. Par exemple, le 25ème percentile correspond à la valeur où on trouve 25% des observations en dessous de cette valeur et le 50ème percentile correspond à la médiane.

Le maximum ou le minimum sur une fenêtre temporelle sont des paramètres faciles à calculer. A partir de ces deux valeurs, il est aussi possible de calculer l'éventail qui est la différence entre le maximum et le minimum.

$$M = \max_{t \in [1;N]} s(t) \quad (2.17)$$

$$m = \min_{t \in [1;N]} s(t) \quad (2.18)$$

$$\Delta_M = M - m \quad (2.19)$$

La déviation absolue médiane mesure la variabilité d'un signal, cette mesure est utilisée dans NI, ZHANG et LI, 2018 et est définie par :

$$DAM = \text{median}_{t \in [1;N]} (|s(t) - \text{median}(s)|) \quad (2.20)$$

2.5.1.3 Corrélation

Le coefficient de corrélation est utilisé pour mesurer l'intensité de la relation linéaire entre deux signaux. Le coefficient de corrélation le plus utilisé est celui de Bravais-Pearson. Pour deux signaux s et r , avec σ_s est l'écart-type de $s(t)$ et σ_r est l'écart-type de $r(t)$, ce paramètre est défini comme suit :

$$\rho_{s,r} = \frac{1}{N\sigma_s\sigma_r} \sum_{t=1}^N s(t)r(t) \quad (2.21)$$

ou plus généralement

$$\rho_{s,r}(i) = \frac{1}{(N-i)\sigma_s\sigma_r} \sum_{t=i+1}^N s(t)r(t-i) \quad i \geq 0 \quad (2.22)$$

$\rho_{s,r}(i)$ mesure la similarité des signaux s et r décalés de i échantillons.

Le maximum de corrélation $\max_{i \in [0;N-1]} \rho_{s,r}(i)$ peut aussi être utilisé comme par exemple dans VELTINK et al., 1996

2.5.1.4 Autres paramètres dans le domaine temporel

La moyenne quadratique correspond à la racine carrée de la moyenne du signal au carré. Pour un signal de moyenne nulle, moyenne quadratique et écart type sont équivalents. Cette métrique est souvent utilisée dans la reconnaissance d'activités et est obtenue à partir de l'équation suivante :

$$RMS = \sqrt{\frac{1}{N} \sum_{t=1}^N s(t)^2} \quad (2.23)$$

Des paramètres d'actimétrie peuvent aussi être utilisés comme le zero crossing mode (ZCM). Le ZCM est le nombre de fois que le signal change de signe. Le seuil peut être remplacé par la moyenne ou un seuil quelconque dans certaines études. Ce paramètre est utilisé par exemple en analyse de sommeil (utilisé dans BLACKWELL et al., 2008). Un autre paramètre utilisé en actimétrie est le time above threshold (TAT) qui correspond au nombre de points au-dessus d'un seuil.

D'autres paramètres peuvent être calculés à l'aide des pics qui correspondent à des points dont la valeur est supérieure à celles du point suivant et du point précédent. Le nombre de pic d'une fenêtre ou le temps moyen entre deux pic peuvent être utilisés comme paramètres. Ces paramètres sont utiles pour la caractérisation des activités périodiques comme la marche ou la course. Cependant, les paramètres qui utilisent les pics sont très sensibles au bruit.

La zone d'amplitude du signal correspond à la moyenne des valeurs absolues des accélérations a_x , a_y et a_z sur les 3 axes et sur une période de temps. Il est défini par :

$$SMA = \frac{1}{N} \left(\sum_{t=1}^N |a_x(t)| + \sum_{t=1}^N |a_y(t)| + \sum_{t=1}^N |a_z(t)| \right) \quad (2.24)$$

Le coefficient d'asymétrie et le kurtosis sont des paramètres de forme. Le coefficient d'asymétrie permet de mesurer la symétrie du signal. Si le coefficient est négatif, le signal sera plus présent sur la droite de la médiane et si le coefficient est positif, le signal sera plus présent sur la gauche de la médiane. Le kurtosis, ou coefficient d'amortissement, permet de mesurer la dispersion des valeurs du signal. Le coefficient d'asymétrie et le kurtosis sont définis par :

$$coef\ asymétrie = \frac{N^2}{(N-1)(N-2)} \cdot \left(\sum_{t=1}^N \frac{(s(t) - \mu)^3}{\sigma^3} \right) \quad (2.25)$$

$$kurtosis = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \cdot \left(\sum_{t=1}^N \frac{(s(t) - \mu)^4}{\sigma^4} - \frac{3(N-1)^2}{(N-2)(N-3)} \right) \quad (2.26)$$

2.5.1.5 La Transformation de Fourier discrète (TFD)

La plupart des activités génèrent des signaux périodiques comme la marche ou la course. Il peut donc être utile de réaliser une analyse dans le domaine fréquentiel pour différencier les activités. L'outil principal utilisé pour l'analyse fréquentielle est la transformation de Fourier discrète qui permet de calculer une représentation spectrale discrète

d'un signal. La TFD est définie par :

$$S(f) = \sum_{t=1}^N s(t)e^{-j2\pi f(t-1)/N} \quad (2.27)$$

où $f \in [0, N - 1]$ désigne la fréquence réduite.

Le calcul de la TFD est généralement réalisé via la transformée de Fourier rapide (pour réduire la charge de calcul). Tous les coefficients de la TFD ne doivent pas nécessairement être analysés. Seuls certains de ces coefficients peuvent être choisis selon leur utilité à discriminer certains mouvements. À partir de la TFD, il est possible de calculer d'autres variables utiles à la classification. L'énergie est une variable souvent utilisée. Elle est calculée à l'aide des coefficients de la TFD de la manière suivante :

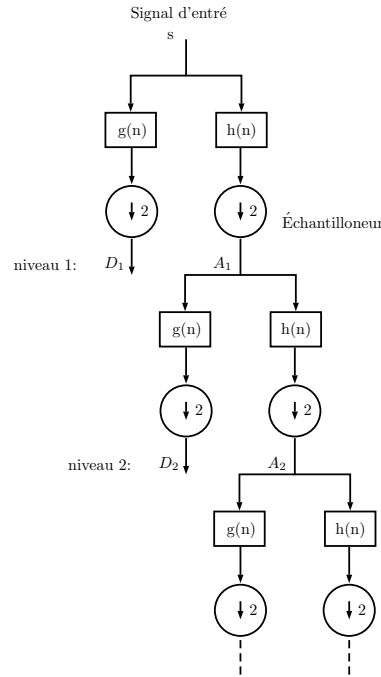
$$E = \frac{1}{N} \sum_{f=0}^{N-1} |S(f)|^2 = N^2 RMS^2 \quad (2.28)$$

L'entropie peut être utilisée pour différencier deux mouvements qui ont une signature différente. Elle se calcule comme suit :

$$H(f) = \frac{1}{N} \sum_{f=0}^{N-1} |S(f)|^2 \log(|S(f)|^2) \quad (2.29)$$

2.5.1.6 Transformée en ondelettes discrète

Comme pour la transformée de Fourier, la Transformée en ondelettes discrète permet de décomposer un signal avec la particularité de réaliser cette décomposition dans le domaine temps et fréquence. Une ondelette est une forme d'onde avec une valeur moyenne nulle et d'une durée limitée. La transformée en ondelettes discrète décompose le signal en deux parties à partir d'un ensemble de filtres passe-haut et passe-bas : une partie qui correspond à l'approximation du signal (obtenue par des composantes basses fréquences) et une partie qui correspond aux détails du signal (obtenus par des composantes hautes fréquences). Un échantillonneur est ensuite utilisé pour diviser par deux la fréquence d'échantillonnage : pour une entrée $s(t)$, la sortie de l'échantillonneur est $s(2t)$. Les sorties du filtre passe-bas correspondent aux coefficients d'approximation et les sorties du filtre passe-haut correspondent aux coefficients des détails du signal. Cette procédure peut être répétée sur les détails du signal jusqu'à ce qu'une échelle maximale soit atteinte. Cette

FIGURE 2.3 – Décomposition du signal s avec la transformée en ondelettes

procédure est illustrée dans la figure 2.3. Les coefficients d'approximation et de détails sont définis par :

$$D_{j+1}(i) = \sum_{l=0}^{L-1} g(l)A_j(2i+l) \quad (2.30)$$

$$A_{j+1}(i) = \sum_{l=0}^{L-1} h(l)A_j(2i+l) \quad (2.31)$$

où i est le niveau de décomposition en ondelette, $g(\cdot)$ est un filtre passe-bas et $h(\cdot)$ est un filtre passe-haut, L est la nombre d'éléments de la réponse impulsionnelle de ces filtres. Ils dépendent de l'ondelette mère choisie. Il est possible d'utiliser différentes ondelettes, les plus utilisées sont les ondelettes de Daubechies, Symelet, Coiflet, biorthogonale et biorthogonale réversible. A_0 correspond au signal de départ s à analyser.

La transformée en ondelette permet ainsi de générer un ensemble de vecteurs A_j et $D_j, j \in [0; j_{max}]$ qui peuvent être utilisés comme paramètres pour la classification (BRUCE, KOGER et LI, 2002, BURRUS, GOPINATH et GUO, 1998, DAUBECHIES, 1992 et PITTNER et KAMARTHI, 1999). Il est aussi possible de calculer d'autres paramètres à partir des A_j

et D_j : dans MITCHELL, MONAGHAN et O'CONNOR, 2013 la variance des A_j et D_j est utilisée pour la classification.

L'énergie de la transformée en ondelette E_T est définie par :

$$E_T = A_{j_{max}} A_{j_{max}}^T + \sum_{j=1}^{j_{max}} D_j D_j^T \quad (2.32)$$

EDR_A et EDR_{Dj} sont les rapports énergétiques des coefficients d'approximation et des coefficients de détail utilisés dans MITCHELL, MONAGHAN et O'CONNOR, 2013. Ils sont définis par :

$$EDR_A = \frac{A_{j_{max}} A_{j_{max}}^T}{E_T} \quad (2.33)$$

$$EDR_{Dj} = \frac{D_j D_j^T}{E_T} \text{ avec } j = 1, \dots, j_{max} \quad (2.34)$$

Différents paramètres obtenus à partir de la transformée en ondelette sont utilisés dans PREECE et al., 2009a pour faire de la reconnaissance d'activités à partir de données accélérométriques.

2.5.1.7 Modèle auto-régressif (AR)

Un modèle AR permet une modélisation structurelle du signal. Ce modèle établit la relation de récurrence entre échantillons. Il est défini par l'équation suivante :

$$s(t) = -\phi_1 s(t-1) - \dots - \phi_p s(t-p) + e_t \quad (2.35)$$

où e est un bruit blanc de variance σ_e^2 (le bruit générateur). p est l'ordre du modèle et les paramètres ϕ_i correspondent aux paramètres du modèle. On définit le vecteur des coefficients $\theta \in \mathbb{R}^p$ du modèle AR par

$$\theta = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix} \quad (2.36)$$

Ce vecteur peut être calculé à partir des équations de Yules Walker :

$$\theta = M(\rho_s(i))^{-1}N(\rho_s(i)) \quad (2.37)$$

avec

$$N(\rho_s(i)) = - \begin{pmatrix} \rho_s(1) \\ \rho_s(2) \\ \vdots \\ \rho_s(p) \end{pmatrix} \quad (2.38)$$

$$M(\rho_s(i)) = \begin{pmatrix} \rho_s(0) & \rho_s(1) & \rho_s(2) & \dots & \rho_s(p-1) \\ \rho_s(1) & \rho_s(0) & \rho_s(1) & \dots & \rho_s(p-2) \\ \vdots & \vdots & \vdots & & \vdots \\ \rho_s(p-1) & \rho_s(p-2) & \rho_s(p-3) & \dots & \hat{\rho}_s(0) \end{pmatrix} \quad (2.39)$$

et

$$\rho_s(i) = \frac{1}{(N-i)\sigma_s^2} \sum_{t=i+1}^N s(t)s(t-i) \quad (2.40)$$

Les coefficients du modèle AR peuvent servir comme variables pour la classification, c'est le cas dans HE et JIN, 2008.

2.5.2 Sélection des paramètres

Il n'existe pas de consensus sur le choix des paramètres dans la littérature et il n'est pas opportun d'utiliser tous les paramètres présentés précédemment. Le choix des paramètres est souvent motivé par le type de mouvement à reconnaître et des ressources matérielles disponibles pour calculer les paramètres. Par exemple, le calcul des paramètres de la transformée de Fourier est beaucoup plus lourd que le calcul des paramètres comme le minimum, le maximum ou la moyenne. La charge de calcul de différents paramètres est exposé dans FIGO et al., 2010. En outre, il est aussi important de réduire le plus possible le nombre de paramètres pour diminuer les besoins en calcul de l'algorithme de classification et éviter les informations redondantes qui pourraient détériorer les résultats de la classification. Il existe plusieurs méthodes possibles pour la sélection des paramètres. Un ensemble de méthodes de sélection de paramètres consiste à sélectionner les paramètres les plus pertinents à partir de l'ensemble des paramètres de départ. Cet ensemble de méthodes est présenté dans DASH et LIU, 1997 et consiste à suivre la procédure présentée par la figure 2.4.

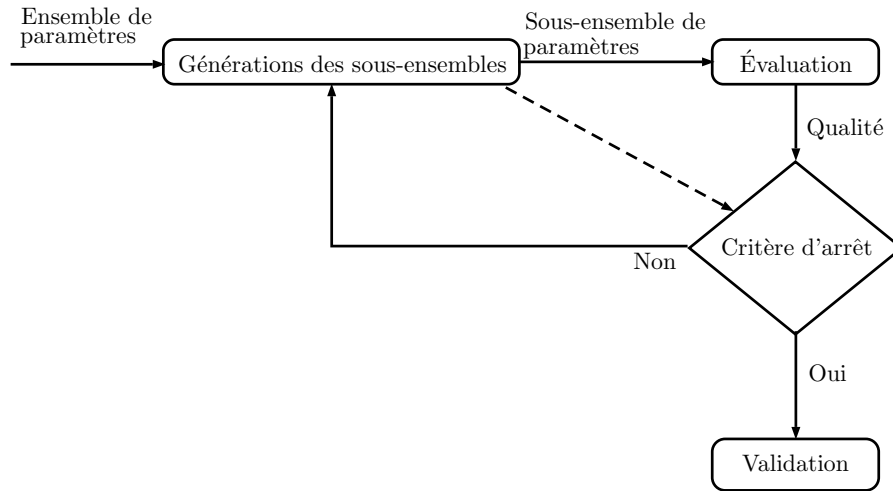


FIGURE 2.4 – Procédure pour la sélection des paramètres

Les différentes méthodes de sélection vont dépendre de l'évaluation des sous-ensembles des paramètres et de la génération de ces ensembles de paramètres. Il existe trois méthodes principales pour la génération des sous-ensembles :

- La génération exhaustive : tous les sous-ensembles de paramètres sont testés. Cette méthode permet de trouver la solution optimale. Cependant, le test de tous les sous-ensembles nécessite beaucoup de calcul pour un grand nombre de paramètres. Cette méthode n'est pas utilisable avec un nombre de paramètres trop important.
- La génération heuristique : des paramètres sont supprimés ou ajoutés à chaque essai jusqu'à trouver la solution optimale. Cette méthode regroupe trois sous-catégories :
 1. forward : le sous-ensemble de départ est vide et des paramètres sont rajoutés à chaque itération.
 2. Backward : le sous-ensemble de départ contient tous les paramètres et des paramètres sont enlevés à chaque itération. .
 3. Stepwise : cette catégorie est le mélange des deux approches. Des paramètres sont ajoutés ou supprimés du sous-ensemble à chaque itération.

- La génération aléatoire : un nombre de sous-ensembles est généré aléatoirement. On détermine ensuite le meilleur sous-ensemble parmi ce groupe. Cette méthode nécessite moins de calcul que la génération exhaustive, car elle ne parcourt pas toutes les possibilités de sous-ensemble.

Il existe deux approches possibles pour l'évaluation des sous-ensembles :

- L'approche "filter" : un score va être attribué à chaque paramètre en fonction de son utilité pour la classification. L'évaluation se fait indépendamment de l'algorithme de classification et plusieurs types de score peuvent être utilisés.
- L'approche "wrapper" : l'algorithme d'apprentissage est utilisé pour sélectionner les paramètres. L'avantage de cette méthode par rapport à la méthode "filter" est de prendre en compte l'algorithme d'apprentissage pour sélectionner les paramètres. Cette méthode nécessite par contre plus de temps de calcul et doit être mise à jour si un autre algorithme d'apprentissage est utilisé pour la classification.

Dans GUPTA et DALLAS, 2014, une approche "filter" et une approche "wrapper" sont comparées pour la sélection des paramètres. Dans cet article, l'algorithme pour l'approche "filter" est le Relief-F. La méthode Relief est présentée par KIRA et RENDELL, 1992 et utilise une approche statistique avec une génération heuristique. Cette méthode a été améliorée en Relief-F par KONONENKO, 1994 pour l'utiliser dans des problèmes de classification multi-classes et réduire la sensibilité de l'algorithme au bruit. Une revue des différents algorithmes basés sur Relief est disponible dans URBANOWICZ et al., 2017. Le deuxième algorithme, celui utilisé dans l'approche "wrapper", est le "Sequential Forward Floating Search" présenté dans PUDIL, NOVICOVA et KITTLER, 1994. Au départ, un ensemble de 31 paramètres est utilisé pour la classification, l'approche "filter" a sélectionné un ensemble de 21 paramètres pour l'algorithme d'apprentissage k plus proches voisins et un ensemble de 30 paramètres pour l'algorithme d'apprentissage Naïves bayésiennes. L'approche "wrapper" permet de sélectionner 11 paramètres pour l'algorithme d'apprentissage k plus proches voisins et 12 paramètres pour l'algorithme d'apprentissage Naïves bayésiennes. Dans cette étude, l'approche "wrapper" permet de mieux réduire le nombre de paramètres pour la classification.

Dans ESPINILLA et al., 2018, deux autres méthodes de sélection de paramètres sont

étudiées pour déterminer les paramètres les plus utiles pour la reconnaissance d'activités. Une des méthodes utilise une génération aléatoire avec l'algorithme Las Vegas Filter présenté dans LIU et SETIONO, 1996. Le but est de choisir, de façon aléatoire, le sous-ensemble des caractéristiques ayant la plus petite mesure d'inconsistance. La mesure d'inconsistance est définie par le rapport du nombre de points inconsistants sur le nombre points total. Deux points sont dits inconsistants si ils possèdent les même valeurs de paramètres et des classes différentes.

D'autres types d'algorithmes sont utilisés pour la sélection de paramètres comme par exemple un algorithme génétique. Cet algorithme est inspiré de la théorie de Darwin et est décrit plus en détail dans ENGELBRECHT, 2007. Cet algorithme est utilisé dans ROSATI, BALESTRA et KNAFLITZ, 2018 pour faire une comparaison entre deux groupes de paramètres : le premier groupe regroupe les paramètres souvent utilisés dans la reconnaissance d'activités comme la moyenne, la variance et la transformée de Fourier. Le second groupe correspond a des paramètres calculés à partir des pics du signal et du "zero crossing mode". Un algorithme génétique est ensuite utilisé pour la sélection des paramètres avant d'effectuer une classification pour comparer les performances des deux groupes de paramètres.

Les méthodes précédentes consistent toutes à sélectionner les paramètres les plus pertinents parmi l'ensemble de paramètres de départ. Une autre approche possible consiste à réduire la dimension de l'ensemble des paramètres de départ en transformant cet ensemble en un sous-ensemble de dimensions plus petit via l'association de différents paramètres.

L'analyse des composantes principales est une méthode descriptive utilisée pour réduire la dimension de l'ensemble de paramètres. Cette méthode permet de trouver un espace de dimension plus petit en construisant de nouveaux paramètres à partir de combinaisons linéaires des paramètres de départ. Les nouveaux paramètres sont choisis pour ne pas être corrélés et maximiser la dispersion des points. Plus de détails sur l'analyse des composantes sont disponibles dans JOLLIFFE, 2002. Une méthode dérivée de l'analyse des composantes principales est par exemple utilisée dans YANG, WANG et CHEN, 2008 pour de la reconnaissance d'activités.

L'analyse discriminante linéaire est une autre méthode utilisée pour la réduction des dimensions et consiste à choisir des combinaisons linéaires des paramètres pour maximiser l'homogénéité de chaque classe. L'analyse discriminante linéaire est par exemple utilisée dans KHAN et al., 2010 pour la réduction de la dimension du vecteur de paramètres.

2.5.3 Normalisation des paramètres

Les différents paramètres choisis peuvent avoir des éventails de valeurs très différents. Une étape de normalisation est nécessaire avant de pouvoir utiliser les paramètres pour la classification. Il existe plusieurs types de normalisation. Dans ROSATI, BALESTRA et KNAFLITZ, 2018, la normalisation min-max est utilisée. Cette normalisation consiste à calculer tous les paramètres pour qu'il soit compris entre 0 et 1. La formule utilisée est :

$$p_{norm_i} = \frac{p_i - \min(p_i)}{\max(p_i) - \min(p_i)} \quad (2.41)$$

où p_i est le paramètre d'index i et p_{norm_i} est le paramètre normalisé entre 0 et 1.

D'autres types de normalisation peuvent être utilisés. L'expression suivante présente une normalisation basée sur la moyenne :

$$p_{norm_i} = \frac{p_i - \mu_p}{\max(p_i) - \min(p_i)} \quad (2.42)$$

où μ_p est la moyenne sur les paramètres p .

La normalisation suivante est parfois aussi utilisée et fait intervenir l'écart-type et la moyenne du paramètre :

$$p_{norm_i} = \frac{p_i - \mu_p}{\sigma_p} \quad (2.43)$$

Cette normalisation est par exemple utilisée dans CHEN et SHEN, 2017 et ZHANG et SAWCHUK, 2011b.

2.6 Classification

La classification consiste à associer chaque donnée à une classe en utilisant un algorithme de classification. Cet algorithme est configuré lors d'une phase d'apprentissage. Il

existe deux grands types d'apprentissage :

- L'apprentissage supervisé : un ensemble de données dont la classe d'appartenance est connue est disponible. Cet ensemble de données va permettre de déterminer la classe de nouvelles données.
- L'apprentissage non supervisé : la classe d'aucune donnée n'est connue au départ. L'objectif est la recherche d'une répartition visant à regrouper les individus dans des classes. Cette répartition est réalisée de manière à avoir les classes les plus homogènes possible et les classes les plus distinctes des unes des autres possibles.

L'apprentissage non supervisé est parfois utilisé pour la reconnaissance d'activités comme par exemple dans NGUYEN, MOORE et MCCOWAN, 2007. Cependant, l'apprentissage supervisé est le plus utilisé dans la reconnaissance d'activités et il est constitué de deux étapes :

- L'apprentissage : le but est d'apprendre un modèle de répartition à l'aide de données étiquetées.
- La décision : utilisation du modèle construit à l'étape précédente pour estimer la classe de nouvelles données.

Il n'existe pas de consensus sur l'algorithme à utiliser pour faire de la reconnaissance d'activités. Le tableau 2.7 énumère différents algorithmes de classification supervisée. Il existe des études comparatives entre les différents algorithmes de classification. Par exemple, dans COMPANJEN, 2009, une étude comparative est proposée entre la classification avec un modèle de Markov caché, le réseau de neurones, la logique floue et le séparateur à vaste marge. Dans SAEZ, BALDOMINOS et ISASI, 2017, douze algorithmes de classification sont comparés avec les mêmes paramètres d'entrées.

Les principaux algorithmes de classification supervisée sont présentés ci-dessous. Pour la suite, on admet donc qu'un ensemble d'exemples est à disposition pour l'apprentissage. On définit $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)\}$ l'ensemble des points d'apprentissage de longueur L avec $x_i \in \{1, 2, \dots, K\}$ le label du vecteur y_i qui contient d paramètres avec $y_i \in R^d$. K est le nombre de classes possibles. x_i correspond à la classe de l'activité correspondant aux paramètres y_i .

TABLE 2.7 – Classification supervisée utilisée pour la reconnaissance d’activité

Classification	publication(s)
K plus proches voisins	ATALLAH et al., 2011 CHEN et SHEN, 2017 COMPANJEN, 2009 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GU et al., 2015 GHAZALI et al., 2018 GUPTA et DALLAS, 2014 JAIN et KANHANGAD, 2018 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 PREECE et al., 2009a WANG et al., 2018
Bayésien Naïf	ATALLAH et al., 2011 CLELAND et al., 2013 DUTTA et al., 2018 GU et al., 2015 GUPTA et DALLAS, 2014 MITCHELL, MONAGHAN et O’CONNOR, 2013 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 WANG et al., 2018
Séparateur à vaste marge	BAYAT, POMPLUN et TRAN, 2014 CHEN et SHEN, 2017 CLELAND et al., 2013 COMPANJEN, 2009 DAVILA, CRETU et ZAREMBA, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GU et al., 2015 GHAZALI et al., 2018 HE et JIN, 2008 HE et JIN, 2009 JAIN et KANHANGAD, 2018 KIM, CHO et KIM, 2012 NI, ZHANG et LI, 2018 MITCHELL, MONAGHAN et O’CONNOR, 2013 PANNURAT et al., 2017 PRASERTSUNG et HORANONT, 2016 RAHIM et al., 2018 WANG et al., 2018 ZHANG et SAWCHUK, 2011b
Réseaux de neurones	BAYAT, POMPLUN et TRAN, 2014 CLELAND et al., 2013 COMPANJEN, 2009 KHAN et al., 2010 KHAN, LEE et KIM, 2008 LEE et al., 2011 MITCHELL, MONAGHAN et O’CONNOR, 2013 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 SAN-SEGUNDO et al., 2018 YANG, WANG et CHEN, 2008
Analyse discriminante linéaire	GU et al., 2015 GHAZALI et al., 2018
Les forêts aléatoires	BAYAT, POMPLUN et TRAN, 2014 CASALE, PUJOL et RADEVA, 2011 CHAPRON et al., 2018 CHEN et SHEN, 2017 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 MEHRANG, PIETILA et KORHONEN, 2018 RAHIM et al., 2018 SAN-SEGUNDO et al., 2018
Arbre de décision	BAYAT, POMPLUN et TRAN, 2014 CLELAND et al., 2013 ERMES, PARKKA et CLUITMANS, 2008 FULLERTON, HELLER et MUNOZ-ORGANERO, 2017 GU et al., 2015 GHAZALI et al., 2018 NI, ZHANG et LI, 2018 PANNURAT et al., 2017 WANG et al., 2018

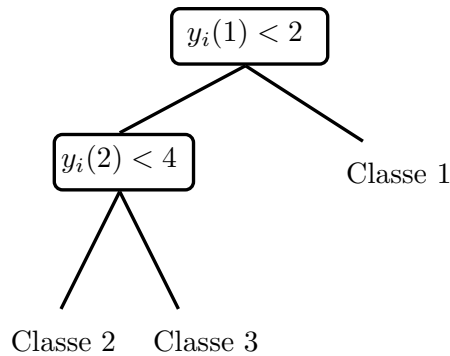


FIGURE 2.5 – Exemple d'un arbre de décision

2.6.1 Apprentissage par arbre de décision

Un arbre de décision décrit dans DUDA, HART et STORK, 2001, MITCHELL, 1997 et ROKACH et MAIMON, 2014 est un modèle hiérarchique constitué de nœuds et de feuilles. Chaque nœud correspond à une condition sur un paramètre et un arbre est constitué d'une série de nœuds qui permet de finir sur une feuille avec une classe attribuée. La figure 2.5 donne un exemple d'arbre de décision.

Le but de l'apprentissage est de définir les différents nœuds et la classe des feuilles. Un algorithme usuel pour cette phase d'apprentissage est le CART présenté dans BREIMAN et al., 1984. Cet algorithme est composé de trois étapes successives. La première étape est la construction d'un arbre binaire maximal. Pour cette étape, il est nécessaire de choisir un critère de construction, une règle d'arrêt et une règle d'affectation. Le critère de construction va définir les différents nœuds de l'arbre en utilisant une fonction d'hétérogénéité comme par exemple la concentration de Gini ou l'entropie. Différentes fonctions d'hétérogénéité sont comparées dans RAILEANU et STOFFEL, 2004. La règle d'arrêt va permettre d'arrêter la construction de l'arbre. La règle d'affectation attribue une classe aux feuilles de l'arbre. L'arbre maximal obtenu n'est pas directement utilisable, car il possède un nombre trop important de feuilles et est très dépendant des données d'apprentissage utilisées. La deuxième étape de l'algorithme a pour objectif d'élaguer l'arbre pour réduire le nombre de feuilles et le rendre moins fidèle aux données d'apprentissage. Cette étape va permettre de construire un ensemble de sous-arbres de l'arbre maximal et de choisir l'arbre le plus performant parmi cet ensemble. Le CART n'est pas le seul algorithme d'apprentissage par arbre de décision. Par exemple, l'algorithme C4.5 est utilisé

dans LARA et LABRADOR, 2013 . Cet algorithme a été présenté dans QUINLAN, 1993 et il est une amélioration de l'algorithme ID3 présenté dans QUINLAN, 1986.

L'avantage de l'apprentissage par arbre de décision est qu'il est facilement interprétable par les utilisateurs. De plus, l'utilisation de l'arbre de décision nécessite peu de calcul et peut donc être facilement embarqué dans un objet connecté qui possède peu de ressources de calcul. Néanmoins la classification par arbre de décision est sensible à l'ordre des données d'apprentissage et sujet au sur-apprentissage.

2.6.2 Les forêts aléatoires

Les forêts aléatoires est un algorithme de classification proposé dans BREIMAN, 2001 et basé sur l'apprentissage par arbre de décision. L'algorithme des forêts aléatoires est une amélioration du bagging. Cette méthode est décrite dans BREIMAN, 1996 et consiste à construire plusieurs arbres de décision à partir de bases de données d'apprentissage différentes. Les bases de données sont construites en tirant au sort des échantillons à partir de la base de données de départ. Comme l'apprentissage par arbres de décision est très sensible aux données d'apprentissage, tous les arbres de décisions sont différents. La classe prédite finale est la classe prédite par le plus d'arbres de décision. La particularité des forêts aléatoires est d'utiliser l'algorithme CART légèrement modifié pour la création des noeuds : le critère de construction pour l'élaboration des noeuds utilise une partie des paramètres disponibles, partie choisie de manière aléatoire.

Cet algorithme a été utilisé dans MEHRANG, PIETILA et KORHONEN, 2018 pour différencier cinq activités différentes (le cyclisme haute intensité et basse intensité, les tâches ménagères, le repos et al position assise). Le meilleur résultat est obtenu avec une précision de 89.6% à partir d'une forêt de 64 arbres et d'un fenêtrage de 13 secondes avec 90% de recouvrement.

2.6.3 Naïve Bayésienne

La classification Naïve Bayésienne décrite dans DUDA, HART et STORK, 2001 et MITCHELL, 1997 suit une approche probabiliste et se base sur la formule de Bayes :

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \quad (2.44)$$

où $P(x|y)$ est la probabilité de la classe x sachant les paramètres y , $P(x)$ est la probabilité de la classe x , $P(y|x)$ est la probabilité des paramètres y sachant la classe x et $P(y)$ est la probabilité des paramètres y . La classification Naïve Bayésienne a pour hypothèse forte que tous les paramètres sont indépendants les uns des autres. On peut écrire alors :

$$P(y|x) = \prod_{i=1}^d P(y(i)|x) \quad (2.45)$$

À partir de (2.44), on peut écrire :

$$P(x|y) = \frac{P(x) \prod_{i=1}^d P(y(i)|x)}{\sum_{j=1}^K P(x) \prod_{j=1}^d P(y(i)|x_j)} \quad (2.46)$$

L'avantage de la classification Naïve Bayésienne est qu'elle peut être aisément mise en oeuvre, son inconvénient est qu'elle est basée sur l'hypothèse de l'indépendance des paramètres, hypothèse rarement valide.

2.6.4 Analyse discriminante

L'analyse discriminante est une technique statistique qui permet la détermination d'une frontière entre les différentes classes. C'est une version plus générale de la classification naïve bayésienne. On considère que les variables d'entrée de la classification suivent toutes une distribution normale. Les frontières sont déterminées à partir des barycentres des différentes classes et la matrice de covariance. Le fonctionnement de l'analyse discriminante est décrit plus en détail dans DUDA, HART et STORK, 2001 et THARWAT et al., 2017. La prédiction d'une classe est obtenue, comme précédemment, en utilisant la formule de Bayes :

$$P(x = k|y) = \frac{P(y|x = k)P(x = k)}{P(y)} \quad (2.47)$$

Il est supposé ici, via la loi normale, $P(y|x = k)$ peut s'exprimer comme suit :

$$P(y|x = k) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(y - \mu_k)^T \Sigma_k^{-1} (y - \mu_k)\right) \quad (2.48)$$

où μ_k est la moyenne des paramètres de la classe k et Σ_k est la matrice de covariance de la classe k . La classe prédite est alors obtenue à partir de :

$$\operatorname{argmax}_{k=1,\dots,K} \left\{ \pi_k P(y|x = k) \right\} \quad (2.49)$$

ou encore

$$\operatorname{argmax}_{y=1,\dots,K} \left\{ \log(\pi_k) - \frac{1}{2} \log(|\Sigma_k|^{1/2}) - \frac{1}{2} (y - \mu_k)^T \Sigma_k^{-1} (y - \mu_k) \right\} \quad (2.50)$$

où $\pi_k = N_k/L$ avec N_k le nombre de points d'apprentissage de la classe k parmi les L disponibles.

Dans le cas de l'analyse discriminante linéaire, l'apprentissage du modèle consiste en l'estimation de la moyenne μ_k de chaque classe, de π_k et de la matrice de covariance Σ_k . Dans l'analyse discriminante linéaire les matrices de covariance sont supposées identiques pour toutes les classes. Les paramètres sont calculés à partir de :

$$\mu_k = \frac{1}{N_k} \sum_{i|x_i=k} y_i \quad (2.51)$$

$$\Sigma = \frac{1}{L} \sum_{k=1}^K \sum_{i|x_i=k} (y_i - \mu_k)(y_i - \mu_k)^T \quad (2.52)$$

Dans le cas de l'analyse discriminante quadratique, chaque classe a une matrice de covariance différente qui est déterminée par :

$$\Sigma_k = \frac{1}{N_k} \sum_{i|x_i=k} (y_i - \mu_k)(y_i - \mu_k)^T \quad (2.53)$$

2.6.5 K plus proche voisins

La méthode des k plus proches voisins décrit dans DUDA, HART et STORK, 2001 et MITCHELL, 1997 est un algorithme de classification supervisée simple qui se base sur la mesure de la distance entre les différents points. La particularité de cette méthode est qu'elle ne nécessite pas d'une phase d'apprentissage pour apprendre un modèle, le modèle est constitué des données d'apprentissage. Pour utiliser cette méthode, il suffit de définir un entier k , une base d'apprentissage et une fonction de distance. Pour pouvoir

déterminer la classe d'un nouveau point, la distance entre ce point et les points de la base d'apprentissage est calculée. La classe la plus fréquente parmi les k points les plus proches est attribuée au nouveau point.

La distance entre les points peut être calculée de différentes façons, quelque exemples usuels sont la distance euclidienne, la distance Manhattan et la distance Minkowski. La distance euclidienne reste la distance la plus utilisée.

L'inconvénient principal de cette méthode est que la classification d'un nouveau point requiert beaucoup de temps de calcul. Un autre inconvénient est que tous les points de la base d'apprentissage doivent être stockés pour pouvoir être utilisés pour la classification, enfin le résultat de la classification est très fortement lié à la base d'apprentissage et au choix du nombre de voisins k .

2.6.6 Séparateur à Vaste Marge (SVM)

Le SVM est un algorithme de classification supervisé souvent utilisé pour la classification d'activités. Cet algorithme est utilisé par exemple dans HE et JIN, 2008 en utilisant les paramètres d'un modèle AR comme entrée du classifieur. Dans HE et JIN, 2009, la transformée en cosinus discrète est utilisée comme entrée pour la classification par SVM. Le but de cet algorithme est de déterminer un hyperplan séparateur à marges maximales entre deux classes. Cette algorithme est décrit dans BURGES, 1998, DUDA, HART et STORK, 2001 et STEINWART et CHRISTMANN, 2008. Dans le cas linéaire et dans le cas d'une classification en 2 classes ($x_i = 1$ ou $x_i = -1$), l'hyperplan séparateur optimal est déterminé par la minimisation la fonction suivante :

$$\min \left\{ \frac{1}{2} |w|^2 + C \sum_{i=1}^l \epsilon \right\}, \epsilon \geq 0, C \geq 0 \quad (2.54)$$

$$\text{tel que } y_i(w^T x_i + b) \geq 1 - \epsilon_i, i = 1, 2, \dots, l \quad (2.55)$$

où C est une variable de pénalisation des points mal classés, w est un vecteur paramétrant l'hyperplan, b un scalaire et les ϵ_i sont des variables positives de relâchement des

contraintes. Le problème (2.54) peut être résolu en utilisant les multiplicateurs de Lagrange. La fonction discriminante peut alors être définie par :

$$f(x) = \text{sgn}\left(\sum_{i=1}^l \alpha_i y_i x_i^T x + b\right) \quad (2.56)$$

où les α_i sont obtenus par :

$$\max\left\{\sum_{i=1}^l \alpha_i - \frac{1}{2}\alpha_i \alpha_j y_i y_j x_i x_j\right\} \quad (2.57)$$

$$C \geq \alpha_i \geq 0, i = 1, \dots, l, \sum_{i=1}^l \alpha_i y_i = 0 \quad (2.58)$$

Le cas linéaire peut être étendu au cas non-linéaire en utilisant une fonction noyau. L'idée ici est de transformer l'espace de représentation des points en un espace de plus grande dimension. On remplace donc les points x_i par $\theta(x_i)$ dans un autre espace. La fonction discriminante est alors définie par :

$$f(x) = \text{sgn}\left(\sum_{i=1}^l \alpha_i y_i K(x_i, x_j)^T x + b\right) \quad (2.59)$$

où $K(x_i, x_j) = \theta(x_i)^T \theta(x_j)$. (2.57) peut être réécrite par :

$$\max\left\{\sum_{i=1}^l \alpha_i - \frac{1}{2}\alpha_i \alpha_j y_i y_j K(x_i, x_j)\right\} \quad (2.60)$$

$$C \geq \alpha_i \geq 0, i = 1, \dots, l, \sum_{i=1}^l \alpha_i y_i = 0 \quad (2.61)$$

Plusieurs noyaux sont présentés dans BURGES, 1998, quelques exemples de noyaux souvent utilisés sont :

- Noyau linéaire : $K(x_i, x_j) = x_i^T x_j$
- Noyau polynomial : $K(x_i, x_j) = (x_i^T x_j + 1)^d$
- Noyau gaussien : $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

2.6.7 Réseaux de neurones

Le réseau de neurones décrit dans DUDA, HART et STORK, 2001 and MITCHELL, 1997 est un algorithme de classification inspiré par le fonctionnement des neurones biologiques. Un réseau de neurones est constitué d'un assemblage plus ou moins complexe de neurones formels. Le neurone formel est défini par une fonction d'activation et des entrées e_1, \dots, e_p :

$$s = g(w_0 + \sum_{i=1}^p w_i e_i) \quad (2.62)$$

où w_0 est une constante appelée biais du neurone et les w_1, \dots, w_p représentent le poids associé à chaque entrée e_i . Ces paramètres sont calculés durant la phase d'apprentissage, ceci pour chaque neurone du réseau. La fonction $g(\cdot)$ représente la fonction d'activation. Les principales fonctions d'activation sont :

- La fonction linéaire : $g(x) = x$
- La fonction Heaviside : $g(x) = 0$ si $x < 0$ sinon $g(x) = 1$
- La fonction sigmoïde : $g(x) = 1/(1 + e^x)$
- La fonction ReLU : $g(x) = 0$ si $x < 0$ sinon $g(x) = x$
- La fonction radiale : $g(x) = \sqrt{1/2\pi e^{-(x^2/2)}}$
- La fonction tangente hyperbolique : $g(x) = \frac{2}{1+e^{-2x}} - 1$

Comme illustré sur la figure 2.6, les neurones sont regroupés en couches et les couches sont reliées entre elles. La sortie d'une couche correspond à l'entrée d'une autre. La dernière couche détermine la classe. La couche finale est en général constituée de la fonction logistique ou softmax. Il existe de nombreux types de réseaux de neurones. Un type de réseau de neurones est défini par le choix du nombre de couches, du type de neurones de chaque couche, la nature de l'interaction entre les couches et de l'algorithme d'apprentissage utilisé. Une des structures les plus répandues est le perceptron multicouche, représenté sur la figure 2.6. Un perceptron multicouche est un réseau constitué de plusieurs couches successives. La première couche a pour fonction de lire les entrées et la dernière couche fournit la sortie. Une ou plusieurs couches cachées sont situées entre la couche d'entrée et la couche de sortie.

Dans LI et al., 2018, plusieurs types de réseaux de neurones sont comparés pour faire de la reconnaissance d'activités. Les différents types de réseaux de neurones présentés dans cette étude sont :

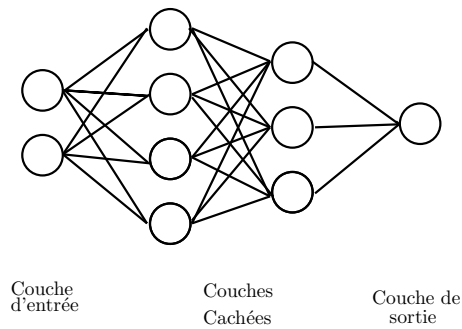


FIGURE 2.6 – Exemple d'un réseau de neurones avec deux couches cachées

- Un perceptron multicouche avec trois couches cachées.
- Un réseau de neurones convolutif. Ce réseau de neurones a été conçu pour le traitement d'images. Ce réseau est caractérisé par la présence de couches de convolution, de pooling, de corrections et de couches entièrement connectées. Cet algorithme a pour entrée directement le signal et la couche de convolution a pour rôle d'extraire les paramètres de ce signal. La couche de pooling sert à réduire la taille du signal en gardant les caractéristiques importantes, cette couche est souvent située entre deux couches de convolution. Plusieurs types de réseaux de neurones convolutifs sont comparés dans RUEDA et al., 2018 pour faire de la reconnaissance d'activités.
- Un réseau de neurones comportant des couches de neurones de mémoire à long terme et à court terme. Ces couches jouent le rôle de mémoire interne et permettent de revenir sur plusieurs étapes en arrière afin de prédire les prochains échantillons. Ce type de réseaux de neurones est utilisé par exemple dans EYOBU et HAN, 2018 pour faire de la reconnaissance d'activités.
- Un réseau de neurones hybride qui est composé de couches de réseaux de neurones convolutifs et de couches de mémoires à long terme et à court terme. Ce réseau de neurones est utilisé aussi dans SAN-SEGUNDO et al., 2018 pour faire de la reconnaissance d'activités à partir d'un smartphone et d'une montre connectée.

2.6.8 Combinaison d'algorithmes de classification

Dans la plupart des cas, un seul algorithme de classification est utilisé pour faire de la reconnaissance d'activités. Cependant, plusieurs algorithmes de classification peuvent être utilisés en combinaison pour améliorer les résultats. Par exemple, dans NI, ZHANG et LI, 2018, trois algorithmes de classification sont utilisés : SVM, arbre de décision et k plus proche voisin. Chaque algorithme propose une classe d'appartenance pour le point en cours de prédiction, le résultat final est la classe majoritaire. Cette méthode est appelée la fusion des décisions. L'avantage de cette méthode est d'améliorer le résultat de la classification comme le montre NI, ZHANG et LI, 2018. Cependant, cette technique nécessite plus de temps de calcul. Le vote majoritaire n'est pas le seul type de fusion de décision. Un autre type de fusion de décision est une moyenne des probabilités des algorithmes de classification, ceci est utilisé dans BAYAT, POMPLUN et TRAN, 2014.

D'autres approches décrites dans SAGI et ROKACH, 2018 sont utilisées pour construire un ensemble de classifieurs avec le même algorithme de classification. Le "bagging" présenté dans BREIMAN, 1996 consiste à créer un ensemble de classifieurs en utilisant des données d'apprentissages différentes. La base de chaque classifieur est construite en utilisant des échantillons aléatoires de la base de données d'apprentissage de départ. Le "bagging" est efficace avec des algorithmes qui sont sensibles aux changements des données de la base d'apprentissage comme par exemple les réseaux de neurones et les arbres de décisions. On retrouve ici le principe des forêts aléatoires. Une autre méthode utilisée est le "boosting". Cette méthode consiste à combiner les résultats de classifieurs faibles en combinaison linéaire pour obtenir un classifieur final plus performant. Un exemple d'algorithme de "boosting" souvent utilisé est "Adaboost". Cet algorithme est par exemple utilisé dans DAGHISTANI et ALSHAMMARI, 2016 et RAHIM et al., 2018. La méthode "random subspace" présenté dans HO, 1998 est similaire au "bagging". Dans le cas du "random subspace", un sous-ensemble de paramètres est généré aléatoirement pour créer un ensemble de classifieurs. Le "bagging", le "boosting" et la méthode "random subspace" sont décrits et comparés dans SKURICHINA et DUIN, 2002.

2.7 Validation

2.7.1 Données de validation

Pour éviter le problème de sur-apprentissage, il est important que les données utilisées pour la validation ne soient pas les mêmes que les données utilisées pour l'apprentissage. La méthode la plus simple est de scinder les données disponibles en deux parties et de garder la première partie pour l'apprentissage et la seconde partie pour la validation. Dans la plupart des cas, la validation croisée est utilisée pour pouvoir utiliser l'intégralité des données pour l'apprentissage et pour la validation. Cette technique consiste à découper l'ensemble des données en k parties de même longueur. À tour de rôles, une des parties va être utilisée pour la validation et le reste pour l'apprentissage, ceci jusqu'à ce que toutes les parties soient utilisées pour la validation.

Chaque individu a une morphologie différente et ceci influe directement sur les données fournies par l'accéléromètre. Pour cette raison, il existe plusieurs solutions possibles de validation décrites dans CHEN et SHEN, 2017 dans le cas où les données sont collectées sur plusieurs individus. La première approche consiste à construire un modèle pour chaque individu avec uniquement les données collectées sur cet individu. Cette approche est difficilement réalisable dans la vie réelle, car il n'est pas possible de collecter un nombre de données suffisant de tous les utilisateurs et le modèle peut changer au cours de temps en fonction de l'état de santé de l'utilisateur. La deuxième méthode consiste à utiliser un modèle généralisé construit à partir de plusieurs utilisateurs. Cette approche est la plus couramment utilisée. Une dernière approche consiste à utiliser l'ensemble des données d'un individu pour la validation et les données issues des autres utilisateurs sont utilisées pour l'apprentissage. Ce processus est répété pour chaque individu. Cette validation a pour avantage de prendre en compte les possibles erreurs dues à la morphologie des personnes ou les différences dans les mouvements entre personnes. Cette technique est par exemple utilisée dans SAEZ, BALDOMINOS et ISASI, 2017.

2.7.2 Indicateurs de performance

Après avoir élaboré un algorithme pour faire de la reconnaissance d'activités, il est important de le valider par un ou plusieurs indicateurs clairs. À partir des données de validation, il est possible de calculer, pour un algorithme donné, les variables suivantes :

- VP qui représente le nombre de vrais positifs
- FP qui représente le nombre de faux positifs
- VN qui représente le nombre de vrais négatifs
- FN qui représente le nombre de faux négatifs

À partir de ces proportions, il est possible de définir la justesse qui est la métrique la plus utilisée dans la littérature pour mesurer la performance de l'algorithme. Il correspond au nombre d'activités correctement classées sur le nombre total d'activités :

$$justesse = \frac{VP + FP}{VP + FP + VN + FN} \quad (2.63)$$

En complément de la justesse, la précision et le rappel sont deux autres métriques souvent utilisées. Le rappel correspond à la probabilité que l'activité soit bien prédite si l'algorithme d'apprentissage prédit cette activité. La précision correspond au nombre de fois où une activité est correctement classée sur le nombre total de fois où elle est prédite.

$$precision = \frac{VP}{VP + FP} \quad (2.64)$$

$$rappel = \frac{VP}{VP + FN} \quad (2.65)$$

À partir de la précision et du rappel, on peut déterminer la F-mesure :

$$F_{\beta} = \frac{(1 + \beta^2) precision * rappel}{\beta^2 (precision + rappel)} \quad (2.66)$$

La F-mesure est souvent utilisée avec $\beta = 1$ et $\beta = 2$ comme dans SAEZ, BALDOMINOS et ISASI, 2017.

La F-mesure pondérée est une mesure qui prend en compte le nombre d'éléments dans chaque classe. Cette mesure est adaptée aux cas où le nombre d'échantillons dans chaque classe n'est pas la même.

$$F_{\beta pond} = \frac{N_k (1 + \beta^2) precision * rappel}{L \beta^2 (precision + rappel)} \quad (2.67)$$

Les résultats de la classification sont aussi, dans la plupart des cas, exposés sous la forme d'une matrice de confusion en comparant les données classées avec des données de référence : les colonnes correspondent au nombre de points d'une classe estimée et les lignes correspondent au nombre de points d'une classe réelle. La courbe de ROC FAWCETT, 2004 est utilisée aussi pour mesurer la performance du classifieur. Elle est par exemple utilisée dans RAHIM et al., 2018 ou VELTINK et al., 1996. La courbe de ROC représente le taux de vrais positifs en fonction du taux de faux positifs. L'aire de la courbe donne une indication sur la performance du classifieur : plus l'aire de la courbe est élevée, plus le classifieur est performant.

2.8 Conclusion

Dans ce chapitre, il est présenté un état de l'art sur la reconnaissance d'activités à partir des données d'un accéléromètre. Le placement de l'accéléromètre sur l'utilisateur joue un rôle important sur les résultats et dépend du système de mesures utilisé. Les smartphones sont maintenant présents dans notre quotidien et nous permettent de récolter un grand nombre de données. Cependant, des plate-formes dédiées peuvent être développées spécialement pour l'acquisition de données. Ces plates-formes ont pour avantage d'avoir un emplacement fixe, ce qui peut faciliter la reconnaissance d'activités.

Ensuite, le pré-traitement des données de l'accéléromètre est une étape nécessaire pour extraire des informations particulières sur le signal. À partir de ce signal, un grand nombre de paramètres peut être calculé pour être utilisé comme entrée pour des algorithmes de classification supervisés. La sélection de ces paramètres est une étape cruciale pour éviter les informations redondantes et plusieurs techniques de sélection sont présentées. Pour ce qui est de l'algorithme de classification, il n'existe pas de consensus sur l'algorithme de classification à utiliser, les algorithmes les plus utilisés sont ainsi présentés. Pour finir, différentes métriques pour mesurer la performance du classifieur sont présentées.

Il convient de noter que seules les données fournies par un accéléromètre ont été prises en compte, d'autres capteurs peuvent bien entendu être utilisés pour la reconnaissance d'activités comme un gyroscope ou un magnétomètre. Les techniques présentées dans ce document pour le traitement des données de l'accéléromètre peuvent être aussi appliquées aux données brutes de ces autres capteurs.

Chapitre 3

L'environnement d'eTact

3.1 Introduction

L'objectif de la thèse est de faire de la reconnaissance d'activités entre cinq activités (repos, marche, marche dans un escalier, saut et course) à partir d'un dispositif développé par Bodycap : eTact. Ce dispositif est présenté dans ce chapitre. La configuration hardware d'eTact a un impact sur les données et les algorithmes utilisables pour faire de la reconnaissance d'activités et les différentes contraintes liées à ce produit vont être exposées dans ce chapitre. Une des fonctionnalités d'eTact est la possibilité d'envoyer les données des capteurs en temps réel ou de les enregistrer en mémoire pour pouvoir ensuite être traitées sur un appareil externe. À partir de cette fonctionnalité, il est possible de suivre plusieurs approches pour faire de la reconnaissance d'activités. Ces différentes approches varient en fonction du support utilisé pour le calcul des paramètres et des algorithmes de classification. Chaque approche est présentée dans ce chapitre afin d'exposer ses avantages et inconvénients. Pour finir, il est nécessaire d'utiliser une base de données d'apprentissage pour faire de la reconnaissance d'activités à partir d'algorithmes d'apprentissage supervisés. La base de données utilisée dans cette thèse est présentée dans ce chapitre.

3.2 Présentation d'eTact

eTact (présenté dans la figure 3.1) est un objet connecté qui permet de mesurer l'actimétrie et d'avoir une indication de la température cutanée. eTact est constitué comme suit :



FIGURE 3.1 – eTact

- Un accéléromètre qui permet de mesurer l'accélération d'une personne suivant trois axes. Dans l'usage actuel, les données de l'accéléromètre sont utilisées pour mesurer l'actimétrie d'une personne en utilisant des algorithmes appropriés.
- Un capteur de température pour la mesure de la température cutanée d'une personne.
- Un microcontrôleur (MCU) qui permet de configurer et de gérer les différents composants d'eTact.
- Un module Bluetooth Low Energy (BLE) qui permet de communiquer en utilisant le protocole BLE avec un appareil externe.
- Une mémoire externe flash. La mémoire interne du MCU est de taille limitée et la mémoire vive est effacée en cas de coupure d'énergie. Pour cette raison, une mémoire externe flash est utilisée pour stocker un nombre de données plus important et restituer les données même en cas de coupure d'énergie.
- Une batterie rechargeable. L'avantage d'utiliser une batterie rechargeable est de pouvoir recharger eTact à l'aide d'un câble Universal Serial Bus (USB). Cependant, la capacité d'une batterie rechargeable est moins élevée que celle d'une pile à poids équivalent.

L'architecture eTact est résumée dans la figure 3.2. Dans notre cas, les données du capteur de température ne sont pas utiles et ne sont pas prises en compte dans la suite de la thèse. Ce capteur est donc désactivé pour diminuer la consommation d'énergie. eTact

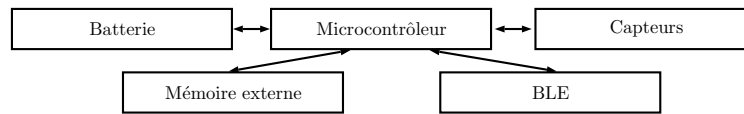


FIGURE 3.2 – Architecture d'eTact

a plusieurs configurations possibles :

- Un mode "données brutes" : les données brutes de l'accéléromètre sont directement sauvegardées dans la mémoire. L'utilisateur accède à ces données en connectant eTact à un ordinateur et en déchargeant les données.
- Le fonctionnement "mémoire" : les données des capteurs sont collectées et traitées par le MCU. Les résultats des algorithmes d'actimétrie sont ensuite stockés dans la mémoire externe. L'utilisateur récupère ensuite les données stockées en connectant eTact sur un ordinateur en utilisant un logiciel.
- Le fonctionnement "BLE" : les données des capteurs sont collectées et traitées par le MCU. Les résultats des algorithmes d'actimétrie sont stockés de manière temporaire dans la mémoire externe. Les résultats des algorithmes sont ensuite envoyés par BLE vers un appareil externe quand l'utilisateur le demande.

Le fonctionnement "mémoire" a pour avantage de ne pas utiliser le module BLE qui consomme de l'énergie sur eTact. Cependant, ce fonctionnement limite la durée de fonctionnement au temps nécessaire pour remplir la mémoire externe. Le fonctionnement BLE permet de transférer les données enregistrées dans la mémoire sans devoir connecter eTact à un ordinateur. Dans la suite, les deux fonctionnements sont envisagés dans l'étude de faisabilité de reconnaissance d'activités avec eTact.

3.3 Les contraintes

3.3.1 Contrainte matériel

L'utilisation d'eTact introduit des contraintes dans l'élaboration de l'algorithme de reconnaissance d'activités à cause des composants hardware. Les premières contraintes sont liées à l'accéléromètre. La fréquence d'échantillonnage peut être configurée à 25 Hz ou 50 Hz et la plage de mesure peut être configurée à $\pm 2g$, $\pm 4g$ ou $\pm 8g$. Chaque donnée est codée sur 12 bits.

Une contrainte importante d'eTact est liée à la capacité de la mémoire externe. La mémoire flash externe a une capacité de 64 Mo et une partie de la mémoire est réservée pour le fonctionnement d'eTact. En fonctionnement "données brutes", eTact peut mémoriser jusqu'à 11h38 de données brutes de l'accéléromètre avec des données brutes de l'accéléromètre codée sur 12 bits et une fréquence d'échantillonnage à 50 Hz. Le temps de remplissage de mémoire externe est de 23h16 avec une fréquence d'échantillonnage à 25 Hz.

3.3.2 Positionnement

Le positionnement envisagé pour l'utilisation d'eTact est le torse, le dos, le poignet, la cheville et la taille. Le choix du placement du capteur sera réalisé par la suite parmi ces emplacements en fonction des besoins de l'utilisateur et performances de l'algorithme de classification. Il est possible d'utiliser plusieurs capteurs en même temps positionnés à différents endroits pour la reconnaissance d'activités. Dans le cas de l'utilisation d'eTact, il a été choisi d'utiliser un seul dispositif pour chaque individu.

3.4 Les différentes approches proposées

Plusieurs approches sont possibles pour mettre en œuvre la reconnaissance d'activités sur eTact. Les différentes approches vont varier en fonction du support matériel utilisé pour le calcul des paramètres et la classification de l'activité. Ainsi, les données à enregistrer puis à transmettre par eTact vont varier selon l'approche utilisée. Voici les différentes approches possibles :

- *Approche 1* : Les données brutes sont envoyées par BLE ou sauvegardées dans la mémoire. Le calcul des paramètres et la classification de l'activité est fait sur un appareil externe. L'avantage de cette approche est la possibilité d'utiliser des paramètres et un algorithme de classification coûteux en nombre d'opérations. Par contre, l'envoi de toutes les données brutes par BLE consomme énormément d'énergie sur eTact et a un impact sur l'autonomie du système. En fonctionnement mode "mémoire", toutes les données brutes de l'accéléromètre doivent être

stockées dans la mémoire. 300 octets doivent être envoyés chaque seconde si l'accéléromètre est configuré à 50 Hz ou 150 octets si l'accéléromètre est configuré à 25 Hz.

- *Approche 2* : Le calcul des paramètres est effectué sur eTact. Le but de cette approche est d'envoyer moins de données par BLE ou de stocker moins de données dans la mémoire. Cependant, les paramètres calculés sur eTact doivent solliciter eTact le moins possible pour limiter la consommation de batterie. De plus, cette approche a un intérêt seulement si la taille de l'ensemble des paramètres est inférieure à la taille de l'ensemble des données brutes nécessaires pour calculer ces paramètres sur une même période. La classification de l'activité est effectuée sur un appareil externe et peut donc utiliser un nombre d'opérations important.
- *Approche 3* : Le calcul des paramètres et la classification sont effectués sur eTact. Seuls les résultats de la classification sont envoyés par BLE ou stockés en mode "mémoire". Comme pour l'approche 2, pour assurer une autonomie suffisante, le calcul des paramètres et l'algorithme de classification doivent solliciter eTact le moins possible. L'avantage de cette méthode est d'envoyer un faible nombre de données vers un appareil externe et donc de limiter l'utilisation de la mémoire ou du BLE.
- *Approche 4* : Cette approche est similaire à l'approche 1. Les calculs des paramètres et la classification de l'activité sont réalisés sur un appareil externe pour éviter une consommation excessive de l'énergie sur eTact. La particularité de cette approche est de binariser les données brutes avant de les envoyer par BLE. La binarisation des données brutes a pour but de réduire le nombre de données à envoyer par BLE ou à stocker dans la mémoire externe. Une donnée brute de l'accéléromètre est codée sur deux octets et il est possible de coder seize données binarisées sur ces deux octets. Au final, seize fois moins de données sont envoyées par BLE que l'approche 1 ce qui permet de réduire la consommation liée aux communications BLE. En contre partie, on perd en résolution de la mesure de l'accélération et une grande partie des informations fournies par les signaux de l'accéléromètre. De plus, certains paramètres ne peuvent pas être calculés avec des données binarisées.

Pour chacune des approches, l'autonomie de la mémoire et la batterie sont étudiées.

TABLE 3.1 – Récapitulatif des différentes approches

Approches	Types de données utilisées	Réalisation des calculs des paramètres	Réalisation des calculs de la classification
1	brutes	Appareil externe	Appareil externe
2	brutes	eTact	Appareil externe
3	brutes	eTact	eTact
4	binaire	Appareil externe	Appareil externe

Le but de cette thèse est de comparer les différentes approches en se basant sur la durée de vie d'eTact et sur la qualité de la reconnaissance d'activités. La durée de vie peut être limitée par la consommation de batterie ou par le temps nécessaire pour remplir entièrement la mémoire. Les différentes approches sont résumées dans le tableau 3.1. Les approches 1, 2 et 3 sont étudiées dans le chapitre 4 et l'approche 4 est étudiée dans le chapitre 6.

3.5 Les données d'apprentissages

La collecte des données est une étape fastidieuse mais nécessaire pour construire une base d'apprentissages solide pour la reconnaissance d'activités. Il a été choisi d'utiliser une base de données publiques. Il existe dans la littérature plusieurs bases de données publiques pour la reconnaissance d'activités. Une liste de ces différentes bases de données est disponible dans TWOMEY et al., 2018.

La base de données est choisie en fonction de plusieurs critères :

- Les données de la base de données sont similaires aux données fournies par l'accéléromètre d'eTact pour pouvoir utiliser les différents algorithmes développés sur eTact. La fréquence d'échantillonnage d'eTact peut être configurée à 25 Hz ou 50 Hz. Les bases de données choisies doivent donc avoir un multiple de 50 comme fréquence d'échantillonnage.
- La base de données doit contenir des données avec un capteur fixé à différentes parties du corps pour déterminer l'impact du positionnement d'eTact sur la reconnaissance d'activités.
- Le nombre de données de chaque activité doit être important et les données doivent être collectées sur plusieurs individus.

Deux bases de données ont été choisies pour la suite. La première base de données utilisée est PAMAP2 collectée par l'Université de Californie (la base de données est décrite dans REISS et STRICKER, 2012a et REISS et STRICKER, 2012b). Cette base de données est composée de 12 activités différentes : allongé, assis, debout, marche, course, vélo, marche nordique, monter un escalier, descendre d'un escalier, passer l'aspirateur, repasser, et saut à la corde. Le matériel utilisé pour l'acquisition est trois IMU (Inertial Measurement Units) et un tensiomètre cardiaque. Les IMUs sont positionnés à trois endroits différents (sur la poitrine, le poignet du bras dominant et la cheville du même coté que le bras dominant) et chaque IMU est composé de deux accéléromètres, un gyroscope et d'un magnétomètre. Seules les données des accéléromètres sont utilisées pour la suite. La fréquence d'échantillonnage des accéléromètres est de 100 Hz et la résolution est de 13 bits. Un accéléromètre a une plage de mesure de $\pm 6g$ et le deuxième accéléromètre a une plage de mesure de $\pm 16g$. Les données sont collectées sur 9 personnes différentes (8 hommes et 1 femme) pour chaque activité.

La deuxième base de données utilisée est USC-HAD collectée par l'Université de Californie (la base de données est décrite dans ZHANG et SAWCHUK, 2011a). Les 12 activités étudiées dans cette base de données sont marche en avant, marche en arrière, marche vers la droite, marche vers la gauche, montée d'un escalier, descente d'un escalier, course, assis, debout, allongé, montée dans un ascenseur et descente dans un ascenseur. Le capteur utilisé est un IMU fixé au niveau de la taille. Cet IMU comporte un accéléromètre, gyroscope et un magnétomètre. Comme pour la base de données précédente, seules les données de l'accéléromètre sont utilisées pour la suite. La fréquence d'échantillonnage est de 100 Hz avec une étendue de mesure de $\pm 6g$. Les données sont collectées sur 14 personnes différentes (7 hommes et 7 femmes) pour chaque activité.

Certaines activités dans la base de données sont similaires et sont regroupées dans une même classe. Les différentes positions d'une personne quand celle-ci n'est pas en mouvement comme la position allongée, assise, debout ou en attente dans un ascenseur sont regroupées dans la même activité repos. Les différentes marches définies dans la base de données USC-HAD (marche vers l'avant, e, arrière, vers la droite ou vers la gauche) sont considérées comme une même activité et sont regroupées dans l'activité marche. Les différentes activités à classer dans la suite de la thèse sont définies dans le tableau 3.2 pour la base de données PAMAP2 et dans le tableau 3.3 pour la base de données

TABLE 3.2 – Description des différents types d'activité de la base de données PAMAP2

label	activité	activités regroupées
A_1	repos	allongé, assis, debout
A_2	marche	marche
A_3	marche dans un escalier	monter et descente dans un escalier
A_4	corde a sauter	corde a sauter
A_5	course	course

TABLE 3.3 – Description des différents types d'activité de la base de données USC-HAD

label	activité	activités regroupées
B_1	repos	assis, debout, allongé, monté et descente dans un ascenseur
B_2	marche	marche en avant, marche en arrière, marche vers la droite, marche vers la gauche,
B_3	marche dans un escalier	montée et descente dans un escalier
B_4	saut	saut
B_5	course	course

USC-HAD. La fréquence d'échantillonnage des deux bases de données est réduite à 50 Hz ou à 25 Hz pour être similaire aux données de l'accéléromètre d'eTact. Le tableau 3.4 énumère le temps d'acquisition des différentes activités pour les deux bases de données.

3.6 Conclusion

Dans cette section, le dispositif eTact est présenté pour illustrer les différentes contraintes à l'élaboration d'un algorithme de reconnaissance d'activités à partir des données récoltées sur cet objet connecté. Ces contraintes peuvent être liées pour différentes raisons : la plateforme utilisée définit des contraintes liées à l'autonomie du système et l'utilisation souhaitée d'eTact définit le positionnement qui a un impact sur les données collectées.

eTact est un objet connecté qui peut envoyer des données via BLE ou les stocker en mémoire. Ces deux modes de fonctionnement donnent la possibilité de déporter les calculs des algorithmes sur une plateforme externe qui possède des ressources de calcul beaucoup plus importantes qu'eTact. Quatre différentes approches sont proposées pour

TABLE 3.4 – Temps d’acquisition des différentes activités pour chaque emplacement du capteur de la base de données

Activité	temps total en seconde
A_1	10774
A_2	4581
A_3	859
A_4	4057
A_5	1817
B_1	17485
B_2	17305
B_3	1804
B_4	7511
B_5	3192

faire de la reconnaissance d’activités avec eTact. Les trois premières approches utilisent les données brutes fournies dans le capteur, mais diffèrent selon le support pour le calcul des paramètres et la classification des activités. Ces trois différentes approches sont présentées dans le chapitre 4 suivant. La dernière approche a pour originalité de n’utiliser que des données binarisées de l’accéléromètre pour augmenter l’autonomie d’eTact. Cette approche est étudiée dans le chapitre 6.

Comme vu dans chapitre 2, les algorithmes les plus utilisés pour faire de la reconnaissance d’activités sont des algorithmes d’apprentissage supervisé et ces algorithmes requièrent une base d’apprentissage. Deux bases de données utilisées dans la thèse ont été présentées dans ce chapitre. Ces bases de données disponibles dans la littérature contiennent des données d’accéléromètres équivalentes aux données fournies par eTact. Ces bases de données sont utilisées pour développer les différents algorithmes dans la suite de la thèse et permettent de comparer les performances des algorithmes développés pour différents positionnements d’eTact ciblés sur le corps humain : la taille, sur la poitrine, le poignet du bras dominant et la cheville du même coté que le bras dominant.

Chapitre 4

Reconnaissance d'activités avec eTact

4.1 Introduction

Le but de ce chapitre est d'appliquer les différentes techniques de reconnaissance d'activité présentées dans le chapitre 2 sur le dispositif eTact en prenant en compte ses contraintes. Dans le chapitre 3, plusieurs approches sont exposées pour permettre de faire de la reconnaissance d'activités à partir de la plateforme eTact. Les différentes approches varient en fonction du support utilisé pour le calcul des paramètres et la classification des différentes activités. En fonction de l'approche utilisée, les données envoyées ou stockées sur eTact ne sont pas les mêmes et ceci a un impact sur la durée de vie du dispositif. Parmi les quatre approches, les approches 1, 2 et 3 qui utilisent les données brutes de l'accéléromètre sont étudiées dans ce chapitre. L'approche 4 qui utilise les données binarisées de l'accéléromètre est étudiée dans le chapitre 6.

Comme exposé dans le chapitre 2, les algorithmes de reconnaissance d'activités se décomposent en différentes étapes. La première étape est le prétraitement des données, ce prétraitement varie en fonction de l'approche utilisée. La seconde étape consiste en le calcul des paramètres qui vont être utilisés pour l'algorithme de classification. Le choix des paramètres va varier aussi selon le support utilisé pour le calcul. Les différents ensembles de paramètres utilisés par les différentes approches sont décrits dans la section 4.2. Pour chacun des ensembles, le nombre de paramètres est important. Il est donc nécessaire de réduire la dimension de ces ensembles pour améliorer les résultats de classification et réduire la charge de calcul. Les algorithmes de réduction de dimension mis en oeuvre ici sont présentés dans la section 4.3. Ensuite, plusieurs algorithmes de classification supervisée sont utilisés pour la classification des activités. Les résultats des différentes approches sont décrits dans la section 4.7. L'impact de différentes variables, telle que la taille

de la fenêtre glissante et la fréquence d'échantillonnage, est analysé.

4.2 Choix des paramètres

Le choix des paramètres est une étape importante dans la reconnaissance d'activités. Les paramètres les plus utilisés dans la littérature sont exposés dans la section 2.5. Dans notre cas, une première sélection des paramètres est faite en fonction des ressources disponibles pour le calcul de ces paramètres. Si les paramètres sont calculés sur eTact, ce qui est le cas pour les approches 2 et 3, les paramètres choisis correspondent à ceux impliquant la charge de calcul la plus faible possible. Les paramètres choisis sont détaillés dans la section 4.2.2. Pour l'approche 1, il est possible de calculer tous les paramètres possibles à partir des données brutes de l'accéléromètre car les paramètres sont calculés sur un appareil externe qui est supposé disposé d'une capacité de calcul suffisante. Les paramètres choisis pour l'approche 1 sont détaillés dans la section 4.2.1.

4.2.1 *Ensemble_Param_A* : Ensemble de paramètres choisis calculés sur une plateforme externe

Cet ensemble est constitué par les paramètres qui vont être utilisés pour l'approche 1. Pour le calcul de ces paramètres, on dispose d'une capacité de calcul importante (un ordinateur ou un smartphone) qui permet de calculer un grand nombre de paramètres. Les données brutes reçues par l'appareil externe sont les accélérations suivant les 3 axes de l'accéléromètre. À partir de ces trois séries temporelles, on va calculer de nouvelles séries temporelles. Pour commencer, la norme de la l'accéléromètre (définie dans (2.1)), la projection de la norme sur plan xy (définie dans (2.6)), la projection de la norme sur plan yz (définie dans (2.7)) et la projection de la norme sur plan xz (définie dans (2.8)) sont calculées. Ensuite, la partie continue et la partie dynamique sont extraites pour les données des trois axes et les différentes normes. La partie continue est calculée en utilisant une moyenne glissante sur 50 échantillons et la partie dynamique est obtenue en soustrayant la partie continue à la série temporelle. Les angles de rotations sont ensuite calculés à partir de la composante continue des trois axes. Au total, on dispose de 24 séries temporelles à partir desquelles il est possible de calculer des paramètres qui vont servir à la classification.

L'ensemble *Ensemble_Param_A* comprend les paramètres calculés sur une fenêtre suivants :

- La moyenne, la variance et la médiane des 3 axes, des normes et de leur composante continue et dynamique.
- Le maximum, le minimum et l'éventail des 3 axes, des normes et de leur composante continue et dynamique.
- Le 25ème percentile et le 75ème percentile des 3 axes, des normes et de leur composante continue et dynamique.
- La moyenne quadratique des 3 axes, des normes et de leur composante continue et dynamique.
- L'asymétrie et le kurtosis des 3 axes, des normes et de leur composante continue et dynamique.
- Le ZCM de chaque axe et le ZCM de la norme à laquelle on soustrait la gravité.
- Le nombre de pics positifs, négatifs des normes et de chaque axe.
- La moyenne, la variance et la médiane des pics positifs et négatifs des normes et de chaque axe.
- La moyenne, la variance du temps entre deux pics positif et négatif des normes et de chaque axe.
- Le maximum, le minimum et la médiane du temps entre deux pics positif et négatif des normes et de chaque axe.
- La magnitude de l'écart type des 3 axes, des normes et de leur composante continue et dynamique.
- La moyenne de la valeur absolue des écarts des 3 axes, des normes et de leur composante continue et dynamique.
- La moyenne, la variance et la médiane de chaque angle.
- Les coefficients d'un modèle AR des normes et de chaque axe. Le degré du modèle AR est fixé à 7.
- Le coefficient de corrélation entre chaque axe, leurs composantes continues et dynamiques.
- Le maximum de l'intercorrélétion entre chaque axes, ainsi que leurs composantes continues et dynamiques.
- Les coefficients de la TFD de chaque axe et des normes.
- Les fréquences principales de chaque axe et des normes obtenue à partir de la TFD.

- La moyenne des composantes de fréquence pour obtenir une fréquence moyenne de chaque axe et des normes.

Au total, *Ensemble_Param_A* comporte 373 paramètres auxquels il faut ajouter le nombre de paramètres de la TFD qui varie selon le nombre d'échantillons de la fenêtre de temps. Le nombre total des paramètres est obtenu par le calcul suivant :

$$nb_{param} = 373 + 100 * E\left(\frac{f * T_w}{50}\right) \quad (4.1)$$

avec f la fréquence d'échantillonnage de l'accéléromètre, $E(x)$ est la partie entière de x et T_w la longueur de la fenêtre de temps en secondes.

4.2.2 *Ensemble_Param_B* : Ensembles de paramètres calculés sur eTact

Pour l'approche 2 et 3, le calcul des paramètres est effectué sur eTact. il est donc primordial de choisir des paramètres qui nécessitent peu de charge de calcul. Par exemple, les différents paramètres calculés à l'aide de la TFD ou transformée en ondelettes discrète ne sont pas utilisées. De plus, pour limiter la charge de calcul, seulement la norme des trois axes est calculée sur eTact. La normes de deux axes ainsi que les différents angles de rotations ne sont donc pas disponibles pour le calculs de paramètres. Les différents paramètres pourront être calculés à partir de la normes et de l'accélération d'un des trois axes de l'accéléromètre.

L'ensemble *Ensemble_Param_B* comprend des paramètres calculés dans le domaine temporel. Cet ensemble est composé comme suit :

- Le maximum, le minimum et l'éventail de la norme et de chaque axe.
- La moyenne, la variance et la médiane de la norme et de chaque axe.
- Le ZCM de chaque axe et le ZCM de la norme à laquelle on soustrait la gravité.
- Le nombre de pics positifs, négatifs et total de la norme et chaque axe.
- La moyenne, la variance, et la médiane des pics positif et négatif de la norme et chaque axe.
- La moyenne, la variance, du temps entre deux pics positif et négatif de la norme et chaque axe.
- Le maximum, le minimum et la médiane du temps entre deux pics positifs et négatifs de la norme et chaque axe.

Ensemble_Param_B comporte au total 84 paramètres. Le calcul de certains paramètres est modifié pour réduire le nombre de calculs. Par exemple, le calcul de la norme est modifié en supprimant la racine carrée car cette opération induit une charge de calcul importante sur le MCU d'eTact. Toujours dans le but de réduire la charge de calcul, les données de l'accéléromètre ne sont pas converties en g ; les calculs sont directement réalisés sur les données brutes qui correspond à une valeur proportionnel de l'accélération.

4.2.3 Normalisation

Comme vu dans le chapitre 2, les valeurs de chaque paramètre peuvent avoir des intervalles d'appartenance de valeurs très différents. Il est nécessaire de normaliser les paramètres pour améliorer les résultats de la classification. La normalisation min-max est ici appliquée aux deux ensembles de paramètres pour que tous les paramètres soient compris entre 0 et 1. Rappelons que cette normalisation est calculée de la manière suivante :

$$p_{norm_i} = \frac{p_i - \min(p_i)}{\max(p_i) - \min(p_i)} \quad (4.2)$$

où p_i est le paramètre d'index i , $\min(p_i)$ correspond au minimum du paramètre p_i sur toutes les fenêtres, $\max(p_i)$ correspond au maximum du paramètre p_i sur toutes les fenêtres et p_{norm_i} est le paramètre normalisé entre 0 et 1

4.3 Réduction de dimension des ensembles de paramètre

4.3.1 Algorithme de réduction de dimension utilisé

Les ensembles de paramètres *Ensemble_Param_A* et *Ensemble_Param_B* possèdent un nombre élevé de paramètres et certain de ces paramètres peuvent être redondants ou inutiles pour la classification. Ces variables redondantes ou inutiles augmentent la charge de calcul et peuvent détériorer les performances de l'algorithme de reconnaissance d'activités. Il est donc nécessaire de réduire la dimension des ensembles au maximum en supprimant les variables redondantes ou inutiles. Il existe dans la littérature plusieurs méthodes de réduction de dimension (décrites dans 2.5.2). Deux algorithmes de réduction vont être utilisés et sont décrits dans cette section. L'ensemble de paramètres réduit obtenue par chaque algorithme sera ensuite utilisé pour la classification dans la section 4.4.

4.3.1.1 Sélection de paramètres avec l'algorithme ReliefF

L'algorithme ReliefF est une méthode "filter" qui permet la sélection de paramètres. Cet algorithme est décrit dans ROBNIK-SIKONJA et KONONENKO, 2003. Il s'agit d'une variante de l'algorithme Relief qui permet de traiter le cas multi-classes, la présence de bruit ainsi que les données manquantes. Cette approche "filter" a pour avantage de ne pas dépendre de l'algorithme de classification utilisé. L'algorithme ReliefF consiste à estimer la qualité des paramètres à l'aide d'une mesure de distance. La sortie de l'algorithme est un vecteur $W \in R^d$ qui donne le poids de chaque paramètre, poids compris entre 1 et -1. Plus le poids est proche de 1, plus le paramètre est de bonne qualité pour discriminer les différentes classes.

Après avoir initialisé le vecteur de poids à zéros, on choisit un point aléatoire x_i d'une classe de façon aléatoire. Ensuite, les k plus proches voisins H_j appelé hits de la même classe et les k plus proches voisins $M_j(C)$ appelé misses des autres classe sont recherchés. Le vecteur de poids $W(A)$ est ensuite mis à jour en fonction du point x_i , les hits H_j et les misses $M_j(C)$. L'impact des misses $M_j(C)$ est pondéré par la probabilité antérieur $P(C)$ de la classe C estimé à partir de l'ensemble d'entraînement. Chaque poids est divisé par $1 - P(class(R_i))$ qui correspond la somme de probabilités des classes de misses. Ce processus est répété m fois. Le pseudo-code de l'algorithme est présenté dans le tableau 4.1.

4.3.1.2 Sélection de paramètres avec l'algorithme Sequential Forward Selection

Une autre méthode de sélection de paramètre de type "wrapper" est utilisée dans cette section : l'algorithme Sequential Forward Selection (SFS). Cette approche est proposée dans WHITNEY, 1971. La particularité de l'approche "wrapper" est d'utiliser l'algorithme de classification pour sélectionner les paramètres. Au départ de l'algorithme, on dispose d'un sous-ensemble de paramètres vide. A chaque itération, chaque paramètre non encore sélectionné est testé et le paramètre qui obtient le meilleur résultat de classification est ajouté au sous-ensemble. L'algorithme est arrêté quand l'ajout d'un paramètre ne permet plus d'amélioration lors de la classification ou quand le nombre maximal de paramètres M est atteinte.

Algorithme ReliefF

Entrées : Un ensemble $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)\}$ qui contient l'ensemble des points d'apprentissage de longueur L avec $x_i \in \{1, 2, \dots, K\}$ le label du vecteur y_i qui contient d paramètres avec $y_i \in R^d$. K est le nombre de classes possibles.

Sorties : Le vecteur de poids $W \in R^d$ qui définit la qualité de chaque paramètre

- Initialisation du vecteur de poids $W(A) = 0 \forall A$:
- pour $t = 1$ à m (m limité à le nombre de données d'apprentissage N) :
 - Sélection aléatoire d'un point x_i dans la base d'apprentissages T
 - Recherche les k plus proches voisins H_j de la même classe que x_i
 - Pour chaque classe différente de l'instance x_i , faire :
 - recherche des k plus proches voisins $M_j(C)$ de la classe C
 - mise à jour du vecteur de poids avec :

$$W(A) = W(A) - \sum_{j=1}^k \frac{\text{diff}(A, x_i, H_j)}{m.k} \quad (4.3)$$

$$+ \sum_{C \neq \text{class}(R_i)} \frac{\frac{P(C)}{1-P(\text{class}(R_i))} \sum_{j=1}^k \text{diff}(A, x_i, M_j(C))}{m.k} \quad (4.4)$$

avec

$$\text{diff}(A, I_1, I_2) = \begin{cases} 1 & \text{si } \text{valeur}(A, I_1) = \text{valeur}(A, I_2) \\ 0 & \text{sinon} \end{cases} \quad (4.5)$$

TABLE 4.1 – pseudo-code de l'algorithme ReliefF

Algorithme Sequential Forward Selection

Entrées : Un ensemble $= \{(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)\}$ qui contient l'ensemble des points d'apprentissage de longueur L avec $x_i \in \{1, 2, \dots, K\}$ le label du vecteur y_i qui contient d paramètres avec $y_i \in R^d$. K est le nombre de classes possibles.

Sorties : Le sous-ensemble de paramètre réduit E .

- Initialisation du sous-ensemble de paramètre réduit $E = \emptyset$.
- Pour $t = 1$ à M , faire
 - Pour chaque paramètre non encore présent dans le sous-ensemble E , évaluer ce paramètre rajouter à l'ensemble E .
 - Si un des paramètres améliore le résultats de la classification, ole paramètre qui améliore le plus le résultat est rajouté a E
 - Si aucun des paramètres améliore le résultat de classification, on retourne l'ensemble E .

TABLE 4.2 – pseudo-code de l'algorithme Sequential Forward Selection

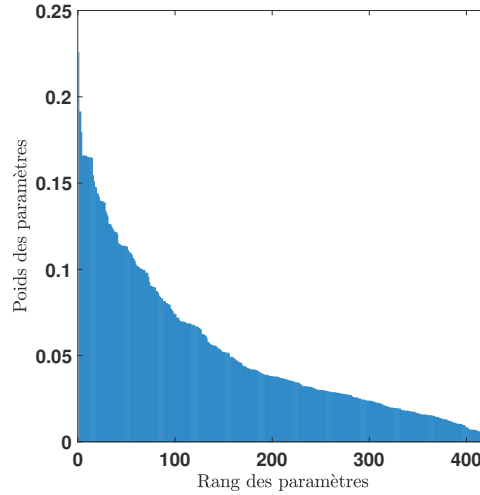


FIGURE 4.1 – Poids des différents paramètres de l'ensemble *Ensemble_Param_A* avec l'utilisation de l'algorithme ReliefF pour $T_W = 1s$

4.3.2 Réduction de l'ensemble *Ensemble_Param_A*

Dans la littérature, il est montré que les approches "wrapper" donnent généralement de meilleurs résultats que les méthodes "filter" comme il a été illustré dans GUPTA et DALLAS, 2014. Cependant, l'algorithme SFS demande beaucoup plus de calculs que l'algorithme ReliefF. Dans le cas de l'ensemble de paramètres *Ensemble_Param_A*, la dimension de l'ensemble de paramètres est trop importante pour utiliser directement l'algorithme SFS. Dans un premier temps, l'algorithme ReliefF est utilisé pour garder les 100 paramètres avec le poids le plus fort. L'algorithme ReliefF est utilisé avec une distance euclidienne et $k = 5$. La figure 4.1 illustre les poids des paramètres de l'ensemble *Ensemble_Param_A* avec l'utilisation de l'algorithme ReliefF pour une longueur de fenêtre $T_W = 1s$. On remarque sur cette figure qu'il y a un grand nombre de paramètres qui ont un poids faible et qui peuvent donc être enlevés de l'ensemble de paramètres. Dans un second temps, l'algorithme SFS est utilisé pour sélectionner les paramètres avec un nombre maximal de paramètre fixé à 30.

4.3.3 Réduction de l'ensemble *Ensemble_Param_B*

Pour l'ensemble de paramètres *Ensemble_Param_B*, la dimension de l'ensemble est beaucoup moins importante. L'algorithme SFS est directement utilisé sur l'ensemble de

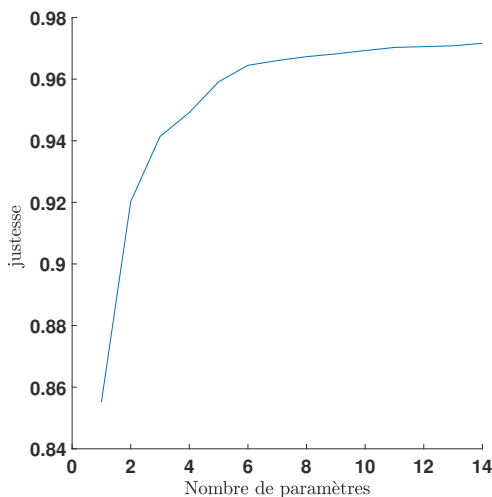


FIGURE 4.2 – Justesse de la classification en fonction du nombre de paramètres avec l’algorithme SFS pour l’algorithme de classification des K plus proches voisins avec $T_w = 2s$ et le capteur positionné au torse

paramètres pour la réduction de dimension. Dans le cas de l’approche 2 et de l’approche 3, les paramètres sont calculés directement sur eTact. Il est important ici de réduire au maximum la dimension de l’ensemble de paramètres car le nombre de paramètres à calculer sur eTact a un impact direct sur la durée de vie de la batterie. L’impact de la consommation de la batterie est encore plus important pour l’approche 3 car, en plus du calcul de paramètre embarqué sur Etact, la mise en oeuvre de la classification est embarquée aussi sur eTact et la charge de calcul des algorithmes de classification augmente si le nombre de paramètres en entrée augmente.

La figure 4.2 montre l’évolution de la justesse en fonction du nombres de paramètre avec l’algorithme SFS pour l’algorithme de classification des K plus proches voisins avec $T_w = 2s$ et le capteur positionné au torse. Cette figure montre que le rajout de paramètre permet d’augmenter la qualité de la classification. Cependant l’amélioration devient de moins en moins importante quand le nombre de paramètres augmente. On limite donc le nombre maximal de paramètres de la réduction de l’ensemble de paramètres *Ensemble_Param_B* à 10.

4.4 Algorithmes d'apprentissage

Les algorithmes d'apprentissage utilisés pour la reconnaissance d'activités sur eTact sont :

- Analyse discriminante linéaire (LDA)
- Analyse discriminante quadratique (QDA)
- Arbre de décision. L'algorithme CART est utilisé pour la construction de l'arbre de décision avec l'indice de diversité de Gini comme critère de division des branches.
- K plus proches voisins avec $k = 5$.
- Les forêts aléatoires. Le nombre d'arbres de décision de l'ensemble d'arbres est fixé à 10.
- Naïve Bayésienne.
- Réseaux de neurones. Un perceptron multicouche avec deux couches est utilisé.

Ces algorithmes de classification sont décrit plus en détail dans la section 2.6. Les calculs de la classification sont réalisés sur un appareil externe pour l'approche 1 et 2 et sur eTact pour l'approche 3. Pour cette raison, le temps de calcul de la fonction de décision qui permet de classifier les nouveaux échantillons aura un impact important sur l'autonomie d'eTact. Au contraire, pour l'approche 1 et 2, le temps de calcul de la fonction de décision n'aura aucun impact sur la durée de vie de la batterie d'eTact. Par exemple, l'algorithme des K plus proches voisins ne pourra pas être implémenté sur eTact car cet algorithme impose que tous les points de la base d'apprentissage soient présents sur eTact et il nécessite un grand nombre de calculs pour classifier de nouveaux points. Les algorithmes LDA ou QDA requièrent des calculs matriciels pour la classification de nouveaux points ce qui implique une charge de calcul non négligeable du MCU d'eTact. Pour l'arbre de décision et les forêts aléatoires, seuls les arbres obtenus à l'issue de l'apprentissage doivent être embarqués sur eTact. Les arbres sont facilement implémentables et consomment peu de ressource. Ces deux algorithmes d'apprentissage seront privilégiés pour l'approche 3.

4.5 Impact de la taille de la fenêtre glissante T_w

La taille de la fenêtre glissante T_w est un paramètre crucial pour la reconnaissance d'activités. Il est important d'avoir une fenêtre de petite taille pour éviter d'avoir plusieurs types d'activités dans une même fenêtre. Les figures 4.3, 4.4, 4.5 et 4.6 illustrent la

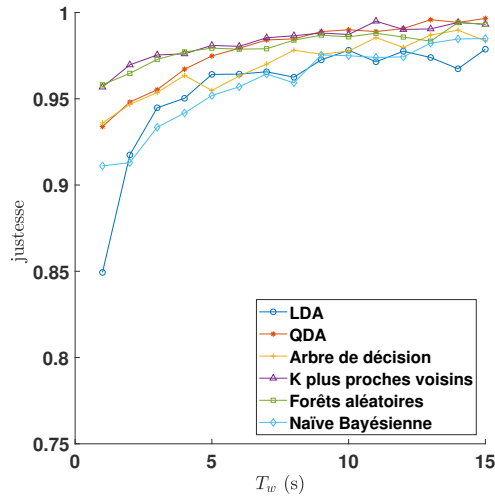


FIGURE 4.3 – Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné au torse pour l'ensemble de paramètres *Ensemble_Param_B* avec $f = 50\text{Hz}$

justesse de la classification en fonction de la taille de la fenêtre glissante T_w pour les différentes positions de l'accéléromètre et pour les différents algorithmes de classification. On remarque sur ces figures que plus la fenêtre de temps est élevée, plus la classification est de bonne qualité. Cette amélioration est surtout significative entre 1s et 5s.

On choisit donc pour la suite de l'étude de fixer la taille de la fenêtre glissante à $T_w = 5\text{s}$ avec un recouvrement de 50%. A partir de ces valeurs, on obtient une granularité de 2.5s. Si on souhaite réduire la taille de la fenêtre de temps, il est possible d'augmenter le pourcentage de recouvrement pour garder $T_w = 5\text{s}$ et ainsi améliorer les résultats de classification.

4.6 Impact de la fréquence d'échantillonnage de l'accéléromètre

La figure 4.7 illustre la justesse de la classification des K plus proches voisins en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné à la taille pour l'ensemble de paramètre *Ensemble_Param_B* pour $f = 25\text{Hz}$ et $f = 50\text{Hz}$. On remarque ici que les résultats de classification sont de meilleure qualité avec une fréquence d'échantillonnage de $f = 50\text{Hz}$. Ce résultat est observé dans la plupart des cas, dans les différentes

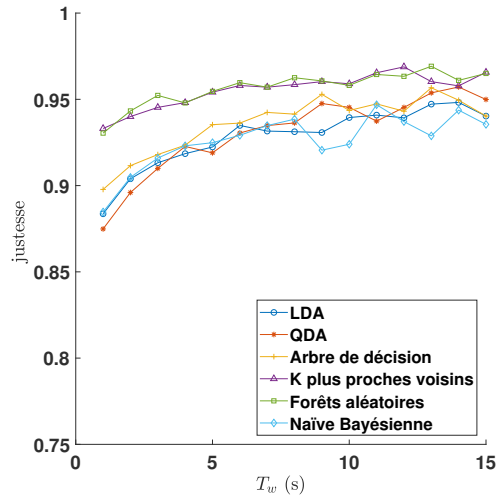


FIGURE 4.4 – Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné au poignet pour l'ensemble de paramètres *Ensemble_Param_B* avec $f = 50\text{Hz}$

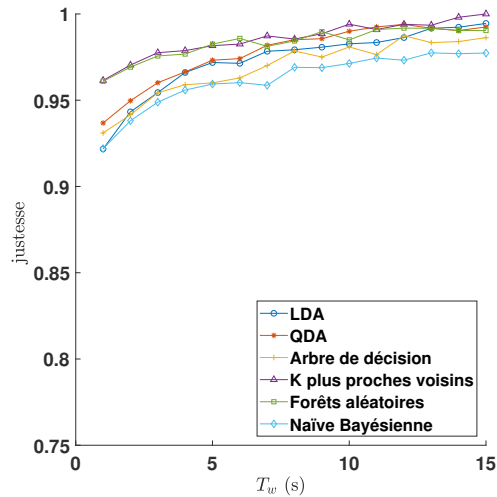


FIGURE 4.5 – Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné à la cheville pour l'ensemble de paramètres *Ensemble_Param_B* avec $f = 50\text{Hz}$

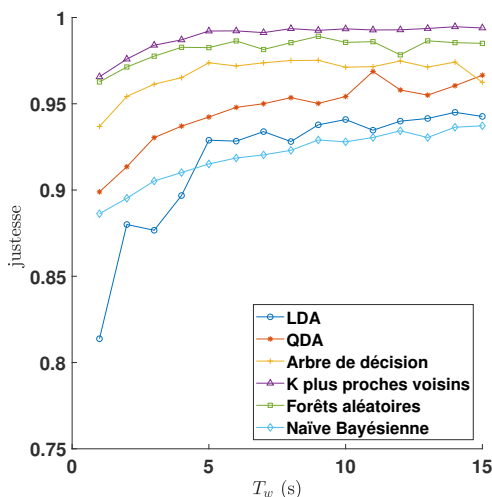


FIGURE 4.6 – Justesse de la classification en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné à la taille pour l'ensemble de paramètres *Ensemble_Param_B* avec $f = 50\text{Hz}$

positions avec les différents algorithmes de classification.

4.7 Comparaison des différents approches

4.7.1 Résultats de l'ensemble de paramètres *Ensemble_Param_A*

Le tableau 4.3 expose la justesse et le tableau 4.4 expose le F1score pour les différentes positions avec les différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_A*. On remarque que tous les algorithmes de classification donnent de bons résultats à part le réseau de neurones. La structure utilisée du réseau de neurones est probablement la cause de ces mauvais résultats. L'augmentation du nombre de couches cachées ou l'utilisation d'autres structures (les réseaux de neurones convolutif par exemple) pourrait améliorer les résultats. Les algorithmes qui donnent les meilleurs résultats sont l'algorithme des K plus proches voisins ($k=5$) et l'algorithme des forêts aléatoires. En terme de positionnement, le poignet donne les moins bons résultats et le torse, la cheville, et la taille donnent des résultats de qualité similaire

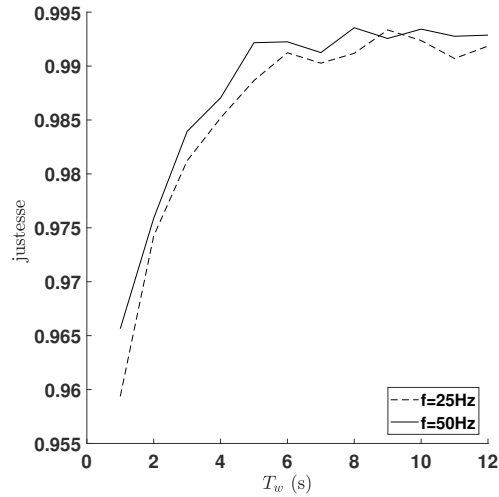


FIGURE 4.7 – Justesse de la classification des k plus proches voisins en fonction de la taille de la fenêtre glissante T_w avec le capteur positionné à la taille pour l'ensemble de paramètres *Ensemble_Param_B* pour $f = 25\text{Hz}$ et $f = 50\text{Hz}$

TABLE 4.3 – Justesse des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_A* avec $T_w = 5\text{s}$

position fréquence	torse		poignet		cheville		taille	
	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz
LDA	0,913	0,971	0,930	0,942	0,975	0,922	0,922	0,963
QDA	0,976	0,981	0,924	0,936	0,970	0,976	0,962	0,982
Arbre de décision	0,969	0,969	0,931	0,937	0,969	0,968	0,972	0,977
K plus proches voisins	0,980	0,984	0,952	0,964	0,984	0,983	0,993	0,995
Forêts aléatoires	0,980	0,980	0,952	0,963	0,977	0,980	0,9885	0,987
Naïve Bayésienne	0,957	0,964	0,928	0,936	0,962	0,972	0,916	0,955
Réseaux de neurones	0,860	0,847	0,808	0,784	0,834	0,867	0,836	0,869

TABLE 4.4 – F1score des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_A* avec $T_w = 5s$

position	torse		poignet		cheville		taille	
	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz
LDA	0,887	0,961	0,913	0,935	0,919	0,953	0,895	0,943
QDA	0,971	0,978	0,913	0,936	0,947	0,967	0,949	0,977
Arbre de décision	0,960	0,961	0,925	0,9379	0,945	0,945	0,937	0,968
K plus proches voisins	0,977	0,982	0,949	0,964	0,973	0,971	0,992	0,993
Forêts aléatoires	0,975	0,975	0,945	0,958	0,965	0,968	0,986	0,984
Naïve Bayésienne	0,946	0,960	0,913	0,930	0,937	0,953	0,889	0,928
Réseaux de neurones	0,541	0,598	0,584	0,567	0,618	0,547	0,507	0,537

TABLE 4.5 – Justesse des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_B* avec $T_w = 5s$

position	torse		poignet		cheville		taille	
	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz
LDA	0,963	0,964	0,932	0,922	0,964	0,971	0,930	0,928
QDA	0,967	0,974	0,927	0,919	0,970	0,973	0,942	0,942
Arbre de décision	0,967	0,954	0,931	0,935	0,957	0,960	0,966	0,973
K plus proches voisins	0,981	0,980	0,959	0,954	0,980	0,981	0,988	0,992
Forêts aléatoires	0,974	0,979	0,953	0,954	0,977	0,982	0,983	0,982
Naïve Bayésienne	0,942	0,951	0,928	0,924	0,953	0,959	0,915	0,915
Réseaux de neurones	0,850	0,848	0,823	0,820	0,832	0,856	0,841	0,818

4.7.2 Résultats de l'ensemble de paramètres *Ensemble_Param_B*

Le tableau 4.5 expose la justesse et le tableau 4.6 expose le F1score pour les différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_B*. Comme pour l'ensemble de paramètres *Ensemble_Param_A*, on remarque que le réseau de neurones ne donne pas de bons résultats. On observe en moyenne une petite perte de qualité de la classification mais les résultats sont similaires à l'ensemble de paramètres *Ensemble_Param_A* bien que l'ensemble de paramètres *Ensemble_Param_B* possède moins de paramètres que l'ensemble *Ensemble_Param_A*. Les algorithmes qui donnent les meilleurs résultats sont aussi l'algorithme K plus proches voisins et l'algorithme des forêts aléatoires. Pour ce qui est de la position de eTact, ici encore le poignet donne de moins bons résultats que le torse, la cheville, et la taille.

TABLE 4.6 – F1score des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_B* avec $T_w = 5s$

position	torse		poignet		cheville		taille	
	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz
LDA	0,949	0,955	0,927	0,912	0,944	0,954	0,905	0,901
QDA	0,961	0,974	0,919	0,907	0,952	0,957	0,930	0,933
Arbre de décision	0,956	0,947	0,924	0,930	0,933	0,936	0,960	0,963
K plus proches voisins	0,977	0,978	0,958	0,951	0,972	0,973	0,987	0,989
Forêts aléatoires	0,969	0,985	0,949	0,953	0,965	0,975	0,980	0,978
Naïve Bayésienne	0,918	0,939	0,917	0,919	0,927	0,935	0,898	0,887
Réseaux de neurones	0,531	0,598	0,583	0,503	0,607	0,629	0,509	0,596

4.7.3 Autonomie d'eTact en fonction de l'approche utilisée

Une des contraintes importantes d'eTact est l'autonomie liée à la mémoire. L'autonomie mémoire d'eTact en fonction des différentes approches est résumée dans le tableau 4.7. eTact possède une mémoire flash avec une capacité de 64 Mo pour stocker des données avant de les envoyer vers une plateforme externe. Toute la mémoire n'est pas disponible car environ 1 Mo est réservé pour le fonctionnement d'eTact.

On peut observer dans le tableau 4.7 que l'autonomie mémoire varie selon l'approche utilisée. Pour l'approche 1, les données brutes sont directement stockées dans la mémoire avant d'être envoyées vers une plateforme externe. L'autonomie dépend de la fréquence d'échantillonnage de l'accéléromètre et de la taille des données de l'accéléromètre (codées sur 12 bits). Avec cette approche, eTact possède une autonomie mémoire de 5h49 si eTact fonctionne en 50Hz et une autonomie mémoire de 11h38 si eTact fonctionne en 25Hz.

Pour l'approche 2, les paramètres sont calculés sur eTact avant d'être stockés dans eTact. L'autonomie dépend du nombre de paramètres, du nombre de bit nécessaire pour coder les paramètres et de la taille de la fenêtre T_w . L'avantage d'utiliser cette approche est de diminuer le nombre de données à stocker dans la mémoire par rapport à l'approche 1. Par exemple, eTact a une autonomie mémoire de 87h15 dans le cas de l'utilisation de 10 paramètres et d'une fenêtre de temps de $T_w = 2s$. On remarque cependant que l'autonomie mémoire diminue si on augmente le nombre de paramètres à stocker et si on diminue la taille T_w de la fenêtre. Pour la troisième approche, la classification est directement faite sur eTact, seul le résultat de la classification est stocké pour chaque fenêtre. L'autonomie

TABLE 4.7 – Autonomie mémoire eTact en fonction des différentes approches en utilisant l'algorithme des forêts aléatoires

Approches	1	2 3 paramètres ($T_w = 5$)	2 10 paramètres ($T_w = 2$)	2 10 paramètres ($T_w = 5$)	3 ($T_w = 5$)
f=25Hz	11h38	290h50	87h15	218h7	4362h
f=50Hz	5h49	290h50	87h15	218h7	4362h

dépend ainsi principalement de la fenêtre de temps T_w . Cette approche possède donc l'autonomie mémoire la plus élevée (1745h pour une fenêtre de temps de $T_w = 2s$).

Cependant l'autonomie ne se limite pas à la mémoire. L'autonomie de la batterie est aussi une contrainte importante à prendre en compte sur eTact. L'autonomie de la batterie d'eTact en fonction des différentes approches est résumée dans le tableau 4.8. L'autonomie de la batterie est déterminé à partir de la consommation mesurée sur Etact. Pour commencer, on remarque que la fréquence d'échantillonnage de l'accéléromètre a un impact sur toutes les approches. Cette observation peut simplement être expliquée par le fait que l'accéléromètre a besoin de plus d'énergie pour fonctionner avec une fréquence plus élevée. De plus, pour les approches 2 et 3, l'augmentation de la fréquence augmente le nombre de calculs pour une fenêtre de temps donnée ce qui réduit aussi l'autonomie de la batterie.

On remarque que pour l'approche 1, l'autonomie de la batterie est supérieure à l'autonomie de la mémoire. C'est la mémoire qui limite l'autonomie générale d'eTact. Pour l'approche 2, on remarque que l'autonomie de la batterie est moins élevée que pour l'approche 1. Cette observation peut être expliquée par le fait que les calculs des paramètres entraînent de la consommation d'énergie. Cependant, l'écriture dans la mémoire flash entraîne aussi de la consommation d'énergie et le nombre de données enregistrées dans la mémoire est moins élevé que l'approche 1. On déduit donc que la diminution du nombre de données en mémoire ne permet pas de compenser la consommation d'énergie des calculs des paramètres. De plus, on remarque que si on diminue le nombre de paramètres ou on augmente le temps de la fenêtre glissante, alors l'autonomie de la batterie est plus élevée. On obtient pour l'approche 3 l'autonomie batterie la moins élevée à cause des calculs de la classification embarquée sur eTact. On observe que pour les approches 2 et 3, c'est l'autonomie de la batterie qui limite l'autonomie générale du système.

TABLE 4.8 – Autonomie de la batterie d’eTact en fonction des différentes approches en utilisant l’algorithme des forêts aléatoires

Approches	1	2	2	2	3
		3 paramètres ($T_w = 5$)	10 paramètres ($T_w = 2$)	10 paramètres ($T_w = 5$)	($T_w = 5$)
f=25Hz	84h	52h	35h	29h	28h
f=50Hz	72h	40h	30h	36h	24h

4.8 Conclusion

Dans ce chapitre, des algorithmes de reconnaissance d’activités avec eTact sont mis en oeuvre. Les trois approches qui utilisent les données brutes de l’accéléromètre sont étudiées. Deux ensembles de paramètres sont construits. Le premier ensemble correspond aux paramètres qui peuvent être calculés sur une plateforme externe. On dispose d’une grande capacité de calcul. Ces paramètres sont utilisés par la première approche. Pour le deuxième ensemble, il correspond aux paramètres destinés à être calculés sur eTact. Les paramètres de cet ensemble sont choisis pour limiter la charge de calcul sur Etact et garder ainsi une autonomie de batterie la plus élevée possible. Ces paramètres sont utilisés pour les approches 2 et 3.

Les deux ensembles de paramètres comportent plus d’une centaine de paramètres. Deux algorithmes de réduction de dimensions sont choisis pour réduire les dimensions de ces ensembles. Le premier algorithme est l’algorithme ReliefF qui utilise une approche filter. Le deuxième algorithme est SFS qui utilise une approche wrapper. Ces deux algorithmes permettent de sélectionner les paramètres qui ont le plus d’impact dans la classification et d’écartier les paramètres redondants. La réduction de dimension permet ainsi d’améliorer le résultat de classification et de diminuer la charge de calcul.

L’impact de variables importantes dans la reconnaissance d’activités est analysé. On peut ainsi observer que la taille de la fenêtre glissante et la fréquence d’échantillonnage ont une influence sur la performance de la classification. Il a été choisi de fixer la fenêtre de temps à 5 secondes pour avoir le meilleur compromis entre les performances de classification et le caractère temps réel de la reconnaissance d’activités.

Pour finir, des résultats de la classification en fonction du positionnement du capteur sont comparés. Une analyse sur l’autonomie d’eTact en fonction de l’approche choisie est

aussi proposée.

Chapitre 5

Identification d'un modèle autorégressif à partir de données binaires ou quantifiées

5.1 Introduction

Dans le chapitre précédent, les données de l'accéléromètre sont de bonne qualité avec une résolution élevée et peu de bruit. Dans certains cas, on n'a pas accès à un capteur avec une résolution suffisante. La raison peut être économique, car un capteur avec une plus grande résolution est souvent plus onéreux qu'un capteur avec une résolution plus restreinte ou simplement par ce qu'il n'existe pas de capteur disponible avec la résolution requise. Dans notre cas, une contrainte technique nous mène à vouloir utiliser des données avec une résolution faible. Dans le cas où les données de l'accéléromètre sont stockées, puis envoyées vers un système externe (smartphone ou ordinateur) pour faire de la reconnaissance d'activités, la durée de vie d'eTact est limitée par la durée de vie de la batterie et la capacité de la mémoire externe d'eTact qui permet de stocker les données de l'accéléromètre. L'utilisation de données binaires ou quantifiées permet d'augmenter le nombre maximal de données stockées dans la mémoire et de réduire le nombre de données envoyées par BLE. Une donnée de l'accéléromètre est initialement codée sur 12 bits, et deux octets sont nécessaires pour stocker cette donnée dans la mémoire. Avec des données binaires, il est possible de stocker 16 données de l'accéléromètre sur deux octets. De plus, pour une même durée, le nombre de données à envoyer par BLE est 16 fois moins important, ce qui a un impact positif sur la durée de vie de la batterie, car le BLE est la partie qui consomme le plus sur eTact.

L'utilisation de données binaires a pour principal défaut la perte importante d'information par rapport aux données initiales. Dans le cas de la reconnaissance d'activités, des paramètres ne peuvent plus être utilisés comme par exemple le maximum ou le minimum. Comme vu dans le chapitre 2, les paramètres d'un modèle AR du signal peuvent être utilisés comme paramètres pour la reconnaissance d'activités. L'objet de ce chapitre est de mettre en évidence qu'il est possible d'estimer les paramètres d'un modèle AR d'un signal à partir des données binaires de ce même signal. Ces paramètres pourront être ensuite utilisés pour la reconnaissance d'activités.

Dans un premier temps, l'algorithme pour l'estimation d'un modèle AR d'un signal binarisé avec un seuil égal à zéro sera présenté. Ensuite, un nouvel algorithme est présenté dans le cas où le seuil pour binariser les données est différent de zéro. Pour continuer, la méthode présentée sera modifiée pour être utilisée en temps réel, utilisée avec une série temporelle non-linéaire ou dans le cas où le signal est quantifié avec plusieurs niveaux de quantification. Pour finir, ce résultat sera appliqué à un problème de reconnaissance d'activités avec des données réelles.

Les résultats proposés dans ce chapitre sont publiés dans AUBER et al., 2018a, AUBER et al., 2018b et dans AUBER et al., 2019b (en cours d'expertise).

5.2 Formulation du problème

Considérons deux séries temporelles $\{y(t)\}$ et $\{z(t)\}$ de la forme suivante :

$$\begin{cases} y(t) = \phi_1 y(t-1) + \dots + \phi_p y(t-p) + e(t) \\ z(t) = Q(y(t)) \end{cases} \quad (5.1)$$

où :

- $\{y(t)\}$ est décrit par un modèle AR de degré p et de paramètres ϕ_i . $\{y(t)\}$ est une série temporelle modélisant la sortie du capteur.
- l'opérateur $Q(\cdot)$ modélise le capteur binaire de la manière suivante :

$$Q(y(t)) = \begin{cases} 1 & \text{if } y(t) \geq C \\ 0 & \text{if } y(t) < C \end{cases} \quad (5.2)$$

- $\{z(t)\}$ modélise la sortie binaire du capteur.

- $\{e(t)\}$ est un bruit blanc stationnaire de variance finie. Il s'agit du bruit générateur.
- C est un seuil constant qui peut être différent de 0.

On définit $\theta \in \mathbb{R}^n$ le vecteur de paramètres du modèle AR de la manière suivante :

$$\theta = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix} \quad (5.3)$$

L'objectif est d'estimer le vecteur θ en utilisant uniquement le signal $\{z(t)\}$.

Les hypothèses suivantes sont considérées :

Hypothèse 1 : Le degré p du modèle AR est supposé connu.

Hypothèse 2 : Le seuil C est supposé connu.

On définit :

- $\mu_y = \mathcal{E}\{y(t)\}$ est la moyenne de la série temporelle $\{y(t)\}$
- $\sigma_y^2 = \mathcal{E}\{y(t)^2\}$ est la variance de la série temporelle $\{y(t)\}$
- $\rho_y(i) = \mathcal{E}\{y(t)y(t-i)\}$ est l'autocorrélation de $\{y(t)\}$ avec un décalage i . Pour $i \geq 0$, une estimation de $\rho_y(i)$ à partir de N échantillons est donnée par :

$$\hat{\rho}_y(i) = \frac{1}{N-i} \sum_{t=i+1}^N y_t y_{t-i} \quad (5.4)$$

- $\rho_{y^{(1)},y^{(2)}}(i) = \mathcal{E}\{y^{(1)}(t)y^{(2)}(t-i)\}$ est l'intercorrélacion entre $\{y^{(1)}(t)\}$ et $\{y^{(2)}(t)\}$ avec un décalage i . Une estimation de $\rho_{y^{(1)},y^{(2)}}(i)$ à partir de N échantillons est donnée par :

$$\hat{\rho}_{y^{(1)},y^{(2)}}(i) = \frac{1}{N-i} \sum_{t=i+1}^N y^{(1)}(t)y^{(2)}(t-i) \quad (5.5)$$

- $\{\bar{z}_t\}$ est la séquence complémentaire de $\{z_t\}$ définie par :

$$\bar{z}(t) = 1 - z(t) \quad (5.6)$$

Il existe plusieurs solutions pour estimer les paramètres du modèle AR à partir de l'estimation $\hat{\rho}_y(i)$ de l'autocorrélation $\rho_y(i)$ (voir ZHEN et BASAWA, 2010). La solution

usuelle est celle des équations de Yule Walker. Cette solution est la suivante :

$$\hat{\theta} = M(\hat{\rho}_y)^{-1}N(\hat{\rho}_y) \quad (5.7)$$

avec

$$N(\hat{\rho}_y) = \begin{bmatrix} \hat{\rho}_y(1) \\ \hat{\rho}_y(2) \\ \vdots \\ \hat{\rho}_y(p) \end{bmatrix} \quad (5.8)$$

et

$$M(\hat{\rho}_y) = \begin{bmatrix} \hat{\rho}_y(0) & \hat{\rho}_y(1) & \hat{\rho}_y(2) & \dots & \hat{\rho}_y(p-1) \\ \hat{\rho}_y(1) & \hat{\rho}_y(0) & \hat{\rho}_y(1) & \dots & \hat{\rho}_y(p-2) \\ \vdots & \vdots & \vdots & & \vdots \\ \hat{\rho}_y(p-1) & \hat{\rho}_y(p-2) & \hat{\rho}_y(p-3) & \dots & \hat{\rho}_y(0) \end{bmatrix} \quad (5.9)$$

Cependant, la série temporelle $\{y(t)\}$ ici n'est pas connue et il n'est donc pas possible de calculer directement l'autocorrélation $\rho_y(i)$ à partir de la série temporelle $\{y(t)\}$. Une solution pour estimer l'autocorrélation $\hat{\rho}_y(i)$ à partir de la série temporelle $\{z(t)\}$ existe dans la littérature dans le cas où le seuil $C = 0$. Cette solution va être présentée dans la section suivante 5.3. Ensuite, nous présentons dans la section 5.4 notre solution dans le cas où le seuil C est non nul.

5.3 Estimation d'un modèle AR pour un seuil égal à 0

Il y a déjà des solutions proposées dans la littérature pour l'estimation de $\rho_y(i)$ dans le cas d'un seuil nul : KEDEM, 1980, KEDEM et SLUD, 1981, BAGNALL et JANACEK, 2004, BAGNALL et JANACEK, 2005 et ZHEN et BASAWA, 2010. Toutes ces solutions sont basées sur un même résultat présenté dans SHEPPARD, 1899. Ce résultat permet de calculer l'autocorrélation $\rho_y(i)$ à partir des équations suivantes :

$$\rho_y(i) = \sigma_y^2 \cos \left(\pi \frac{R(i)}{P(i) + R(i)} \right) \quad (5.10)$$

ou encore :

$$\rho_y(i) = \sigma_y^2 \cos \left(\pi - \pi \frac{P(i)}{P(i) + R(i)} \right) \quad (5.11)$$

où $P(i)$ et $R(i)$ correspond aux probabilités suivantes :

$$P(i) = \mathcal{P}_r\{y(t) \geq 0, y(t-i) \geq 0\} \quad (5.12)$$

$$R(i) = \mathcal{P}_r\{y(t) \geq 0, y(t-i) < 0\} \quad (5.13)$$

Dans le plan $(y(t); y(t-i))$, $P(i)$ correspond à la probabilité que $y(t) \geq 0$ et $y(t-i) \geq 0$. Comme $z(t) = Q(y(t)) = 1$ si $y(t) \geq 0$ et $z(t-i) = Q(y(t-i)) = 1$ si $y(t-i) \geq 0$, on obtient l'égalité suivante :

$$\mathcal{E}\{z(t) = 0, z(t-i) = 0\} = \mathcal{E}\{y(t) \geq 0, y(t-i) \geq 0\} = P(i) \quad (5.14)$$

Et on peut en déduire que la probabilités $P(i)$ correspond aussi à l'autocorrélation $\rho_z(i)$.

$$P(i) = \rho_z(i) \quad (5.15)$$

Comme $\{y_t\}$ suit une distribution normale de moyenne nulle, on a :

$$P(i) + R(i) = \frac{1}{2} = \mathcal{P}_r\{y(t) \geq 0\} \quad (5.16)$$

On montre alors, à partir de l'équation (5.11), qu'on retrouve le résultat utilisé dans DAVID, 1953 :

$$\rho_y(i) = \sigma_y^2 \cos(\pi(1 - 2\rho_z(i))) \quad (5.17)$$

Dans KEDEM, 1980, BAGNALL et JANACEK, 2005 et ZHEN et BASAWA, 2010, une équation équivalente est utilisée :

$$\rho_y(i) = \sigma_y^2 \sin\left(2\pi\left(\rho_z(i) - \frac{1}{4}\right)\right) \quad (5.18)$$

Le tableau 5.1 résume l'algorithme pour l'estimation d'un modèle AR partir de données binaires avec un seuil nul.

5.4 Estimation d'un modèle AR pour un seuil différent de 0

5.4.1 Algorithme proposé

Avec le seuil C non nul, la solution présentée dans le section précédente ne peut pas être utilisée. L'autocorrélation $\rho_z(i)$ est différente de la probabilité $P(i)$ car $\rho_z(i) = \mathcal{E}\{y(t) \geq C, y(t-i) \geq C\}$ est différent de $\mathcal{E}\{y(t) \geq 0, y(t-i) \geq 0\} = P(i)$.

Algorithme pour un seuil nul

— Estimation de l'autocorrélation $\rho_z(i)$ pour $i \in \{1; \dots; n\}$:

$$\hat{\rho}_z(i) = \frac{1}{N-i} \sum_{t=i+1}^N z_t z_{t-i} \quad (5.19)$$

— Estimation de l'autocorrélation $\rho_y(i)$ pour $i \in \{1; \dots; n\}$:

$$\hat{\rho}_y(i) = \sigma_y^2 \sin \left(2\pi \left(\hat{\rho}_z(i) - \frac{1}{4} \right) \right) \quad (5.20)$$

— Estimation du vecteur de paramètre θ à partir des équations de Yules Walker :

$$\hat{\theta} = M(\hat{\rho}_y)^{-1} N(\hat{\rho}_y) \quad (5.21)$$

TABLE 5.1 – Résumé de l'algorithme pour un seuil $C = 0$

Comme les probabilités $P(i)$ et $R(i)$ ne sont plus accessibles dans le cas où $C \neq 0$, on définit les trois probabilités suivantes :

$$P_C = \mathcal{P}_r \{y(t) \geq C, y(t-i) \geq C\} \quad (5.22)$$

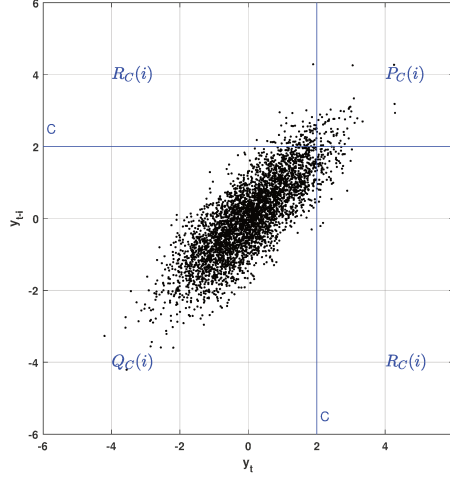
$$\begin{aligned} R_C &= \mathcal{P}_r \{y(t) \geq C, y(t-i) < C\} \\ &= \mathcal{P}_r \{y(t) < C, y(t-i) \geq C\} \end{aligned} \quad (5.23)$$

$$Q_C = \mathcal{P}_r \{y(t) < C, y(t-i) < C\} \quad (5.24)$$

Chacune de ces trois proportions peut être déterminée avec la série temporelle $\{z(t)\}$:

$$P_C = \mathcal{E} \{z(t)z(t-i)\} = \rho_z(i) \quad (5.25)$$

$$\begin{aligned} R_C &= \mathcal{E} \{z(t)\bar{z}(t-i)\} = \rho_{z\bar{z}}(i) \\ &= \mathcal{E} \{\bar{z}(t), z(t-i)\} = \rho_{\bar{z}z}(i) \end{aligned} \quad (5.26)$$

FIGURE 5.1 – Représentation graphique $P_C(i)$, $Q_C(i)$ et $R_C(i)$ pour $C = 2$.

$$Q_C = \mathcal{E}\{\bar{z}(t)\bar{z}(t-i)\} = \rho_{\bar{z}}(i) \quad (5.27)$$

Ces trois proportions sont illustrées dans le plan $(y(t); y(t-i))$ dans la figure 5.1.

Comme $\{y(t)\}$ suit une loi normale centrée, la fonction de distribution du couple $(y(t); y(t-i))$ est définie par :

$$\psi(y(t), y(t-i)) = \frac{1}{2\pi\sigma_y^2 \sqrt{1 - \frac{\rho_y(i)^2}{\sigma_y^4}}} e^{-\frac{\frac{y(t)^2}{\sigma_y^2} + \frac{y(t-i)^2}{\sigma_y^2} - 2\frac{\rho_y(i)}{\sigma_y^2} \frac{y(t)y(t-i)}{\sigma_y^2}}{2\left(1 - \frac{\rho_y(i)^2}{\sigma_y^4}\right)} \quad (5.28)$$

En utilisant la fonction de distribution $\psi(y(t), y(t-i))$, les trois proportions P_C , R_C et Q_C peuvent être exprimées en fonction de l'autocorrélation $\rho_y(i)$ de la manière suivante :

$$P_C = \int_C^{+\infty} \int_C^{+\infty} \psi(y(t), y(t-i)) \, dy(t) dy(t-i) \quad (5.29)$$

$$\begin{aligned} R_C &= \int_{-\infty}^C \int_C^{+\infty} \psi(y(t), y(t-i)) \, dy(t) dy(t-i) \\ &= \int_C^{+\infty} \int_{-\infty}^C \psi(y(t), y(t-i)) \, dy(t) dy(t-i) \end{aligned} \quad (5.30)$$

$$Q_C = \int_{-\infty}^C \int_{-\infty}^C \psi(y(t), y(t-i)) \, dy(t) dy(t-i) \quad (5.31)$$

Remarque : L'équation (5.28) utilise la variance σ_y^2 . Une estimation de cette variance σ_y^2 est donc nécessaire pour pouvoir exprimer les proportions $P_C(\rho_y(i))$, $R_C(\rho_y(i))$ et $Q_C(\rho_y(i))$ en fonction de l'autocorrélation $\rho_y(i)$. Comme on suppose que $\{y(t)\}$ suit une distribution normale, la fonction de répartition est définie par la fonction $F(\cdot)$ suivante :

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt \quad (5.32)$$

Il est alors possible d'accéder à l'écart type σ_y comme suit :

$$\sigma_y = \frac{C}{F^{-1}(1 - \mu_z)} \quad (5.33)$$

Cet algorithme n'est pas donc pas utilisable avec un seuil $C = 0$. Dans le cas d'un seuil C proche de zéro, l'estimation de la variance est moins fiable ce qui entraîne une baisse des performances de l'algorithme.

Il peut être démontré que $P_C(\rho_y(i))$, $R_C(\rho_y(i))$ et $Q_C(\rho_y(i))$ sont des fonctions continues et monotones de $\rho_y(i)$. Il est possible de définir les fonctions $P_C^{-1}(\cdot)$, $R_C^{-1}(\cdot)$ et $Q_C^{-1}(\cdot)$ tel que :

$$P_C^{-1}(P_C(x)) = x \quad (5.34)$$

$$R_C^{-1}(R_C(x)) = x \quad (5.35)$$

$$Q_C^{-1}(Q_C(x)) = x \quad (5.36)$$

Il suffit donc de calculer l'estimation de $\rho_y(i)$ à partir d'une de ces trois équations :

$$\hat{\rho}_y(i) = P_C^{-1}(\hat{\rho}_z(i)) \quad (5.37)$$

$$\hat{\rho}_y(i) = R_C^{-1}(\hat{\rho}_{z,\bar{z}}(i)) \quad (5.38)$$

$$\hat{\rho}_y(i) = Q_C^{-1}(\hat{\rho}_{\bar{z}}(i)) \quad (5.39)$$

Cependant, il n'existe pas d'expression littérale des fonctions P_C^{-1} , R_C^{-1} et Q_C^{-1} . Par conséquent, dans la pratique, on minimise un des trois critères pour obtenir ci-dessous une

estimation de $\rho_y(i)$ avec une estimation de $\rho_z(i)$, $\rho_{z,\bar{z}}(i)$ et $\rho_{\bar{z}}(i)$:

$$\hat{\rho}_y(i) = \underset{\rho_y(i)}{\operatorname{argmin}} \left\{ \left| \hat{\rho}_z(i) - P_C(\rho_y(i)) \right| \right\} \quad (5.40)$$

$$\hat{\rho}_y(i) = \underset{\rho_y(i)}{\operatorname{argmin}} \left\{ \left| \hat{\rho}_{z,\bar{z}}(i) - R_C(\rho_y(i)) \right| \right\} \quad (5.41)$$

$$\hat{\rho}_y(i) = \underset{\rho_y(i)}{\operatorname{argmin}} \left\{ \left| \hat{\rho}_{\bar{z}}(i) - Q_C(\rho_y(i)) \right| \right\} \quad (5.42)$$

Utiliser une seule de ces solutions est suffisant pour estimer l'autocorrélation de $\rho_y(i)$. Cependant l'utilisation des trois est préférable pour avoir une estimation plus précise de l'autocorrélation de $\rho_y(i)$. On peut alors définir les matrices $\widehat{\mathcal{M}}(i)$ et $\mathcal{M}(i, \rho_y(i))$ de telle sorte que :

$$\widehat{\mathcal{M}}(i) = \begin{pmatrix} \hat{\rho}_{z,\bar{z}}(i) & \hat{\rho}_z(i) \\ \hat{\rho}_{\bar{z}}(i) & \hat{\rho}_{z,\bar{z}}(i) \end{pmatrix} \quad (5.43)$$

$$\mathcal{M}(\rho_y(i)) = \begin{pmatrix} R_C(\rho_y(i)) & P_C(\rho_y(i)) \\ Q_C(\rho_y(i)) & R_C(\rho_y(i)) \end{pmatrix} \quad (5.44)$$

Il est alors possible d'estimer l'autocorrélation de $\rho_y(i)$ en minimisant la fonction suivante :

$$\hat{\rho}_y(i) = \underset{\rho_y(i)}{\operatorname{argmin}} \left\{ \left\| \widehat{\mathcal{M}}(i) - \mathcal{M}(\rho_y(i)) \right\|_F \right\} \quad (5.45)$$

où $\|\cdot\|_F$ est la norme de Frobenius. L'algorithme ainsi proposé est résumé dans le tableau 5.2.

5.4.2 Résultats de convergence

Le théorème ci-dessous établit une première propriété sur l'algorithme proposé.

Théorème :

Algorithme pour un seuil non nul

— Estimation de la variance σ_y^2 :

$$\sigma_y = \frac{C}{F^{-1}(1 - \mu_z)} \quad (5.46)$$

— Estimation des autocorrélations $\rho_z(i)$, $\rho_{z,\bar{z}}(i)$ et $\rho_{\bar{z}}(i)$ pour $i \in \{1; \dots; n\}$:

$$\hat{\rho}_z(i) = \frac{1}{N-i} \sum_{t=i+1}^N z(t)z(t-i) \quad (5.47)$$

$$\hat{\rho}_{z,\bar{z}}(i) = \frac{1}{N-i} \sum_{t=i+1}^N z(t)\bar{z}(t-i) \quad (5.48)$$

$$\hat{\rho}_{\bar{z}}(i) = \frac{1}{N-i} \sum_{t=i+1}^N \bar{z}(t)\bar{z}(t-i) \quad (5.49)$$

— Estimation de l'autocorrélation $\rho_y(i)$ pour $i \in \{1; \dots; n\}$:

$$\hat{\rho}_y(i) = \underset{\rho_y(i)}{\operatorname{argmin}} \left\{ \left\| \widehat{\mathcal{M}}(i) - \mathcal{M}(\rho_y(i)) \right\|_F \right\} \quad (5.50)$$

— Estimation du vecteur de paramètre θ à partir des équations de Yules Walker :

$$\hat{\theta} = M(\hat{\rho}_y)^{-1}N(\hat{\rho}_y) \quad (5.51)$$

TABLE 5.2 – Résumé de l'algorithme pour un seuil $C \neq 0$

Si on considère les hypothèses présentées dans la section 5.2, l'algorithme présenté dans le tableau 5.2 est asymptotiquement non biaisé :

$$\lim_{N \rightarrow \infty} \hat{\theta} = \theta \quad (5.52)$$

■

Démonstration :

L'estimation de l'autocorrélation $\hat{\rho}_z(i)$ est obtenue à l'aide de l'équation (5.4) qui donne une estimation asymptotiquement non biaisée de $\rho_z(i)$. On obtient :

$$\lim_{N \rightarrow \infty} \hat{\rho}_z(i) = \rho_z(i) \quad (5.53)$$

$\hat{\rho}_y(i)$ est obtenue à partir de $\hat{\rho}_y(i) = P_C^{-1}(\hat{\rho}_z(i))$, $\hat{\rho}_y(i) = R_C^{-1}(\hat{\rho}_z(i))$ et $\hat{\rho}_y(i) = Q_C^{-1}(\hat{\rho}_z(i))$. Comme $P_C^{-1}(\cdot)$, $Q_C^{-1}(\cdot)$ et $R_C^{-1}(\cdot)$ sont continues, on a :

$$\lim_{N \rightarrow \infty} \hat{\rho}_y(i) = \rho_y(i) \quad (5.54)$$

Finalement, on peut conclure que l'algorithme converge :

$$\lim_{N \rightarrow \infty} \hat{\theta} = \theta \quad (5.55)$$

L'algorithme proposé est ainsi asymptotiquement non biaisé.

■

5.4.3 Analyse de la variance

Une analyse de la variance de l'estimation est proposée dans ce paragraphe. Cette analyse est réalisée pour une estimation obtenue via la proportion P_C . Des résultats similaires peuvent être obtenus avec les autres proportions R_C et Q_C .

Pour faciliter l'analyse de la variance, il est choisi d'exprimer le seuil en fonction de l'écart type σ_y . On définit donc un nouveau seuil tel que $\bar{C} = \frac{C}{\sigma_y}$ et la variance σ_y est supposée connue. On peut alors exprimer z_t de la manière suivante :

$$z(t) = \begin{cases} 1 & \text{if } \frac{y(t)}{\sigma_y} \geq \bar{C} \\ 0 & \text{if } \frac{y(t)}{\sigma_y} < \bar{C} \end{cases} \quad (5.56)$$

On définit la corrélation normalisée par :

$$\overline{\rho}_x(i) = \frac{\rho_x(i)}{\sigma_x^2} \quad (5.57)$$

Théorème :

Si on considère les hypothèses présentées dans la section 5.2, l'algorithme présenté dans le tableau 5.2 est tel que :

- La distribution asymptotique de l'estimation de l'autocorrélation $\widehat{\rho}_z(i)$ est une distribution normale $\mathcal{N}(\rho_z(i), \frac{1}{N} \Sigma^z)$ où Σ^z est la matrice $(n+1) \times (n+1)$ avec les éléments $\sigma_{i,j}^z$ définie par :

$$\sigma_{i,j}^z = \sum_{h=-\infty}^{\infty} (\mathcal{E} \{z(t)z(t-i)z(t+h)z(t+h-j)\} - \rho_z(i)\rho_z(j)) \quad (5.58)$$

- La distribution asymptotique de l'estimation de l'autocorrélation $\widehat{\rho}_y(i)$ est une distribution normale $\mathcal{N}(\overline{\rho}_y(i), \frac{1}{N} \Sigma^y)$ où Σ^y est la matrice $(n+1) \times (n+1)$ avec les éléments $\sigma_{i,j}^y$ définie par :

$$\sigma_{i,j}^y = \left(P_C^{-1'}(\rho_z(i)) \right) \sigma_{i,j}^z \left(P_C^{-1'}(\rho_z(j)) \right) \quad (5.59)$$

avec :

$$P_C^{-1'}(x) = \frac{dP_C^{-1}(x)}{dx} \quad (5.60)$$

- La distribution asymptotique du vecteur de paramètres $\hat{\theta}$ est une distribution normale $\mathcal{N}\left(\theta, \frac{1}{N} \Sigma^\theta\right)$ avec Σ^θ est la matrice $n \times n$ définie par :

$$\Sigma^\theta = \sum_{i,j} \left(\frac{\partial \theta}{\partial \rho_y(i)} \right) \sigma_{i,j}^y \left(P \frac{\partial \theta}{\partial \rho_y(j)} \right)^T \quad (5.61)$$

■

Démonstration :

L'utilisation d'un seuil normalisé par rapport à l'écart-type permet de simplifier l'équation (5.28) de la fonction de distribution du couple $(y(t); y(t-i))$ si $y(t)$ est centré de la manière suivante :

$$\psi(y(t), y(t-i)) = \frac{1}{2\pi\sigma_y^2 \sqrt{1 - (\overline{\rho_y(i)})^2}} e^{-\frac{\frac{y(t)^2}{\sigma_y^2} + \frac{y(t-i)^2}{\sigma_y^2} - 2\overline{\rho_y(i)} \frac{y(t)y(t-i)}{\sigma_y^2}}{2(1 - (\overline{\rho_y(i)})^2)}} \quad (5.62)$$

On peut alors exprimer $P_C(i, \overline{\rho_y(i)})$ par :

$$P_C(i, \overline{\rho_y(i)}) = \int_{\overline{C}}^{+\infty} \int_{\overline{C}}^{+\infty} \psi\left(\frac{y(t)}{\sigma_y}, \frac{y(t-i)}{\sigma_y}\right) d\frac{y(t)}{\sigma_y} d\frac{y(t-i)}{\sigma_y} \quad (5.63)$$

qui peut être simplifiée par :

$$P_C(i, \overline{\rho_y(i)}) = \frac{1}{2\pi\sqrt{1 - (\overline{\rho_y(i)})^2}} \int_{\overline{C}}^{+\infty} \int_{\overline{C}}^{+\infty} \psi_1(y(t), y(t-i)) dy(t) dy(t-i) \quad (5.64)$$

avec

$$\psi_1(y(t), y(t-i)) = e^{-\frac{y(t)^2 + y(t-i)^2 - 2\overline{\rho_y(i)}y(t)y(t-i)}{2(1 - (\overline{\rho_y(i)})^2)}} \quad (5.65)$$

Dans la section précédente, il a été montré que l'estimation de l'autocorrélation $\widehat{\rho}_z(i)$ est une estimation asymptotiquement non biaisée de l'autocorrélation $\rho_z(i)$. On peut définir l'estimation de la covariance de $\widehat{\rho}_z(i)$ par :

$$Cov_z(i, j) = \mathcal{E}\{(\widehat{\rho}_z(i) - \rho_z(i))(\widehat{\rho}_z(j) - \rho_z(j))\} \quad (5.66)$$

À partir de l'équation (5.4), on obtient :

$$\begin{aligned} Cov_z(i, j) &= \frac{1}{(N-i)(N-j)} \mathcal{E}\left\{\left(\sum_{t=i+1}^N (z(t)z(t-i)) - \rho_z(i)\right)\left(\sum_{t'=j+1}^N (z(t')z(t'-j)) - \rho_z(j)\right)\right\} \\ &= \frac{1}{(N-i)(N-j)} \sum_{t=i+1}^N \sum_{t'=j+1}^N (\mathcal{E}\{z(t)z(t-i)z(t')z(t'-j)\} - \rho_z(i)\rho_z(j)) \end{aligned} \quad (5.67)$$

Et si on définit $h = t' - t$, on obtient :

$$Cov_z(i, j) = \frac{1}{(N-i)(N-j)} \sum_{t=i+1}^N \sum_{h=j+1-t}^{N-t} (\mathcal{E}\{z(t)z(t-i)z(t+h)z(t+h-j)\} - \rho_z(i)\rho_z(j)) \quad (5.68)$$

D'après DANDAWATE et GIANNAKIS, 1993, on a asymptotiquement :

$$\lim_{N \rightarrow \infty} \text{Cov}_z(i, j) = \sum_{h=-\infty}^{\infty} (\mathcal{E} \{z(t)z(t-i)z(t+h)z(t+h-j)\} - \rho_z(i)\rho_z(j)) \quad (5.69)$$

La distribution asymptotique de l'estimation de l'autocorrélation $\widehat{\rho}_z(i)$ est ainsi une distribution normale $\mathcal{N}(\rho_z(i), \frac{1}{N} \Sigma^z)$ où Σ^z est la matrice $(n+1) \times (n+1)$ dont les éléments $\sigma_{i,j}^z$ sont définis par :

$$\sigma_{i,j}^z = \sum_{h=-\infty}^{\infty} (\mathcal{E} \{z(t)z(t-i)z(t+h)z(t+h-j)\} - \rho_z(i)\rho_z(j)) \quad (5.70)$$

La valeur $\mathcal{E} \{z(t)z(t-i)z(t+h)z(t+h-j)\}$ représente la proportion $= \mathcal{P}_r \{ \frac{y(t)}{\sigma_y} > \bar{C}, \frac{y(t-i)}{\sigma_y} > \bar{C}, \frac{y(t+h)}{\sigma_y} > \bar{C}, \frac{y(t+h-j)}{\sigma_y} > \bar{C} \}$. Elle peut donc être exprimée par une quadruple intégrale comme dans la fonction (5.64).

Il est ainsi possible d'exprimer la variance $\sigma_{i,j}^z$ en fonction de $\overline{\rho_y(i)}$, $\overline{\rho_y(j)}$, $\overline{\rho_y(h)}$ et $\overline{\rho_y(h-j)}$. On peut ainsi en déduire la distribution asymptotique de l'estimation de l'autocorrélation $\widehat{\rho}_y(i)$ est une distribution normale $\mathcal{N}(\overline{\rho_y(i)}, \frac{1}{N} \Sigma^y)$ avec Σ^y la matrice $(n+1) \times (n+1)$ avec les éléments $\sigma_{i,j}^y$ définis par :

$$\sigma_{i,j}^y = \left(P_C^{-1'}(\rho_z(i)) \right) \sigma_{i,j}^z \left(P_C^{-1'}(\rho_z(j)) \right) \quad (5.71)$$

avec :

$$P_C^{-1'}(x) = \frac{dP_C^{-1}(x)}{dx} \quad (5.72)$$

Étant donné que le vecteur de paramètre θ est estimé à partir de l'estimation de l'autocorrélation $\widehat{\rho}_y(i)$, on montre que la distribution asymptotique du vecteur de paramètres $\hat{\theta}$ est une distribution normale $\mathcal{N}(\theta, \frac{1}{N} \Sigma^\theta)$ avec Σ^θ est la matrice $n \times n$ définie par :

$$\Sigma^\theta = \sum_{i,j} \left(\frac{\partial \theta}{\partial \rho_y(i)} \right) \sigma_{i,j}^y \left(\frac{\partial \theta}{\partial \rho_y(j)} \right)^T \quad (5.73)$$

■

il apparaît, à partir de ce théorème, que la variance asymptotique du vecteur de paramètres $\hat{\theta}$ dépend principalement du nombre de données N , du vecteur de paramètres θ et

du seuil C . On remarque qu'il est possible de réduire la variance en augmentant le nombre de données N .

Pour ce qui est de l'impact du seuil C , on montre que, pour un seuil C et un vecteur de paramètre θ donnés, on a :

$$P_C^{-1'}(\rho_z(i)) = \frac{1}{P_C'(\overline{\rho}_y(i))} \quad (5.74)$$

avec

$$P_C'(x) = \frac{dP_C(x)}{dx} \quad (5.75)$$

et

$$P_C'(\overline{\rho}_y(i)) = \frac{\overline{\rho}_y(i)}{2\pi(1-\overline{\rho}_y(i)^2)^{3/2}} \int_{\overline{C}}^{+\infty} \int_{\overline{C}}^{+\infty} \psi_1(y(t), y(t-i)) dy(t) dy(t-i) \\ + \frac{\overline{\rho}_y(i)}{2\pi(1-\overline{\rho}_y(i)^2)^{5/2}} \int_{\overline{C}}^{+\infty} \int_{\overline{C}}^{+\infty} \phi_1(y(t), y(t-i)) dy(t) dy(t-i) \quad (5.76)$$

avec

$$\phi_1(y(t), y(t-i)) = ((1 + \overline{\rho}_y(i)^2)y(t)y(t-i) - \overline{\rho}_y(i)(y(t)^2 + y(t-i)^2)) \psi_1(y(t), y(t-i)) \quad (5.77)$$

On peut alors tracer la figure 5.2 qui représente $P_C'(\overline{\rho}_y(i))$ en fonction de $\overline{\rho}_y(i)$ pour différentes valeurs de \overline{C} . On remarque sur cette figure que la variance asymptotique $\sigma_{i,j}^y$ est plus élevée si la valeur absolue du seuil C augmente. Ceci s'explique intuitivement par le fait que si le seuil C est trop élevé, il n'y a pas assez de points dans la proportion $P_C(i)$ pour estimer correctement l'autocorrélation $\rho_y(i)$. Ce résultat d'analyse est confirmé par des résultats numériques dans la section 5.4.4

5.4.4 Résultats numériques

Dans cette section, un exemple numérique est présenté pour illustrer le comportement de l'algorithme et évaluer ses performances. Les données sont générées suivant les équations (5.1) et (5.2) avec le vecteur de paramètres :

$$\theta = \begin{pmatrix} -0.5297 \\ -0.2685 \\ 0.1113 \end{pmatrix} \quad (5.78)$$

et avec $e(t)$ un bruit gaussien centré de variance unitaire

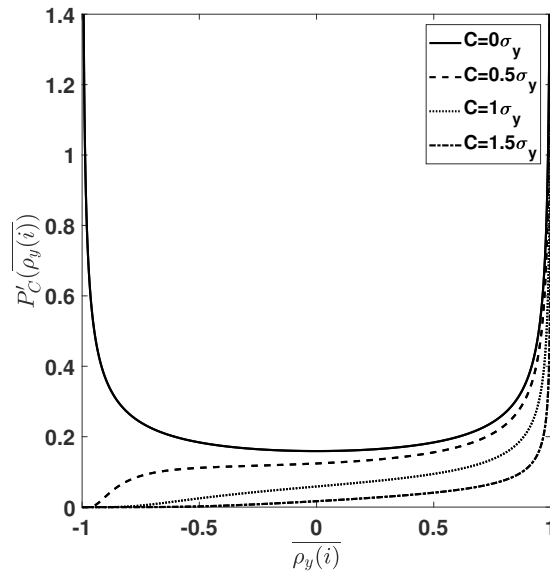


FIGURE 5.2 – $P'_C(\bar{\rho}_y(i))$ en fonction de $\bar{\rho}_y(i)$ pour les seuils $C = 0\sigma_y$, $C = 0.5\sigma_y$, $C = 1\sigma_y$ et $C = 1.5\sigma_y$

Le première simulation consiste à mesurer l'influence du seuil C sur l'algorithme. La méthode de Monte Carlo est utilisée avec 1000 simulations. Pour mesurer les performances de l'algorithme, on utilise la norme du vecteur d'erreur de paramètres définie par :

$$E = \|\theta - \hat{\theta}\|_2 \tag{5.79}$$

Dans un premier temps, l'algorithme avec un seuil $C = 0$ est comparé avec l'algorithme avec le seuil $C \neq 0$. La figure 5.3 illustre l'erreur de l'estimation des paramètres du modèle AR en fonction du seuil C de ces deux algorithmes pour $N = 10000$. Ceci correspond à une étude "expérimental", en simulation, du biais de l'estimation du vecteur de paramètre θ . Cette figure illustre le fait que l'algorithme avec le seuil $C \neq 0$ est plus performant que l'algorithme avec un seuil $C = 0$ jusqu'à un seuil limite où les performances se dégradent rapidement. Cette dégradation est due à la difficulté d'estimer les différentes proportions $P_C(\rho_y(i))$, $R_C(\rho_y(i))$ et $Q_C(\rho_y(i))$ pour un seuil trop élevé.

Dans la figure 5.4, l'erreur en fonction du seuil est affichée pour $N = 1000$, $N = 5000$,

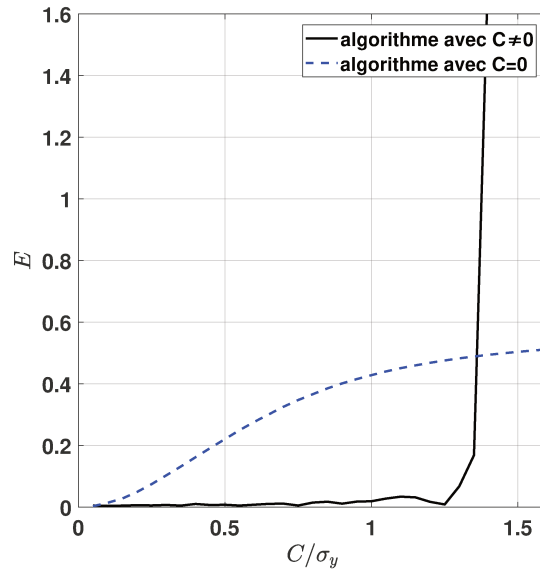


FIGURE 5.3 – Erreur de l'estimation des paramètres du modèle AR en fonction du seuil C de l'algorithme avec le seuil $C = 0$ et l'algorithme avec le seuil $C \neq 0$ pour $N = 10000$

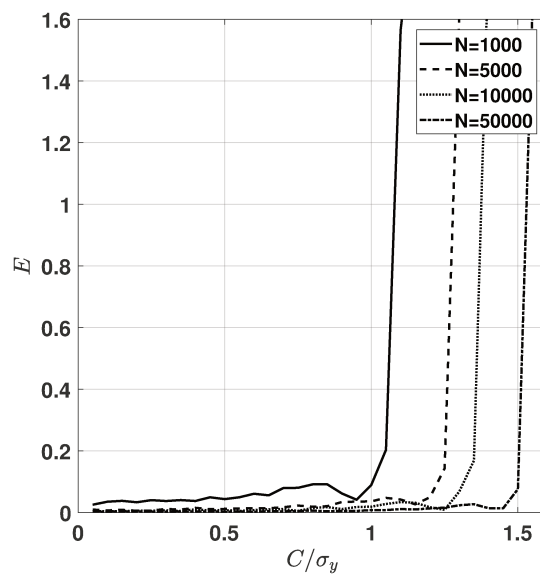


FIGURE 5.4 – Erreur de l'estimation des paramètres du modèle AR en fonction du seuil C avec un nombre de données $N=1000$, $N=5000$, $N=10000$ et $N=50000$

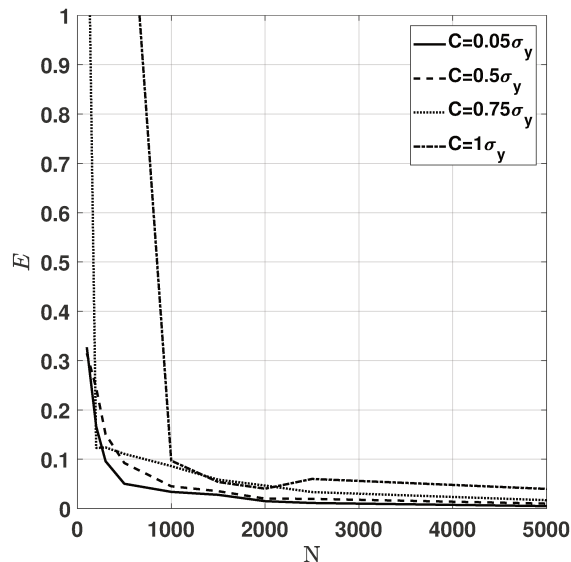


FIGURE 5.5 – Erreur de l'estimation des paramètres du modèle AR en fonction du nombre de données N pour les seuils $C = 0.05\sigma_y$, $C = 0.5\sigma_y$, $C = 0.75\sigma_y$ and $C = \sigma_y$

$N = 10000$ et $N = 50000$. On remarque que l'augmentation du nombre de points permet d'augmenter le seuil limite pour lequel les performances se dégradent. Ce résultat montre qu'un plus grand nombre de points permet de mieux estimer les différentes proportions $P_C(\rho_y(i))$, $R_C(\rho_y(i))$ et $Q_C(\rho_y(i))$ quand le seuil est élevé.

Une autre simulation illustre l'influence de N sur les performances de l'algorithme. La méthode de Monte Carlo est utilisée avec 1000 simulations pour chacun des seuils $C = 0.05\sigma_y$, $C = 0.5\sigma_y$, $C = 0.75\sigma_y$ et $C = \sigma_y$. La figure 5.5 affiche l'erreur en fonction du nombre de points N pour différents seuils. Cette figure confirme le fait qu'un nombre de points plus important permet à l'algorithme de donner une estimation plus précise des paramètres du modèle AR.

Dans la section 5.4.1, plusieurs solutions sont proposées pour l'estimation de l'autocorrélation $\rho_y(i)$. Il est possible de prendre en compte uniquement les proportions P_C , Q_C ou R_C avec les équation (5.40), (5.41) et (5.42) ou utiliser à la fois toutes ces proportions à partir de la matrice M en utilisant l'équation (5.45). Dans la figure 5.6, l'erreur en fonction du seuil est affichée en utilisant la proportion P_C , Q_C ou la matrice M en utilisant

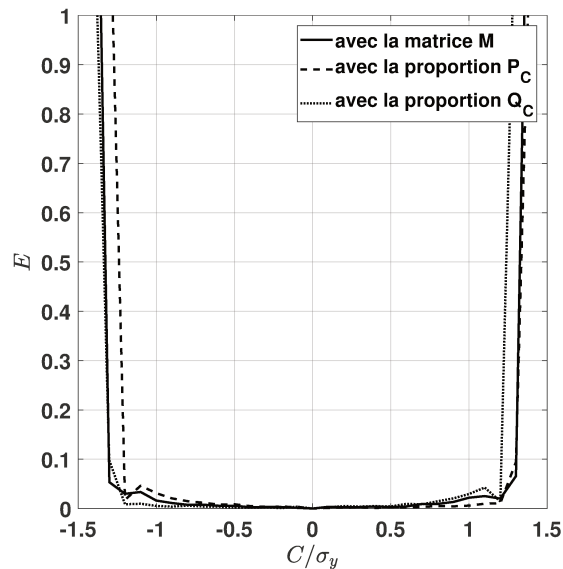


FIGURE 5.6 – Erreur de l'estimation des paramètres du modèle AR en fonction du seuil C avec l'algorithme utilisant la proportion P_C , la Proportion Q_C et la matrice M

la méthode de Monte Carlo avec 1000 simulations. On remarque que si on utilise qu'une seule proportion, on obtient une courbe non symétrique : la valeur absolue du seuil limite où les performances se détériorent est plus élevée pour les seuils positifs que les seuils négatifs pour la proportions P_C . On obtient l'inverse avec la proportion Q_C . L'utilisation de la matrice M permet d'avoir une meilleure estimation du modèle peu importe le signe du seuil C .

Sur la figure 5.7, les valeurs expérimentales et théoriques de $\sum_{i \in [1, n]} \sigma_{\hat{\theta}(i)}^2 N$ en fonction du seuil \bar{C} pour $N = 10000$ et $N = 40000$ sont affichées. Cette simulation confirme le résultat de la variance présenté dans la section 5.4.3. On peut noter que la variance est plus faible pour un seuil $\frac{C}{\sigma_y}$ proche de zéro. Cependant la valeur minimum n'est pas obtenue pour un seuil $\frac{C}{\sigma_y} = 0$ et on observe la présence d'une dissymétrie par rapport au seuil $\frac{C}{\sigma_y} = 0$. cette dissymétrie est induite par l'utilisation de la proportion P_C .

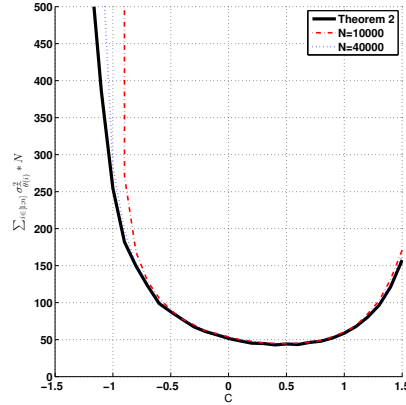


FIGURE 5.7 – La valeur expérimentale de $\sum_{i \in [1, n]} \sigma_{\hat{\theta}(i)}^2 + N$ en fonction du seuil C pour $N = 10000$ et $N = 40000$

5.5 Estimation d'un modèle AR en temps réel

L'algorithme décrit dans la section 5.4 ne permet pas d'estimer les paramètres du modèle AR en temps réel. Dans cette partie, il est présenté une nouvelle formulation de l'algorithme d'estimation pouvant être utilisée en temps réel.

5.5.1 Algorithme temps réel proposé

La première étape est de réaliser une estimation récursive de l'autocorrélation $\rho_z(i)$ pour $i \in [1; n]$. Cette estimation est réalisée avec la méthode du gradient de la manière suivante :

$$\hat{\rho}_z(i)_t = \hat{\rho}_z(i)_{t-1} + \mu_1 (z(t)z(t-i) - \hat{\rho}_z(i)_{t-1}) \quad (5.80)$$

où μ_1 est le pas d'adaptation et $\hat{\rho}_z(i)_t$ est l'estimation de $\rho_z(i)$ l'instant t .

La seconde étape consiste à réaliser une estimation de l'autocorrélation $\rho_y(i)$. La solution 5.40 ne peut pas être utilisée en temps réel et il n'existe pas de solution analytique pour déterminer $\rho_y(i)$ à partir de $\rho_z(i)$ pour un seuil C différent de zéro. Il est proposé d'utiliser une approximation de $P_C^{-1}(\cdot)$ noté $\widehat{P}_C^{-1}(\cdot)$. Cette approximation peut être obtenue par exemple en utilisant un réseaux de neurones, une table, ou une fonction polynomiale et permet d'estimer l'autocorrélation $\rho_y(i)$ de la manière suivante :

Algorithme en temps réel

— Initialisation des variables $n, \mu_1, \mu_2, \hat{\theta}_0, \hat{\rho}_z(i)_0$ pour $i \in [1, n]$

— Estimation de l'autocorrélation $\hat{\rho}_z(i)$ pour $i \in [1, n]$:

$$\hat{\rho}_z(i)_t = \hat{\rho}_z(i)_{t-1} + \mu_1 (z(t)z(t-i) - \hat{\rho}_z(i)_{t-1}) \quad (5.83)$$

— Estimation de l'autocorrélation $\hat{\rho}_y(i)$ pour $i \in [1, n]$:

$$\hat{\rho}_y(i)_t = \widehat{P_C^{-1}} (\hat{\rho}_z(i)_t) \quad (5.84)$$

— Estimation du vecteur de paramètre $\hat{\theta}$

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \mu_2 (N(\hat{\rho}_{y_t}) - M(\hat{\rho}_{y_t})\hat{\theta}_{t-1}) \quad (5.85)$$

TABLE 5.3 – Résumé de l'algorithme en temps réel

$$\hat{\rho}_y(i)_t = \widehat{P_C^{-1}} (\hat{\rho}_z(i)_t) \quad (5.81)$$

A partir de $\hat{\rho}_y(i)_t$, il est possible de faire une estimation récursive du vecteur des paramètres θ à l'aide la méthode du gradient de la manière suivante :

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \mu_2 (N(\hat{\rho}_{y_t}) - M(\hat{\rho}_{y_t})\hat{\theta}_{t-1}) \quad (5.82)$$

où μ_2 est le gain d'adaptation et $\hat{\theta}_t$ est l'estimation de θ à l'instant t . L'algorithme est résumé dans le tableau 5.3.

5.5.2 Résultats de convergence

Théorème :

Si on considère les hypothèses présentées dans la section 5.2, si $\widehat{P_C^{-1}} = P_C^{-1}$ et si μ_1 et μ_2 respectent les conditions suivante :

$$0 < \mu_1 < 2 \quad (5.86)$$

$$0 < \mu_2 < \frac{2}{\lambda_{max}} \quad (5.87)$$

où λ_{max} est la plus grande valeur propre de $M(\widehat{\rho}_y(i)_t)$, alors l'algorithme présenté dans le tableau 5.3 est tel que :

$$\lim_{t \rightarrow \infty} \mathcal{E}\{\tilde{\theta}_t\} = 0 \quad (5.88)$$

■

Démonstration :

Le produit $z(t)z(t-i)$ est défini par :

$$z_t z_{t-i} = \rho_z(i) + v_t(i) \quad (5.89)$$

avec $\mathcal{E}\{v_t(i)\} = 0$. On définit l'erreur de d'estimation $\tilde{\rho}_z(i)_t$ par :

$$\tilde{\rho}_z(i)_t = \rho_z(i) - \widehat{\rho}_z(i)_t \quad (5.90)$$

On peut donc déduire que :

$$\tilde{\rho}_z(i)_t = (1 - \mu_1)\tilde{\rho}_z(i)_{t-1} - \mu_1 v_t(i) \quad (5.91)$$

$$\mathcal{E}\{\tilde{\rho}_z(i)_t\} = (1 - \mu_1)\mathcal{E}\{\tilde{\rho}_z(i)_{t-1}\} - \mu_1 \mathcal{E}\{v_t(i)\} \quad (5.92)$$

$$\mathcal{E}\{\tilde{\rho}_z(i)_t\} = (1 - \mu_1)\mathcal{E}\{\tilde{\rho}_z(i)_{t-1}\} \quad (5.93)$$

Finalement, si $0 < \mu_1 < 2$, l'espérance de l'erreur d'estimation de l'autocorrélation $\tilde{\rho}_z(i)$ tend vers 0 :

$$\lim_{t \rightarrow \infty} \mathcal{E}\{\tilde{\rho}_z(i)_t\} = 0 \quad (5.94)$$

et donc l'estimation de $\rho_z(i)$ converge en moyenne :

$$\lim_{t \rightarrow \infty} \mathcal{E}\{\widehat{\rho}_z(i)_t\} = \rho_z(i) \quad (5.95)$$

Si $\widehat{P}_C^{-1} = P_C^{-1}(\cdot)$, alors on a $\lim_{t \rightarrow \infty} \mathcal{E}\{\widehat{\rho}_y(i)_t\} = \rho_y(i)$ car $P_C^{-1}(\cdot)$ est continue.

On définit l'erreur d'estimation des paramètres $\tilde{\theta}_t$ par :

$$\tilde{\theta}_t = \theta - \hat{\theta}_t \quad (5.96)$$

$$\tilde{\theta}_t = (I_{p+1} - \mu_2 M(\hat{\rho}_{y_t})) + \mu_2 (N(\hat{\rho}_{y_t}) - M(\hat{\rho}_{y_t})\theta) \quad (5.97)$$

Comme il a été montré que $\lim_{t \rightarrow \infty} \mathcal{E}\{\tilde{\rho}_y(i)_t\} = 0$, alors on obtient :

$$\lim_{t \rightarrow \infty} \mathcal{E}\{(N(\hat{\rho}_{y_t}) - M(\hat{\rho}_{y_t})\theta)\} = 0 \quad (5.98)$$

Finalement, la convergence de l'espérance $\mathcal{E}\{\tilde{\theta}_t\}$ dépend de $M(\hat{\rho}_{y_t})$. Si $0 < \mu_2 < \frac{2}{\lambda_{max}}$, alors :

$$\lim_{t \rightarrow \infty} \mathcal{E}\{\tilde{\theta}_t\} = 0 \quad (5.99)$$

où λ_{max} est la plus grande valeur propre de $M(\hat{\rho}_{y_t})$

■

Pour réduire le temps de convergence de l'algorithme, il est possible de faire varier les pas μ_1 et μ_2 . Une décroissance exponentielle est utilisée pour accélérer la convergence au début puis réduire la variance à la fin. On définit $\mu(t)$ par :

$$\mu(t) = (\mu^{ini} - \mu^{fin}) e^{-t/\tau} + \mu^{fin} \quad (5.100)$$

avec τ est une constante qui définit la décroissance exponentielle, μ^{ini} est la valeur initiale du pas d'adaptation et μ^{fin} est valeur finale du pas.

Pour éviter une mauvaise initialisation, il est possible d'utiliser un opérateur de projection pour contraindre $\hat{\theta}_t$ dans un ensemble convexe comme utilisé dans GUO et ZHAO, 2013 et GUO et al., 2015. L'équation 5.82 est modifiée par :

$$\hat{\theta}_t = \Pi_{\Omega} \left(\hat{\theta}_{t-1} + \mu_2 \left(N(\hat{\rho}_{y_t}) - M(\hat{\rho}_{y_t})\hat{\theta}_{t-1} \right) \right) \quad (5.101)$$

avec, pour un ensemble convexe Ω , $\Pi_{\Omega}(\cdot)$ est un opérateur de projection défini par :

$$\Pi_{\Omega}(x) = \underset{\rho_y(i)}{\operatorname{argmin}} \left\{ |x - w| \right\} \quad (5.102)$$

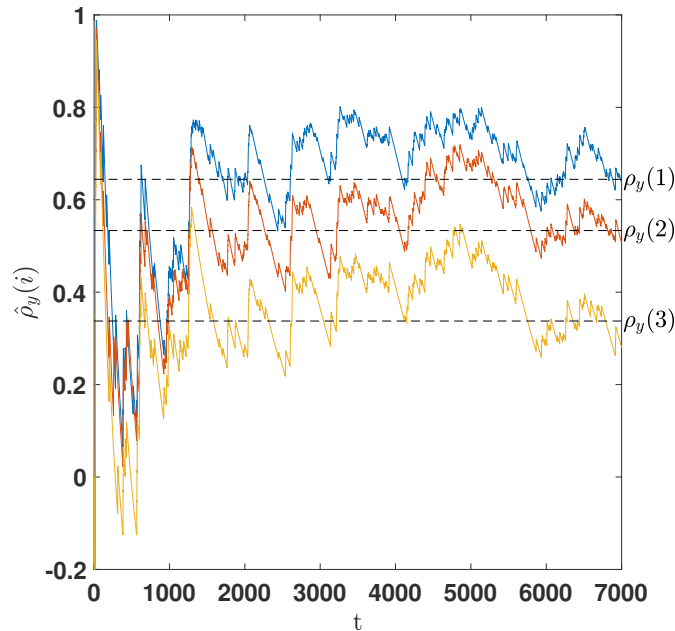


FIGURE 5.8 – évolution de $\hat{\rho}_y(i)_t$ en fonction de t

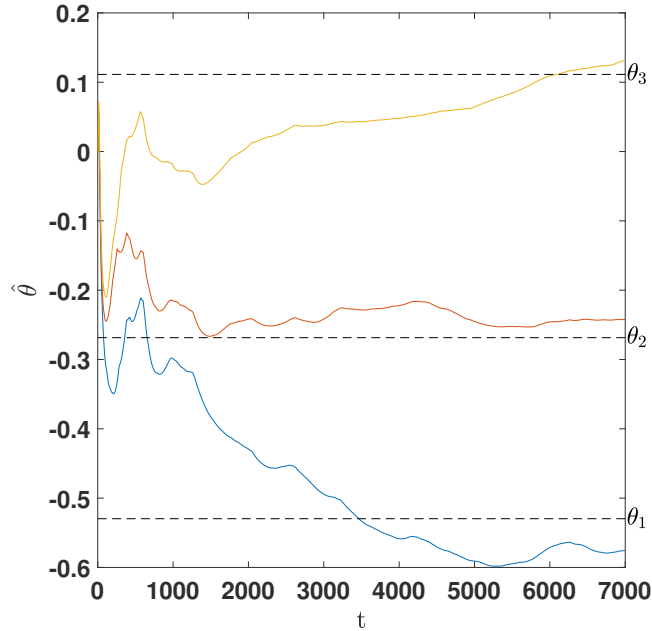
5.5.3 Résultats numériques

Dans cette section, un exemple numérique illustre le comportement de l'algorithme temp réel. Les données sont générées suivant les équations (5.1) et (5.2) avec le vecteur de paramètres (5.78). Le seuil C est fixé à $1\sigma_y$, μ^{ini} à 0.01, μ^{fin} à 0.001 et τ à 500. La figure 5.8 montre l'évolution de l'estimation de l'autocorrélation $\hat{\rho}_y(i)_t$ en fonction du temps et la figure 5.9 montre l'évolution de l'estimation du vecteur de paramètres $\hat{\theta}_t$ en fonction du temps. Ces résultats montrent que l'algorithme en temps réel peut être utilisé pour l'estimation des paramètres du modèle AR.

5.6 Extension de l'algorithme pour des données quantifiées

5.6.1 Formulation du problème

Dans certain cas, des données quantifiées sont à disposition et il s'agit de proposer ici une extension de l'algorithme de la section 5.4 pour prendre en compte l'utilisation de plusieurs seuils. Considérons une nouvelle série temporelle $\{z_t\}$ définie par :

FIGURE 5.9 – Évolution de $\hat{\theta}_t$ en fonction de t

$$\begin{cases} y(t) = \phi_1 y(t-1) + \dots + \phi_p y(t-n) + e(t) \\ z(t) = Q_q(y(t)) \end{cases} \quad (5.103)$$

avec $Q_q(\cdot)$ défini par :

$$Q_q(y(t)) = \begin{cases} q & \text{if } y(t) \geq C_q \\ q-1 & \text{if } y(t) \geq C_{q-1} \text{ et } y(t) < C_q \\ \vdots & \\ 1 & \text{if } y(t) \geq C_1 \text{ et } y(t) < C_2 \\ 0 & \text{if } y(t) < C_1 \end{cases} \quad (5.104)$$

où les C_j sont q seuils constants pour $j \in \{1; \dots; q\}$ et $e(t)$ est un bruit blanc centré de variance finie.

L'objectif est d'estimer les paramètres du modèle AR en utilisant les équations de Yule Walker (5.7). Pour cela, il faut d'abord déterminer l'estimation de l'autocorrélation $\rho_y(i)$ comme dans le cas des données binaires. Deux approches sont possibles pour l'estimation

de l'autocorrélation $\rho_y(i)$ et vont être présentées dans la suite.

5.6.2 Première approche

La première approche consiste à utiliser l'algorithme présenté dans le paragraphe 5.4 pour chaque seuil C_j pour avoir q estimations de l'autocorrélation $\rho_y(i)$. Pour chaque seuil C_j , on définit :

$$P_{C_j}(\rho_y(i)) = \int_{C_j}^{+\infty} \int_{C_j}^{+\infty} \psi(y(t), y(t-i)) dy(t)dy(t-i) \quad (5.105)$$

$$\begin{aligned} R_{C_j}(\rho_y(i)) &= \int_{-\infty}^{C_j} \int_{C_j}^{+\infty} \psi(y(t), y(t-i)) dy(t)dy(t-i) \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^C \psi(y(t), y(t-i)) dy(t)dy(t-i) \end{aligned} \quad (5.106)$$

$$Q_{C_j}(\rho_y(i)) = \int_{-\infty}^{C_j} \int_{-\infty}^{C_j} \psi(y(t), y(t-i)) dy(t)dy(t-i) \quad (5.107)$$

Avec ces différentes proportions, on peut alors définir la matrice $\widehat{\mathcal{M}}_j(\rho_y(i))$ par :

$$\widehat{\mathcal{M}}_j(\rho_y(i)) = \begin{pmatrix} R_{C_j}(\rho_y(i)) & P_{C_j}(\rho_y(i)) \\ Q_{C_j}(\rho_y(i)) & R_{C_j}(\rho_y(i)) \end{pmatrix} \quad (5.108)$$

L'estimation de l'autocorrélation $\rho_y(i)$ est alors obtenue par la moyenne des estimations de l'autocorrélation $\rho_y(i)$ de chaque seuil :

$$\widehat{\rho}_y(i) = \frac{1}{q} \sum_{j=1}^q \operatorname{argmin}_{\rho_y(i)} \left\{ \left\| \mathcal{M}_j(i) - \widehat{\mathcal{M}}_j(\rho_y(i)) \right\|_F \right\} \quad (5.109)$$

5.6.3 Seconde approche

Pour la seconde approche, les différents seuils ne sont pas traités indépendamment les uns des autres. De nouvelles proportions sont définies par :

$$\begin{aligned} P_{i,k}(i) &= \mathcal{P}_r\{y(t) \geq C_i, y(t) < C_{i+1}, y(t-i) \geq C_k, y(t-i) < C_{k+1}\} \\ &= \mathcal{P}_r\{z(t) = i, z(t-i) = k\} \end{aligned} \quad (5.110)$$

avec $C_0 = -\infty, C_{q+1} = +\infty, i = k = 0, \dots, q$. Une représentation graphique des différentes proportions est présentée dans la figure 5.10 dans le cas de l'utilisation de deux seuils.

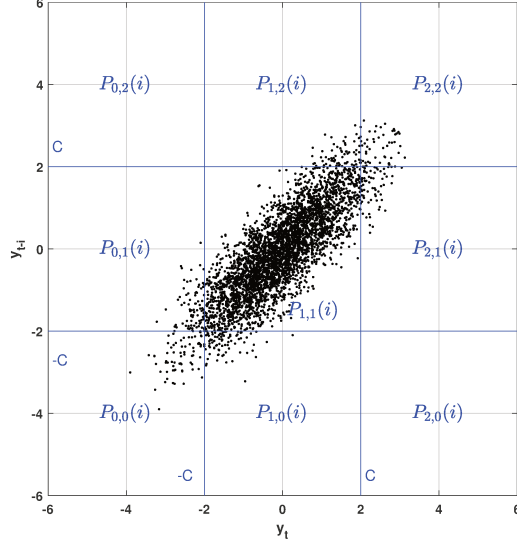


FIGURE 5.10 – Représentation graphique des différentes proportions $P_{j,k}(i)$ dans le cas de l'utilisation de deux seuils $C_1 = -2$ et $C_2 = 2$

Comme dans le cas binaire, on définit la matrice $\mathcal{M}(i)$ de dimension $(q+1) \times (q+1)$ avec les différentes proportions $P_{i,k}(i)$ par :

$$\mathcal{M}(i) = \begin{pmatrix} P_{0,q}(i) & \cdots & P_{q,q}(i) \\ \vdots & & \vdots \\ P_{0,0}(i) & \cdots & P_{q,0}(i) \end{pmatrix} \quad (5.111)$$

Les différentes proportions $P_{i,k}(i)$ sont définies aussi à partir de la fonction de distribution $\psi(y_t, y_{t-i})$ par :

$$P_{j,k}(\rho_y(i)) = \int_{C_j}^{C_{j+1}} \int_{C_k}^{C_{k+1}} \psi(y(t), y(t-i)) dy(t) dy(t-i) \quad (5.112)$$

On peut alors définir la matrice $\widehat{\mathcal{M}}(i, \rho_y(i))$ par :

$$\widehat{\mathcal{M}}(\rho_y(i)) = \begin{pmatrix} P_{0,q}(\rho_y(i)) & \cdots & P_{q,q}(\rho_y(i)) \\ \vdots & & \vdots \\ P_{0,0}(\rho_y(i)) & \cdots & P_{q,0}(\rho_y(i)) \end{pmatrix} \quad (5.113)$$

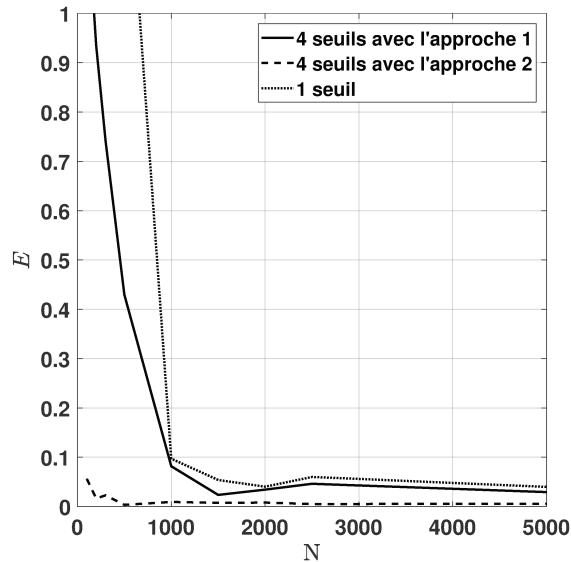


FIGURE 5.11 – Moyenne des erreurs de l'estimation des paramètres du modèle AR en fonction du nombre de données N avec l'algorithme avec un seuil et l'algorithme avec quatre seuils

L'estimation de l'autocorrélation $\rho_y(i)$ est alors obtenue par :

$$\hat{\rho}_y(i) = \underset{\rho_y(i)}{\operatorname{argmin}} \left\{ \left\| \mathcal{M}(i) - \widehat{\mathcal{M}}(\rho_y(i)) \right\|_F \right\} \quad (5.114)$$

5.6.4 Résultats numériques

Dans cette section, on compare les résultats des deux approches avec plusieurs seuils présentés dans la section précédente avec l'algorithme avec un seuil de la section 5.4. Les seuils utilisés sont $C_1 = -1$, $C_2 = -0.5$, $C_3 = -0.5$ et $C_4 = -1$ pour les algorithmes avec plusieurs seuils et un seuil $C = 1$ pour l'algorithme avec un seul seuil.

La figure 5.11 affiche l'erreur en fonction du nombre de points N pour les différents algorithmes en utilisant la méthode de Monte Carlo sur 1000 simulations. On observe que l'utilisation de plusieurs seuils permet d'améliorer la précision de l'algorithme surtout pour un nombre de points N faible. L'utilisation de plusieurs seuils peut compenser la dégradation des performances de l'algorithme avec un seul seuil pour un nombre d'échantillons faible.

Si on compare les deux approches multi seuils, on remarque que la deuxième approche donne de meilleurs résultats que la première approche. Cette observation peut être expliquée par le fait que les points sont mieux distribués dans les différentes proportions de la deuxième approche.

5.7 Utilisation d'une série temporelle non linéaire

5.7.1 Formulation du problème

Nous considérons dans cette section le cas d'une série temporelle un peu différente de la précédente. Cette structure particulière se justifie par le fait que certaines séries temporelles peuvent ne pas avoir une distribution gaussienne comme dans la section 5.2 mais une distribution demi gaussienne. L'objectif est ici le même que précédemment : estimer les paramètres d'un modèle AR. Dans cette section, nous considérons $\{z(t)\}$ de la forme suivante :

$$\begin{cases} y(t) = \phi_1 y(t-1) + \dots + \phi_p y(t-p) + e(t) \\ z(t) = Q(|y(t)|) \end{cases} \quad (5.115)$$

Le seuil C est supposé ici strictement positif et $\{y(t)\}$ suit une loi normale centrée.

5.7.2 Algorithme proposé

L'algorithme proposé ici suit les mêmes étapes que l'algorithme proposé dans la section 5.4.1. Tout d'abord, il est à noter que l'autocorrélation $\rho_z(i)$ est telle que :

$$\begin{aligned} \rho_z(i) &= \mathcal{P}_r\{y(t) \geq C, y(t-i) \geq C\} \\ &\quad + \mathcal{P}_r\{y(t) < -C, y(t-i) \geq C\} \\ &\quad + \mathcal{P}_r\{y(t) \geq C, y(t-i) < -C\} \\ &\quad + \mathcal{P}_r\{y(t) < -C, y(t-i) < -C\} \end{aligned} \quad (5.116)$$

$\rho_z(i)$ peut s'exprimer en fonction de l'autocorrélation $\rho_y(i)$ via la proportion $T_C(\rho_y(i))$ suivante :

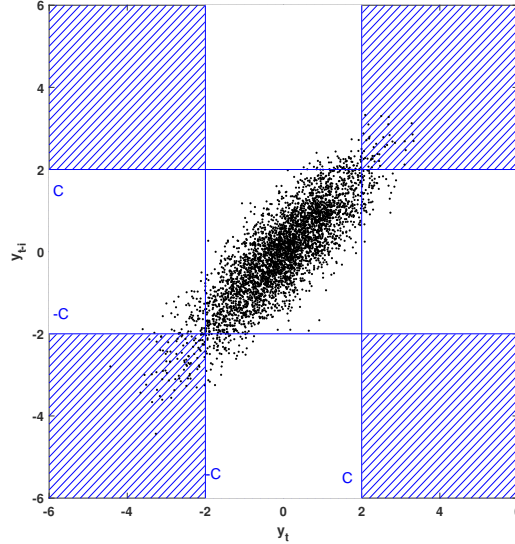


FIGURE 5.12 – Représentation de la proportion $T_C(\rho_y(i))$ (représentée en hachuré)

$$\begin{aligned}
 T_C(\rho_y(i)) &= \int_C^{+\infty} \int_C^{+\infty} \psi(y(t), y(t-i)) d(y(t))dy(t-i) \\
 &+ \int_C^{+\infty} \int_{-\infty}^{-C} \psi(y(t), y(t-i)) d(y(t))dy(t-i) \\
 &+ \int_{-\infty}^{-C} \int_C^{+\infty} \psi(y(t), y(t-i)) d(y(t))dy(t-i) \\
 &+ \int_{-\infty}^{-C} \int_{-\infty}^{-C} \psi(y(t), y(t-i)) d(y(t))dy(t-i) \quad (5.117)
 \end{aligned}$$

Cette proportion est illustrée dans la figure 5.12 et correspond à l'autocorrélation $\rho_z(i)$:

$$\rho_z(i) = T_C(\rho_y(i)) \quad (5.118)$$

$T_C(\rho_y(i))$ est une fonction monotone strictement croissante pour une autocorrélation $\rho_y(i) \in [0; \infty[$. On peut alors définir une fonction $T_C^{-1}(\cdot)$ telle que :

$$T_C^{-1}(T_C(x)) = x \text{ pour } x \geq 0 \quad (5.119)$$

Il est ainsi possible de faire une estimation de $\rho_y(i)$, pour $\rho_y(i) \geq 0$ en minimisant la fonction suivante :

$$\hat{\rho}_y(i) = \underset{\widehat{\rho}_y(i)}{\operatorname{argmin}} \left\{ |\rho_z(i) - T_C(\rho_y(i))| \right\} \quad (5.120)$$

Une fois les autocorrélations $\rho_y(i)$ estimées pour $i \in [1, \dots, p]$, il ne reste plus qu'à utiliser les équations de Yule Walker (5.7) pour obtenir une estimation du vecteur de paramètre $\hat{\theta}$.

Remarque : il est possible d'étendre ce résultat à plusieurs niveaux de quantification.

5.8 Application de l'identification d'un modèle AR sur des données réelles binaires d'un accéléromètre

5.8.1 Contexte

Dans cette section, les algorithmes de ce chapitre sont appliqués sur des données réelles d'accéléromètres binarisées. Cependant, les algorithmes ne peuvent pas être appliqués directement sur les données brutes fournies par l'accéléromètre. La moyenne de la série temporelle $\{y(t)\}$ est supposée nulle. Il est donc nécessaire de supprimer la composante continue de l'accélération avant de binariser les données de l'accéléromètre.

Le but de cette section est de montrer que l'algorithme avec un seuil $C = 0$ décrit dans le tableau 5.1 et l'algorithme avec un seuil $C \neq 0$ décrit dans le tableau 5.2 sont utilisables dans le cadre réel et de définir l'impact de différents paramètres de l'algorithme comme le degré p du modèle AR, le seuil C et la taille de la fenêtre T_w . Cette section n'a pas pour objectif de déterminer l'algorithme d'apprentissage et les paramètres optimaux dans le cadre de la reconnaissance d'activités avec des données binaires. Ce problème sera traité dans le prochain paragraphe 6.

Nous proposons de différencier les trois activités suivantes :

- La marche.
- La course.
- Le saut.

Les données utilisées pour l'apprentissage supervisé sont collectées directement à partir d'eTact sur un individu. La fréquence d'échantillonnage utilisée est fixée à 50 Hz. On dispose de la série temporelle binaire $\{z_t\}$ définie par le modèle 5.1.

5.8.2 Algorithme proposé

La série temporelle $\{y_t\}$ correspond à la norme des données des trois axes fournies par l'accéléromètre. la gravité est ensuite soustraite à la norme pour avoir un signal de moyenne nulle. Un fenêtrage de longueur T_w est appliqué sur la série temporelle binaire $\{z_t\}$ avec un recouvrement de 80%. Pour chaque fenêtre, l'algorithme avec un seuil $C = 0$ décrit dans le tableau 5.1 et l'algorithme avec un seuil $C \neq 0$ décrit dans le tableau 5.2 sont appliqués pour estimer les paramètres du modèle AR du signal $\{y_t\}$. Seuls les paramètres du modèle AR sont utilisés pour la classification.

L'apprentissage supervisé des K plus proches voisins avec $K = 5$ est ensuite utilisé pour classer les différentes fenêtres parmi les activités. L'algorithme est constitué de deux étapes. La première étape est la phase d'apprentissage qui consiste à apprendre les fonctions de décisions. La seconde partie consiste à appliquer la fonction de décision sur des nouvelles données pour déterminer la justesse de l'apprentissage. Une validation croisée est employée pour pouvoir utiliser toutes les données disponibles de la base d'apprentissage : la base d'apprentissage est découpée en quatre parties de taille égale et à tour de rôle, une partie est utilisée pour la validation et les trois autres parties sont utilisées pour l'apprentissage. Les performances de l'algorithme sont mesurées à l'aide de la justesse. Pour rappel, cette métrique correspond au nombre d'activités correctement classées sur le nombre total d'activités et est définie par :

$$acc = \frac{VP + FP}{VP + FP + VN + FN} \quad (5.121)$$

5.8.3 Résultats numériques

Dans un premier temps, les performances de l'algorithme avec un seuil $C = 0$ sont étudiées. La figure 5.13 expose la justesse de la classification en fonction de la taille de fenêtre T_w pour le degré du modèle AR $p = 2$, $p = 5$ et $p = 8$. On remarque sur cette figure que plus la taille de la fenêtre T_w est élevée, plus la précision de l'algorithme de classification augmente. Cela peut s'expliquer par le fait que l'estimation de l'autocorrélation du signal binarisé $\hat{\rho}_z(i)$ est plus précise avec un nombre de points importants. On observe aussi que la classification est plus précise avec un ordre du modèle AR élevé.

La justesse de la classification à partir des paramètres calculés avec l'algorithme avec un seuil $C = 0$ en fonction du degré p du modèle AR pour une taille de fenêtre $T_w = 2s$,

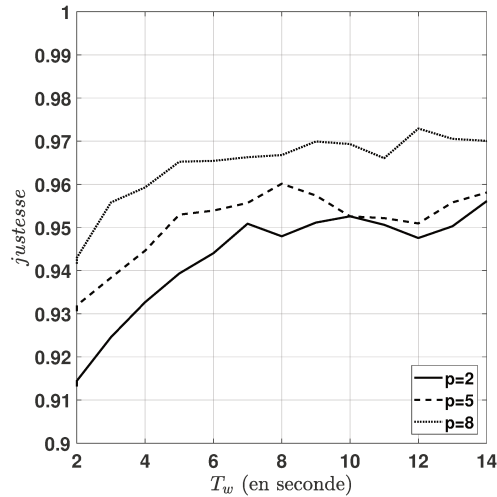


FIGURE 5.13 – La justesse de la classification à partir des paramètre calculée avec l’algorithme avec un seuil $C = 0$ en fonction de la taille de fenêtre T_w pour le degré du modèle AR $p = 2$, $p = 5$ et $p = 8$.

$T_w = 5s$ et $T_w = 8s$ est affichée sur la figure 5.14. Cette figure confirme le fait que plus le degré du modèle AR est important, plus la classification est correcte.

Pour la suite, les performances de l’algorithme avec un seuil $C \neq 0$ sont étudiées pour un degré du modèle AR $p = 10$. La figure 5.15 affiche la justesse de la classification en fonction de la taille de fenêtre T_w pour un seuil $C = 0,4g$, $C = 0,8g$ et $C = 1,2g$. On observe sur cette figure qu’une fenêtre d’au moins de 6 secondes permet d’avoir les meilleurs résultats de classification.

La justesse de la classification à partir des paramètres calculés avec l’algorithme avec un seuil $C \neq 0$ en fonction du seuil C et une taille de fenêtre $T_w = 2s$, $T_w = 7s$ et $T_w = 12s$ est affichée sur la figure 5.16. On observe sur cette figure que le seuil optimal pour l’algorithme est situé entre $1,2g$ et $1,6g$. On remarque que les résultats de la classification se dégrade énormément à partir de $1,6g$. La cause de cette dégradation est sûrement due à la difficulté d’estimer la proportions $P_C(i, \rho_y(i))$ pour un seuil trop élevé comme observé dans la section 5.4.4.

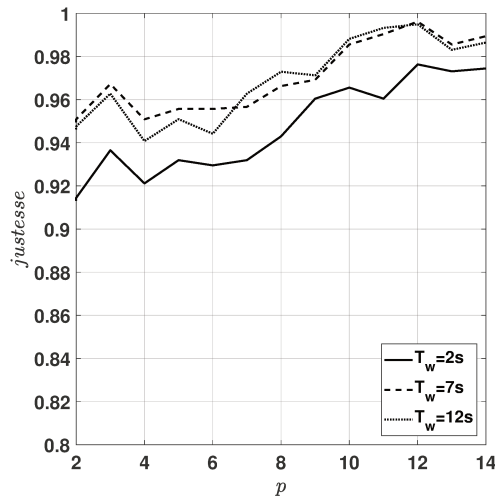


FIGURE 5.14 – La justesse de la classification à partir des paramètre calculé avec l’algorithme avec un seuil $C = 0$ en fonction du degré du modèle AR p pour une taille de fenêtre $T_w = 2s$, $T_w = 5s$ et $T_w = 8s$

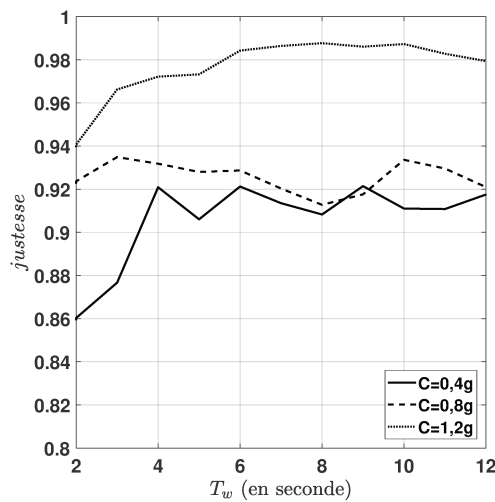


FIGURE 5.15 – La justesse de la classification à partir des paramètre calculé avec l’algorithme avec un seuil $C \neq 0$ en fonction de la taille de fenêtre T_w pour un degré du modèle AR $p = 10$ et pour un seuil $C = 0,4g$, $C = 0,8g$ et $C = 0,12g$.

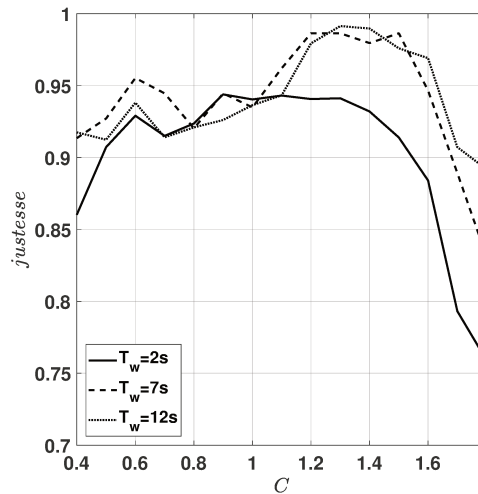


FIGURE 5.16 – La justesse de la classification à partir des paramètre calculé avec l’algorithme avec un seuil $C \neq 0$ en fonction du seuil C pour un degré du modèle AR $p = 10$ et une taille de fenêtre $T_w = 2s$, $T_w = 7s$ et $T_w = 12s$

5.9 Conclusion

Dans ce chapitre, plusieurs algorithmes sont présentés pour identifier un modèle AR à partir de données binaires. Il existe dans la littérature une solution pour l’estimation d’un modèle AR dans le cas où le signal est binarisé avec un seuil nul. Cet algorithme est présenté au début de ce chapitre. Un second algorithme est proposé pour l’estimation d’un modèle AR dans le cas où le signal est binarisé avec un seuil différent de zéro. Une analyse de la convergence et de la variance de cette algorithme est proposée. Des résultats numériques sont ensuite présentés pour montrer l’efficacité de cet algorithme et ses limites.

Plusieurs extensions de cet algorithme sont proposées. Dans un premier temps, l’algorithme est adapté pour une utilisation en temps réel. La convergence de cet algorithme est analysée et testée avec des données de simulation. L’algorithme a été ensuite étendu dans le cas de l’utilisation de données quantifiées. Cet algorithme permet de réduire l’erreur sur l’estimation des paramètres du modèle AR par rapport à l’utilisation d’un seul seuil avec un même nombre d’échantillons. Pour continuer, une version appropriée de cet algorithme est présentée dans le cas d’une série temporelle non linéaire.

Pour finir, les algorithmes pour identifier un modèle AR à partir de données binaires pour un seuil nul et différent de zéro sont appliqués sur des données réelles d'accéléromètres fournies par eTact. Le but de cette simulation est de déterminer s'il est possible de faire de la reconnaissance d'activités (trois activités différentes) à partir des paramètres du modèle AR obtenu à l'aide de ces algorithmes et de déterminer l'impact du seuil ainsi que du degré du modèle AR. Les résultats de cette simulation montrent qu'il est possible d'obtenir un algorithme performant pour la reconnaissance d'activités à partir des algorithmes d'identification d'un modèle AR via de données binaires.

Chapitre 6

Reconnaissance d'activités avec des données binaires

6.1 Introduction

Dans ce chapitre, l'approche 4 pour faire de la reconnaissance d'activités avec eTact est étudiée. La particularité de cette approche est de binariser les données de l'accéléromètre avant de les stocker, puis de les envoyer vers une plateforme externe. Ensuite, sur la plateforme externe, les données binarisées sont traitées pour faire de la reconnaissance d'activités. L'utilisation de données binaires a pour principal avantage de permettre le stockage d'une grande quantité de données et de limiter l'utilisation du BLE sur eTact. Cependant, la binarisation des données entraîne une grosse perte d'information sur le signal. Le but de ce chapitre est déterminer s'il est possible de faire de la reconnaissance d'activités avec des données binaires.

L'utilisation de données binaires ou quantifiées a un impact sur le choix des paramètres pour la reconnaissance d'activités. Par exemple, le maximum et le minimum qui sont deux paramètres souvent utilisés avec des signaux de haute résolution, ne peuvent être déterminés dans le cas où le signal est binaire. Cependant, il a été montré dans le chapitre 5 qu'il est possible de récupérer les paramètres du modèle AR d'un signal à partir des données binarisées de ce signal. Dans un premier temps, les paramètres utilisables dans le cas de l'utilisation de données binaires vont être détaillées. De plus, la binarisation implique le choix d'un seuil de quantification, l'impact du choix de ce seuil va être étudié dans ce chapitre.

La même procédure que le chapitre 4 est suivie. Dans un premier temps, un algorithme de réduction de dimension est utilisé pour réduire le nombre de paramètres pour la classification. Ensuite différents algorithmes de classification supervisées sont comparés pour obtenir une solution admissible pour faire de la reconnaissance d'activités à partir de données binaires. Les résultats sont ensuite comparés avec les approches 1, 2 et 3 étudiées dans le chapitre 4.

6.2 Choix des paramètres

Pour limiter le nombre de données stockées et envoyées par etact, seules les données binaires de la norme sont utilisées. Sur eTact, la récupération des données binaires suit la procédure suivante :

- Récupération de l'accélération suivant les trois axes de l'accéléromètre.
- Calcul de la norme.
- Soustraction de $1g$ à la norme pour centrer le signal autour de 0.
- Binarisation de la norme suivant le seuil C_b (1 si la norme $\geq C_b$, sinon 0).

L'utilisation de données binaires limite le choix des paramètres. L'ensemble *Ensemble_Param_C* comprend les paramètres suivants :

- La moyenne et la variance du signal binaire.
- Le ZCM du signal binaire.
- Le nombre de fois où le signal est positif ou négatif.
- Le maximum, le minimum et la médiane du temps entre les changements d'état du signal binaire.
- La moyenne et la variance du temps entre les changements d'état du signal binaire.
- Les paramètres du modèle AR de la norme en utilisant l'algorithme du tableau 5.2. Le degré p du modèle AR est fixé à 12.
- Les paramètres du modèle AR du signal binarisé. Le degré p du modèle AR est fixé à 12.
- L'autocorrélation du signal binarisé. Cette estimation est obtenue à l'aide de l'équation (5.4).
- L'estimation de l'autocorrélation de la norme. L'autocorrélation est estimée à partir de l'équation (5.40).

- Les coefficients de la TFD du signal binaire. Les coefficients sont obtenus à partir de l'estimation de l'autocorrélation de la norme.

Au total, *Ensemble_Param_C* comporte 40 paramètres auxquelles il faut ajouter le nombre de paramètres de la TFD qui varie selon le nombre d'échantillons de la fenêtre de temps. Le nombre total de paramètres est obtenu par le calcul suivant :

$$nb_{param} = 40 + 25 * E \left(\frac{f * T_w}{50} \right) \quad (6.1)$$

avec f la fréquence d'échantillonnage de l'accéléromètre, $E(x)$ est la partie entière de x et T_w la longueur de la fenêtre de temps en secondes.

6.2.1 Normalisation

Tout comme le chapitre 4, on normalise les paramètres pour améliorer les résultats de la classification. La normalisation min-max est appliquée à l'ensemble de paramètres pour que tous les paramètres soient compris entre 0 et 1. Pour rappel, la normalisation est calculée de la manière suivante :

$$p_{norm_i} = \frac{p_i - \min(p_i)}{\max(p_i) - \min(p_i)} \quad (6.2)$$

où p_i est le paramètre d'index i et p_{norm_i} est le paramètre normalisé entre 0 et 1

6.3 Réduction de dimension de l'ensemble de paramètres

Le même algorithme de réduction de dimension de l'ensemble de paramètres *Ensemble_Param_B* décrit dans 4.3 est utilisé pour l'ensemble *Ensemble_Param_C*. Comme la dimension de l'ensemble *Ensemble_Param_C* n'est pas trop importante, l'algorithme SFS est directement utilisé. Cependant, les paramètres ne sont pas calculés sur eTact comme pour l'ensemble de paramètres *Ensemble_Param_B*, ce qui nous permet d'utiliser un plus grand nombre de paramètres pour la classification comme pour l'approche 1. On fixe alors le nombre maximal de paramètres à 30.

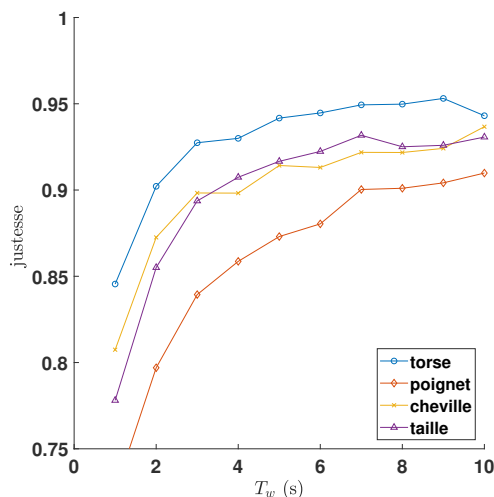


FIGURE 6.1 – Justesse de la classification (Forêts aléatoires) en fonction du temps T_w pour les différentes position du capteur avec $f = 50\text{Hz}$ et $C_b = 0$

6.4 Impact du seuil T_w

La figure 6.1 illustre le rôle de la taille T_w de la fenêtre de temps. La classification est réalisée dans les conditions suivantes : classification par l'algorithme des forêts aléatoires, le seuil $C_b = 0$ et $f = 50\text{Hz}$. On observe que plus la fenêtre est élevée, plus le résultat de la classification est de meilleure qualité. La taille de la fenêtre de temps est fixée à $T_w = 5\text{s}$ pour comparer les résultats de l'approche 4 avec les approches 1,2 et 3.

6.5 Impact du seuil C_b

La figure 6.2 illustre l'impact du choix du seuil de quantification C_b sur les performances de la classification. Cette classification est réalisée à partir de l'algorithme par d'un arbre de décision avec des données échantillonnées à 50 Hz. On remarque sur cette courbe que les performances varient selon le seuil utilisé. Une baisse des performances est observée pour le seuil $C_b = 0$. Le seuil $C_b = 0.15g$ est utilisé pour la suite car ce seuil permet d'avoir de meilleurs résultats pour les différentes positions. on peut remarquer que dans le cas où le capteur est fixé au torse, on obtient un meilleur résultat avec un seuil négatif.

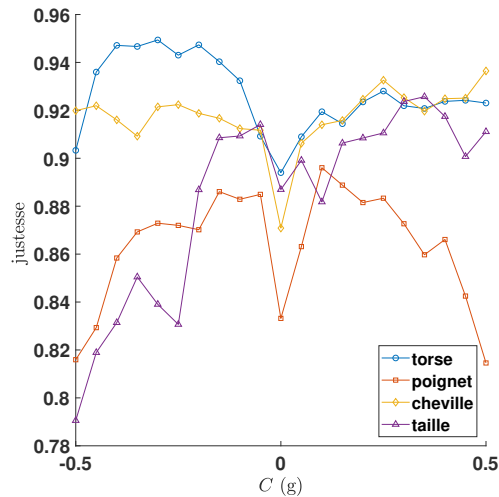


FIGURE 6.2 – Justesse de la classification (arbre de décision) en fonction du seuil C_b pour les différentes positions du capteur avec $f = 50\text{Hz}$

6.6 Comparaison des résultats de classification

6.6.1 Performance des algorithmes de classification

Le tableau 6.1 expose la justesse et le tableau 6.2 expose le F1score pour les différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_C*. On remarque, en général, que la reconnaissance d'activités est de moins bonne qualité que celles obtenue avec les ensembles *Ensemble_Param_A* et *Ensemble_Param_B*. L'utilisation de données binaires entraîne une détérioration de la performance de l'algorithme de reconnaissance d'activités. Les algorithmes de classification qui donnent les résultats les plus performants sont les K plus proches voisins et les forêts aléatoires. Ici encore, le réseau de neurones est celui qui donne les plus mauvais résultats.

La position de l'accéléromètre a un impact beaucoup plus important pour l'approche 4. On observe que la position la plus adéquate est le torse, puis la taille. Le poignet et la cheville sont les positions où l'algorithme de reconnaissance d'activités est le moins performant.

TABLE 6.1 – Justesse des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_C* avec $T_w = 5s$ et $C_b = 0.15g$

position	torse		poignet		cheville		taille	
	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz
LDA	0,881	0,892	0,850	0,849	0,911	0,909	0,875	0,884
QDA	0,895	0,905	0,865	0,857	0,926	0,924	0,874	0,892
Arbre de décision	0,932	0,935	0,889	0,890	0,917	0,919	0,898	0,904
K plus proches voisins	0,960	0,960	0,937	0,935	0,908	0,959	0,937	,941
Forêts aléatoires	0,957	0,954	0,924	0,926	0,945	0,950	0,9299	0,95
Naïve Bayésienne	0,921	0,928	0,895	0,893	0,927	0,936	0,901	0,905
Réseaux de neurones	0,841	0,814	0,769	0,769	0,845	0,822	0,811	0,823

TABLE 6.2 – F1score des différentes positions et des différents algorithmes de classification pour l'ensemble de paramètres *Ensemble_Param_C* avec $T_w = 5s$ et $C_b = 0.15g$

position	torse		poignet		cheville		taille	
	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz	25Hz	50Hz
LDA	0,862	0,875	0,842	0,839	0,892	0,896	0,854	0,869
QDA	0,867	0,872	0,856	0,821	0,872	0,891	0,871	0,883
Arbre de décision	0,896	0,903	0,846	0,850	0,8551	0,848	0,869	0,885
K plus proches voisins	0,945	0,949	0,919	0,919	0,809	0,933	0,918	,923
Forêts aléatoires	0,939	0,939	0,898	0,906	0,896	0,910	0,916	0,914
Naïve Bayésienne	0,918	0,914	0,8974	0,874	0,884	0,876	0,895	0,884
Réseaux de neurones	0,643	0,4958	0,520	0,520	0,526	0,540	0,614	0,584

TABLE 6.3 – Autonomie mémoire eTact en fonction des différentes approches

Approches	1	2	3	4
f=25Hz	11h38	87h15	4362h	186h08
f=50Hz	5h49	87h15	4362h	93h04

TABLE 6.4 – Autonomie de la batterie d'eTact en fonction des différentes approches

Approches	1	2	3	4
f=25Hz	84h	28h	28h	82h
f=50Hz	72h	36h	24h	68h

6.6.2 Autonomie d'eTact

Le tableau 6.3 affiche l'autonomie mémoire selon les différentes approches. La motivation principale de binariser les données de l'accéléromètre est l'autonomie mémoire d'eTact. Les données de l'accéléromètre sont codées sur 12 bits donc on a besoin de deux octets pour stocker les données avec l'approche 1. Pour l'approche 4, on peut stocker 8 données de l'accéléromètre sur un octet. L'approche 4 possède donc une autonomie mémoire 16 fois plus importante que l'approche 1. Comme pour l'approche 1, l'autonomie mémoire de l'approche 4 va varier selon la fréquence d'échantillonnage de l'accéléromètre. On observe que l'approche 3 reste l'approche avec le plus d'autonomie mémoire. Cependant, l'approche 4 a une autonomie mémoire plus importante que l'approche 2.

Le tableau 6.4 illustre l'autonomie de la batterie en fonction des différentes approches. On observe ici que l'approche 4 a une autonomie dans la même ordre de grandeur que l'approche 1. La légère baisse d'autonomie par rapport à l'approche 1 est due aux calculs de la norme et la binarisation des données. L'approche 4 a une autonomie batterie plus importante que l'approche 2 et 3.

L'ensemble de ces résultats sont reportés sur la figure 6.3 qui expose l'autonomie de la mémoire en fonction de l'autonomie de la batterie pour les différentes approches avec $f = 50\text{Hz}$.

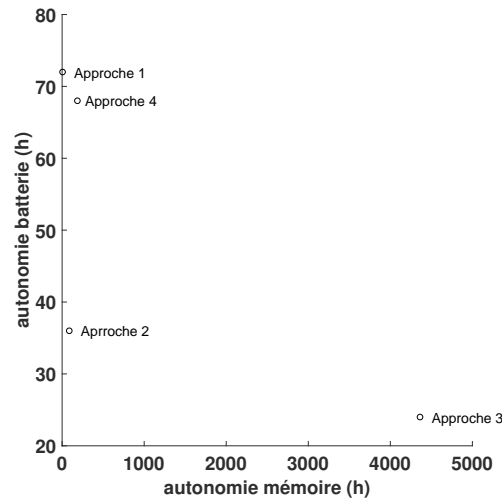


FIGURE 6.3 – L'autonomie de la mémoire en fonction de l'autonomie de la batterie pour les différentes approches avec $f = 50\text{Hz}$

6.7 Conclusion

Dans ce chapitre, une étude de l'approche 4 est proposée. La particularité de cette approche est de binariser les données de l'accéléromètre avant de les envoyer vers une plateforme externe. Les données binaires sont ensuite utilisées pour faire de la reconnaissance d'activités. L'utilisation de données binaires empêche de calculer de nombreux paramètres utilisés par les trois premières approches. Une liste des paramètres utilisables dans le cas de l'utilisation de données binaires est proposée au début de ce chapitre.

La mise en oeuvre de l'approche 4 implique le choix par l'utilisateur d'un paramètre de synthèse supplémentaire par rapport aux approches 1,2 et 3, à savoir la valeur du seuil utilisé pour binariser les données de l'accéléromètre. On observe qu'utiliser un seuil égal à zéro ne donne pas les meilleurs résultats.

Pour finir, une comparaison des résultats de l'approche 4 est faite avec les trois premières approches décrites dans le chapitre 4. On peut noter que l'approche 4 permet d'avoir une plus grande autonomie mémoire que l'approche 1 et 2, et l'autonomie batterie est plus importante que l'approche 2 et 3. Cependant, l'utilisation de données binaires entraîne une perte de qualité lors de la classification. On remarque aussi que la position

du capteur a une plus grande influence sur les résultats pour l'approche 4 que pour les approches 1,2 et 3.

Chapitre 7

Identification d'un modèle commutant linéaire à partir de données binaires

7.1 Introduction

Les résultats du chapitre 4 nous ont montré qu'il était possible de faire de l'identification d'un modèle à partir de données binarisées. Ces résultats encourageants nous ont mené à étudier d'autres systèmes et en particulier l'identification d'un système commutant linéaire. Un système commutant est composé d'un nombre fini de sous-systèmes qui ont chacun leur propre dynamique. A chaque instant, la sortie du système commutant est la sortie d'un seul des sous-systèmes. Le but de ce chapitre est de proposer un algorithme pour l'identification d'un système commutant linéaire à partir de données binaires. L'utilisation de données binaires peut être motivée par plusieurs raisons : un capteur avec une faible résolution moins onéreux, un capteur de haute résolution non disponible, la réduction de la consommation d'énergie du capteur ou la réduction du nombre de données à envoyer en utilisant des données binaires.

L'identification de systèmes commutants est un sujet qui suscite beaucoup d'intérêt dans la littérature récemment. Il existe plusieurs approches pour l'identification de systèmes commutants : Une approche algébrique ou géométrique (voir HASHAMBHOY et VIDAL, 2005, MA et VIDAL, 2005 et LE, LAUER et BLOCH, 2013), une approche basée sur l'optimisation (voir FERRARI-TRECATE et al., 2003, NAKADA, TAKABA et KATAYAMA, 2005 et LAUER, 2013), une approche bayésienne (voir JULOSKI, WEILAND et HEEMELS, 2005), une approche basée sur le partitionnement de données (voir FERRARI-TRECATE

et al., 2003, NAKADA, TAKABA et KATAYAMA, 2005 et LAUER, 2013) et une approche réursive (voir VIDAL, 2008, et BRESCHI, PIGA et BEMPORAD, 2016). Toutes ces méthodes utilisent des données de haute qualité et ne sont pas applicables pour l'utilisation de données binaires.

Un algorithme en deux étapes est proposé dans ce chapitre pour l'identification d'un système commutant linéaire. Dans un premier temps, un sous-ensemble défini par les vecteurs de paramètres des sous-systèmes à estimer est construit. Cette étape permet de transformer le problème initial d'identification en un problème d'identification de systèmes commutants linéaires en présence d'un bruit borné. Ce problème d'identification est ensuite résolu en utilisant les algorithmes présents dans la littérature (voir OZAY, LAGOA et SZNAIER, 2015, GOUDJIL et al., 2016b et GOUDJIL et al., 2016a). Les résultats proposés dans ce chapitre sont publiés dans AUBER et al., 2019a.

Dans la suite de ce chapitre, le problème d'identification est présenté dans la section 7.2. Ensuite, l'algorithme proposé pour l'identification d'un modèle commutant linéaire à partir de données binaires est exposé dans la section 7.3. Dans la section 7.4, l'algorithme est appliqué sur des exemples numériques. Pour finir, une conclusion est apportée à ce chapitre.

7.2 Formulation du problème

Considérons un système commutant linéaire composé d'un nombre fini de sous-systèmes à réponse impulsionnelle finie :

$$\begin{cases} y(t) = \varphi(t)^T \theta_{\lambda(t)} + e(t) \\ s(t) = Q(y(t)) \end{cases} \quad (7.1)$$

où :

— $\varphi(t) \in \mathbb{R}^{(n+1)}$ est le vecteur d'observation défini par :

$$\varphi(t) = (u(t) \ u(t-1), \dots, u(t-n))^T \quad (7.2)$$

avec $u(t)$ le signal d'entrée du système commutant à l'instant t et n est l'ordre du système.

— l'opérateur $Q(\cdot)$ modélise le capteur binaire de la manière suivante :

$$Q(y(t)) = \begin{cases} 1 & \text{if } y(t) \geq C \\ 0 & \text{if } y(t) < C \end{cases} \quad (7.3)$$

- $\lambda(t) \in \{1, \dots, n_s\}$ est l'état discret qui indique le sous-système actif à l'instant t , n_s est le nombre de sous-systèmes et $\theta_{\lambda(t)}$ et le vecteur de paramètres associé au sous-système $\lambda(t)$. A noter que $\lambda(t)$ est inconnue.
- $\{e(t)\}$ est un bruit additif.

L'objectif est d'estimer les différents paramètres caractérisant les sous-systèmes $\{\theta_i\}_{i=1}^{n_s}$ à partir de la connaissance de $\{u(t)\}$ et $\{s(t)\}$.

Les hypothèses suivantes sont considérées :

Hypothèse 1 : $\|\theta_i\|_2 = 1$ pour chaque sous-système ($\|\cdot\|_2$ est la norme l_2).

Hypothèse 2 : Le signal d'entrée $\{u(t)\}$ est borné : il existe M tel que $|u(t)| \leq M$.

Remarque : L'hypothèse 1 est une hypothèse standard dans l'identification à partir de données binaires (voir JAFARI, JUILLARD et ROGER, 2012, BOURGOIS et JUILLARD, 2012 ou BOTTEGAL, HJALMARSSON et PILLONETTO, 2017). Elle garantit une solution unique à l'identification. Il est impossible de distinguer $\theta_{\lambda(k)}$ de $\gamma\theta_{\lambda(k)}$ avec $\gamma \neq 0$ sans cette hypothèse.

7.3 Algorithme d'identification

7.3.1 Principe de l'algorithme

La présentation du fonctionnement de l'algorithme nécessite plusieurs définitions. Pour la suite, on considère le bruit $e(t) = 0$. Le comportement de l'algorithme avec un bruit non nul sera traité dans la section 7.4 avec une simulation de Monte Carlo.

On définit les ensembles $\mathcal{O} \subset \mathbb{R}^{n+1}$, \mathcal{O}_i^+ et \mathcal{O}_i^- for $i \in \{1, \dots, n_s\}$ tels que :

$$\mathcal{O} = \{\varphi / \|\varphi\|_1 \leq M\} \quad (7.4)$$

où $\|\cdot\|_1$ est la norme 1.

$$\mathcal{O}_i^+ = \left\{ \varphi \in \mathcal{O} / \varphi^T \theta_i > 0 \right\} \quad (7.5)$$

$$\mathcal{O}_i^- = \left\{ \varphi \in \mathcal{O} / \varphi^T \theta_i < 0 \right\} \quad (7.6)$$

\mathcal{O}_i^+ correspond à l'ensemble possible des vecteurs d'observations φ tels que $\varphi^T \theta_i > 0$.

Le comportement de certains sous-systèmes peut être masqué par d'autres sous-systèmes. La définition suivante est donnée pour indiquer la notion de co-discernabilité pour n_s sous-systèmes composant le systèmes commutants et la notion de discernabilité pour un sous-système parmi n_s sous-systèmes co-discernables.

Définition 1 : n_s sous systèmes commutants sont co-discernables si :

$$\bigcup_i \mathcal{O}_i^+ \neq \mathcal{O} \quad (7.7)$$

■

En d'autres termes, n_s sous-systèmes composant un système commutant ne sont pas co-discernables si il n'est pas possible de différencier le signe de la sortie pour tous les vecteurs d'observations φ . Cela signifie que la sortie du système peut être positive ou négative pour tout vecteur d'observation φ suivant la valeur de l'état discret. Par exemple, si le système commutant est composé de deux sous-systèmes θ_1 et $\theta_2 = -\theta_1$, les deux sous-systèmes ne sont pas co-discernables. n_s sous-systèmes compose un système commutant sont co-discernables si il existe un vecteur d'observation φ tel que $\varphi^T \theta_i$ est positif pour n'importe quelle valeur de i .

Définition 2 : Le sous système i est discernable parmi n_s sous-systèmes si :

$$\mathcal{O}_i^+ \not\subseteq \bigcup_{j \neq i} \mathcal{O}_j^+ \quad (7.8)$$

■

Un sous-système i est discernable si il existe un vecteur d'observation φ tel que ce sous-système est le seul sous-système qui donne une sortie positive. Dans la figure 7.1,

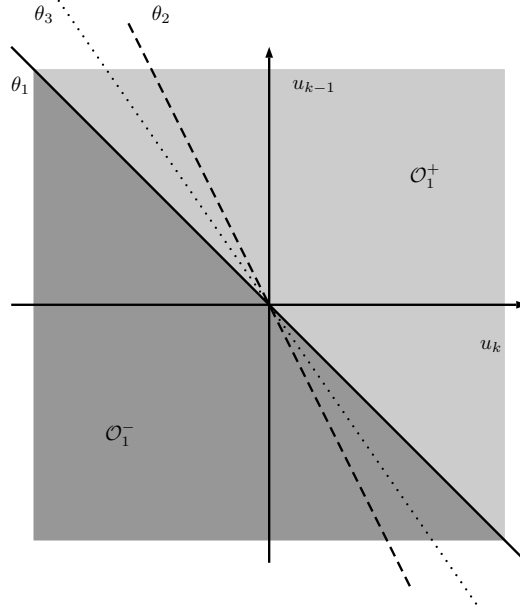


FIGURE 7.1 – Définition des ensemble \mathcal{O}_1^+ and \mathcal{O}_1^- avec trois sous-systèmes pour un exemple de deux dimensions ($n + 1 = 2$)

les sous-systèmes θ_1 , θ_2 et θ_3 sont co-discernables. Les sous-systèmes θ_1 et θ_2 sont discernables. Par contre, θ_3 n'est pas discernable car il est masqué par les sous-systèmes θ_1 et θ_2 .

Dans la suite, on suppose que tous les sous-systèmes sont discernables. On définit les ensembles suivants :

$$\mathcal{O}^+ = \bigcap_i \mathcal{O}_i^+ \quad (7.9)$$

$$\mathcal{O}^- = \bigcap_i \mathcal{O}_i^- \quad (7.10)$$

$$\mathcal{O}^{+/-} = \overline{\mathcal{O}^+ \cup \mathcal{O}^-} \quad (7.11)$$

Ces différents ensembles sont illustrés dans la figure 7.2. On remarque que \mathcal{O}^+ et \mathcal{O}^- sont des polyèdres convexes dans \mathcal{O} délimités par les hyperplans $\{\varphi \in \mathcal{O} / \varphi^T \theta_i = 0\}$, c'est à dire par des hyperplans paramétrés par les vecteurs de paramètres θ_i . Il parait donc que si il est possible d'isoler les vecteurs observations φ le long de ces hyperplans, alors il devrait être possible d'estimer les vecteurs de paramètres $\{\theta_i\}_{i=1}^{n_s}$.

On définit \mathcal{B}^+ comme la frontière de \mathcal{O}^+ et \mathcal{B}^- comme la frontière de \mathcal{O}^- .

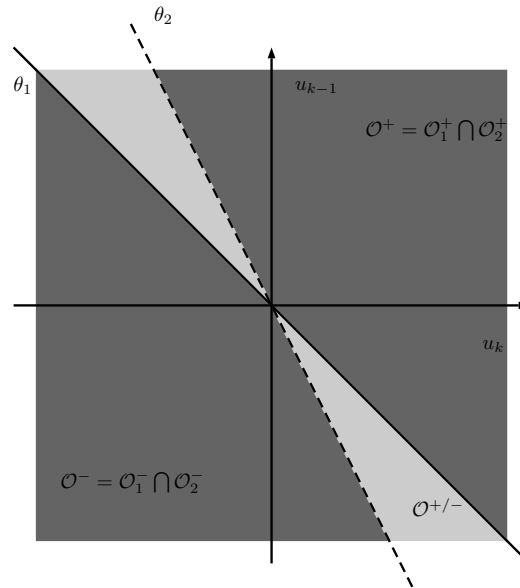


FIGURE 7.2 – Définition de $\mathcal{O}^+ = \bigcap_j \mathcal{O}_j^+$ et de $\mathcal{O}^- = \bigcap_j \mathcal{O}_j^-$ pour un exemple de deux dimensions ($n + 1 = 2$) et de deux sous-systèmes.

Pour finir , on définit \mathcal{B} de la manière suivante :

$$\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^- \quad (7.12)$$

Le principe de l'algorithme proposé est basé sur l'analyse de l'ensemble \mathcal{B} et suit les étapes suivantes :

- *Étape 1* : Estimation de l'ensemble \mathcal{B} en utilisant les mesures de $\{u(t)\}$ et de $\{s(t)\}$.
- *Étape 2* : Estimation des vecteurs de paramètre $\{\theta_i\}_{i=1}^{n_s}$ à partir de l'ensemble \mathcal{B} .

7.3.2 Description de l'algorithme proposé

7.3.2.1 Estimation de l'ensemble \mathcal{B}

Conception de la séquence d'entrée

Dans un premier temps, une séquence d'entrée doit être choisie. Ce signal doit satisfaire la condition d'excitation persistante. On propose ici d'utiliser une séquence d'entrée spécifique pour faciliter l'estimation de l'ensemble \mathcal{B} . On propose d'utiliser la séquence d'entrée suivante :

$$u(t) = -M + q(t)\delta \quad (7.13)$$

avec

$$\delta = \frac{2M}{m-1} \quad (7.14)$$

où $q(k)$ est une variable aléatoire suivant une loi uniforme dans $\{0; 1; \dots; m-1\}$ et m un entier choisi par l'utilisateur.

Cette séquence d'excitation a deux propriétés importantes. La première propriété est qu'elle produit une excitation persistante à l'entrée du système grâce à la variable aléatoire $q(t)$. La deuxième propriété est qu'elle permet de parcourir de manière méthodique dans l'ensemble \mathcal{O} .

Estimation de l'ensemble \mathcal{B}

On définit l'ensemble $\mathcal{O}_N \subset \mathcal{O}$ de la manière suivante :

$$\mathcal{O}_N = \{\varphi(1); \varphi(2); \dots; \varphi(N)\} \quad (7.15)$$

\mathcal{O}_N correspond aux N observations disponibles du vecteur d'observations $\varphi(t)$. À partir cet ensemble, on définit les ensembles suivants :

- L'ensemble $\widehat{\mathcal{O}}_N^+$: Il correspond aux vecteurs d'observations $\varphi(t)$ dont la sortie est toujours positive ou nulle.

$$\widehat{\mathcal{O}}_N^+ = \{\varphi \in \mathcal{O}_N / \forall t \in [1; N] \text{ tel que } \varphi(t) = \varphi \text{ alors } s(t) = 1\} \quad (7.16)$$

- L'ensemble $\widehat{\mathcal{O}}_N^-$: Il correspond aux vecteurs d'observations $\varphi(t)$ dont la sortie est toujours négative.

$$\widehat{\mathcal{O}}_N^- = \{\varphi \in \mathcal{O}_N / \forall k \in [1; N] \text{ tel que } \varphi(t) = \varphi \text{ alors } s(t) = 0\} \quad (7.17)$$

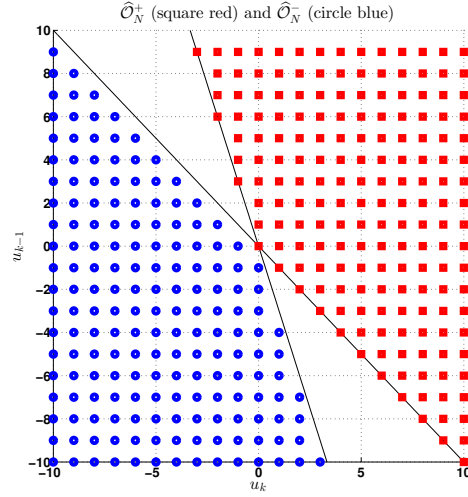


FIGURE 7.3 – Représentation graphique des ensembles $\hat{\mathcal{O}}_N^+$ et $\hat{\mathcal{O}}_N^-$ avec $\delta = 1$.

- L'ensemble $\hat{\mathcal{O}}_N^{+/-}$: Il correspond aux vecteurs d'observations $\varphi(t)$ dont la sortie est parfois négative et parfois positive ou nulle.

$$\hat{\mathcal{O}}_N^{+/-} = \left\{ \varphi \in \mathcal{O}_N / \varphi \notin \hat{\mathcal{O}}_N^+ \cup \hat{\mathcal{O}}_N^- \right\} \quad (7.18)$$

Les ensembles $\hat{\mathcal{O}}_N^+$, $\hat{\mathcal{O}}_N^-$ et $\hat{\mathcal{O}}_N^{+/-}$ sont illustrés dans la figure 7.3 et 7.4 avec un exemple numérique avec $M = 10$ et $m = 21$ (donc $\delta = 1$).

À partir des ensembles $\hat{\mathcal{O}}_N^+$, $\hat{\mathcal{O}}_N^-$ et $\hat{\mathcal{O}}_N^{+/-}$, on définit les ensembles suivants :

- L'ensemble $\hat{\mathcal{B}}_N^+$: il correspond aux vecteurs d'observations $\varphi(t)$ dans l'ensemble $\hat{\mathcal{O}}_N^+$ et à la frontière de $\hat{\mathcal{O}}_N^{+/-}$.

$$\hat{\mathcal{B}}_N^+ = \left\{ \varphi \in \mathcal{O}_N^+ / \exists t \in [1; N] \text{ tel que } \|\varphi(t) - \varphi\|_1 = \delta \text{ et } \varphi(t) \in \hat{\mathcal{O}}_N^{+/-} \right\} \quad (7.19)$$

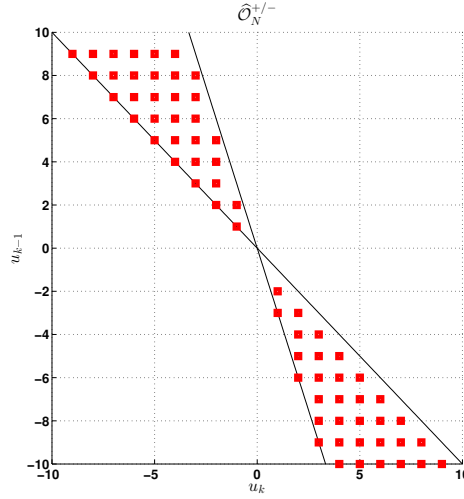


FIGURE 7.4 – Représentation graphique de l'ensemble $\hat{\mathcal{O}}_N^{+/-}$ avec $\delta = 1$.

- L'ensemble $\hat{\mathcal{B}}_N^-$: il correspond aux vecteurs d'observations $\varphi(t)$ dans l'ensemble $\hat{\mathcal{O}}_N^{+/-}$ et à la frontière de $\hat{\mathcal{O}}_N^-$.

$$\hat{\mathcal{B}}_N^- = \left\{ \varphi \in \mathcal{O}_N^{+/-} / \exists t \in [1; N] \text{ tel que } \|\varphi(t) - \varphi\|_1 = \delta \text{ et } \varphi(t) \in \hat{\mathcal{O}}_N^- \right\} \quad (7.20)$$

- L'ensemble $\hat{\mathcal{B}}_N^{+/-}$: il correspond à l'union des ensembles $\hat{\mathcal{B}}_N^+$ et $\hat{\mathcal{B}}_N^-$.

$$\hat{\mathcal{B}}_N = \hat{\mathcal{B}}_N^+ \cup \hat{\mathcal{B}}_N^- \quad (7.21)$$

Les différents ensembles $\hat{\mathcal{B}}_N^+$, $\hat{\mathcal{B}}_N^-$ et $\hat{\mathcal{B}}_N$ sont illustrés dans la figure 7.5.

$\hat{\mathcal{B}}_N$ est une estimation de \mathcal{B} et suit le théorème suivant :

Théorème :

Considérons l'estimation $\hat{\mathcal{B}}_N$ ci-dessus, si la série temporelle $\{q_t\}$ est uniformément

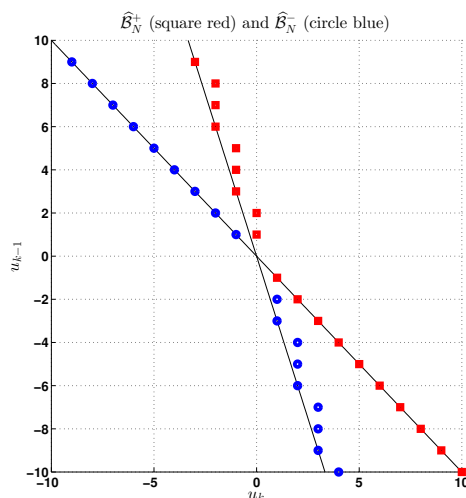


FIGURE 7.5 – Représentation graphique des ensembles $\hat{\mathcal{B}}_N^+$ et $\hat{\mathcal{B}}_N^-$ avec $\delta = 1$. $\hat{\mathcal{B}}_N$ est l'union des ensembles $\hat{\mathcal{B}}_N^+$ et $\hat{\mathcal{B}}_N^-$.

distribuée dans $\{0; 1; \dots; m - 1\}$, si la séquence $\{\lambda(t)\}$ n'est pas corrélée avec la série temporelle $\{q_t\}$, alors

$$\lim_{m \rightarrow \infty} \left(\lim_{N \rightarrow \infty} \hat{\mathcal{B}}_N \right) \rightarrow \mathcal{B} \quad (7.22)$$

Démonstration :

Pour N qui tend vers l'infini, du fait que la série temporelle $\{q(t)\}$ est uniformément distribuée dans $\{0; 1; \dots; m - 1\}$ et que l'état discret $\{\lambda(t)\}$ n'est pas corrélé avec la série temporelle $\{q(t)\}$, chaque sous-système fournit au moins une fois une sortie pour chaque vecteur d'observations $\varphi(t)$. Il s'en suit que $\hat{\mathcal{B}}_N$ correspond aux bandes paramétrées par les vecteurs de paramètres θ_i .

L'épaisseur maximum de ces bandes est $\sqrt{n + 1}\delta$. Si m tend vers l'infini, alors δ tend vers 0, et ainsi

$$\lim_{m \rightarrow \infty} \left(\lim_{N \rightarrow \infty} \hat{\mathcal{B}}_N \right) \rightarrow \mathcal{B} \quad (7.23)$$

7.3.2.2 Identification des sous-systèmes

$\widehat{\mathcal{B}}_N = \widehat{\mathcal{B}}_N^+ \cup \widehat{\mathcal{B}}_N^-$ correspond aux bandes des vecteurs d'observations $\varphi(t)$ dans le voisinage de \mathcal{B} . L'épaisseur de ces bandes est $\sqrt{n+1}\delta$. Ainsi, pour tous les vecteurs d'observations $\varphi(t) \in \widehat{\mathcal{B}}_N$, il existe un vecteur de paramètres θ_i tel que

$$d(\varphi(t), \theta_i) \leq \sqrt{n+1} \frac{\delta}{2} \quad (7.24)$$

où $d(\varphi(t), \theta_i) = \frac{|\varphi(t)^T \theta_i|}{\|\theta_i\|_2} = |\varphi(t)^T \theta_i|$ correspond à la distance entre le vecteur de paramètre $\varphi(t)$ et l'hyperplan paramétré par le vecteur de paramètre θ_i .

Il est possible maintenant, à partir de la connaissance de l'ensemble $\widehat{\mathcal{B}}_N$, de remplacer le problème d'identification initial par le problème d'identification suivant : connaissant les vecteurs d'observations $\varphi(t) \in \widehat{\mathcal{B}}_N$, estimer les vecteurs de paramètres $\{\theta_i\}_{i=1}^{n_s}$ tel que $\|\theta_i\|_2 = 1$ et

$$\varphi(t)^T \theta_i = v(t) \quad (7.25)$$

où $|v(t)| \leq \sqrt{n+1} \frac{\delta}{2}$.

Un problème d'identification similaire est déjà considéré dans OZAY, LAGOA et SZNAIER, 2015, GOUDJIL et al., 2016b and GOUDJIL et al., 2016a. Cela correspond à l'identification de systèmes commutants en présence du bruit borné. Pour la suite, dans un premier temps, l'algorithme proposé dans GOUDJIL et al., 2016b et GOUDJIL et al., 2016a est utilisé pour l'estimation du vecteur de paramètre θ_i , ensuite, le vecteur de paramètre θ_i est normalisé en accord avec l'hypothèse 1.

7.3.2.3 Résumé de l'algorithme

L'algorithme proposé est résumé dans le tableau 7.1. Voici quelques commentaires sur cet algorithme :

- Pour avoir des estimations fiables des ensembles $\widehat{\mathcal{O}}_N^+$ et $\widehat{\mathcal{O}}_N^-$, il est nécessaire que chaque sous-système fournisse au moins une fois une sortie pour chaque vecteur d'observation $\varphi(t)$. Si il n'est pas possible d'avoir au moins une fois une sortie pour chaque vecteur d'observations $\varphi(t)$, des vecteurs d'observations $\varphi(t)$ non inclus dans l'hyperplan paramétré par le vecteur de paramètre θ_i apparaissent

Résumé de l'algorithme pour l'identification d'un modèle commutant à partir de données binaire

- Données d'entrée : $n, n_s, \{u(t)\}, \{s(t)\}, m$
 - Première étape : Estimation de l'ensemble \mathcal{B}
 - Construction des ensembles $\hat{\mathcal{O}}_N^+, \hat{\mathcal{O}}_N^-$ and $\hat{\mathcal{O}}_N^{+/-}$ à partir de (7.16), (7.17) et (7.18).
 - Construction des ensembles $\hat{\mathcal{B}}_N^+, \hat{\mathcal{B}}_N^-$ and $\hat{\mathcal{B}}_N$ à partir de (7.19), (7.20) et (7.21).
 - Deuxième étape : identification des sous-systèmes
 - Estimation du vecteur de paramètre θ_i à parti du problème d'identification (7.25).
 - Normalisation en suivant l'hypothèse 1.
-

TABLE 7.1 – Résumer de l'algorithme en deux étapes

- dans l'ensemble $\hat{\mathcal{B}}_N$. Dans ce cas, il est préférable d'utiliser l'algorithme d'identification proposé par OZAY, LAGOA et SZNAIER, 2015. Cependant, cet algorithme nécessite beaucoup plus de calculs que la solution proposée par GOUDJIL et al., 2016b et GOUDJIL et al., 2016a.
- Le choix de m ou de δ a un impact direct sur l'estimation de l'ensemble \mathcal{B} . Si δ est petit, on obtient une bonne estimation de l'ensemble \mathcal{B} mais le coût en calcul devient important du fait qu'il y a un nombre important de vecteurs d'observations $\varphi(t)$ dans l'ensemble \mathcal{O}_N . Si δ est élevé, la qualité de l'estimation de l'ensemble \mathcal{B} est dégradée car l'épaisseur de la bande \mathcal{B} augmente. L'influence de m est illustrée dans la section 7.4.
 - Le bruit n'a pas été pris en compte jusqu'ici. Le rapport signal sur bruit peut agrandir ou réduire l'épaisseur de la bande $\hat{\mathcal{B}}_N$. Par conséquent, le principe de l'estimation de l'ensemble \mathcal{B} reste valide et l'algorithme d'identification proposé par OZAY, LAGOA et SZNAIER, 2015, GOUDJIL et al., 2016b et GOUDJIL et al., 2016a peut toujours être utilisé pour la deuxième étape. Le bon comportement de l'algorithme proposé dans un cas bruité est montré dans la section 7.4.

7.4 Exemple numérique

Dans cette section, deux exemples numériques sont présentés pour illustrer le comportement de l'algorithme. Les données sont générées à partir des équations (7.1), (7.2) and (7.3).

7.4.1 Premier exemple

On considère un système commutant linéaire tel que $n = 1$ et $n_s = 2$. Les sous-systèmes sont paramétrés par :

$$\begin{cases} \theta_1^T = [0.948 & -0.316] \\ \theta_2^T = [0.832 & 0.554] \end{cases} \quad (7.26)$$

La séquence $\{\lambda(t)\}$ est uniformément distribuée dans $\{1, 2\}$. La séquence d'entrée $\{u(t)\}$ est construite comme suggéré dans le paragraphe 7.3.2.1 avec $M = 10$. Trois différentes valeurs de m sont testées : $m = 5$, $m = 21$ et $m = 41$. Chaque vecteur d'observations $\varphi(t)$ est répété 10 fois pour s'assurer que chaque sous-système fournit au moins une fois une sortie pour chaque vecteur d'observation φ_t possible.

Le tableau 7.2 expose les paramètres à estimer et leurs estimations. On remarque que l'algorithme proposé permet bien d'estimer les paramètres de chaque sous-système. δ a une influence sur la précision de l'estimation. Un δ petit permet de donner une meilleur estimation des paramètres des sous-systèmes.

7.4.2 Deuxième exemple

Pour le deuxième exemple, on considère un système commutant linéaire avec $n = 2$ et $n_s = 3$. Les sous-systèmes sont paramétrés par :

$$\begin{cases} \theta_1^T = [0.808 & 0.505 & -0.303] \\ \theta_2^T = [0.809 & -0.498 & 0.311] \\ \theta_3^T = [0.811 & 0.486 & 0.324] \end{cases} \quad (7.27)$$

	$\theta_1(1)$	$\theta_1(2)$	$\theta_2(1)$	$\theta_2(2)$
valeurs réelles	0.948	-0.316	0.832	0.554
estimation avec $m = 41$ ($\delta = 0.5$)	0.947	-0.319	0.828	0.560
estimation avec $m = 21$ ($\delta = 1$)	0.945	-0.325	0.824	0.565
estimation avec $m = 5$ ($\delta = 5$)	0.961	-0.275	0.845	0.534

TABLE 7.2 – Paramètres estimés du premier exemple

La séquence $\{\lambda(t)\}$ est uniformément distribuée dans $\{1, 2, 3\}$. La séquence d'entrée $\{u(t)\}$ est construite comme suggérée dans le paragraphe 7.3.2.1 avec $M = 10$ et $m = 11$. Le nombre de données disponibles est $N = 50000$.

On souhaite ici étudier l'influence du bruit. Pour cela, une simulation de Monte Carlo est utilisée avec 100 simulations pour un rapport signal sur bruit (ou SNR) de $10dB$ et de $40dB$. La figure 7.6 présente la valeur moyenne de l'erreur ($mean\left(\sum_{i=1}^3 \|\theta_i - \hat{\theta}_i\|_2\right)$) en fonction du rapport signal sur bruit. Cette figure montre que plus le rapport signal sur bruit est élevé, plus l'estimation des paramètres des sous-systèmes est précise. Ce résultat est confirmé par le tableau 7.3 qui expose les paramètres à estimer et leurs estimations pour un rapport signal sur bruit de $10dB$ et de $40dB$.

7.5 Conclusion

Ce chapitre propose une solution pour l'identification d'un système commutant linéaire à partir de données binaires. L'algorithme proposée consiste en deux étapes. La

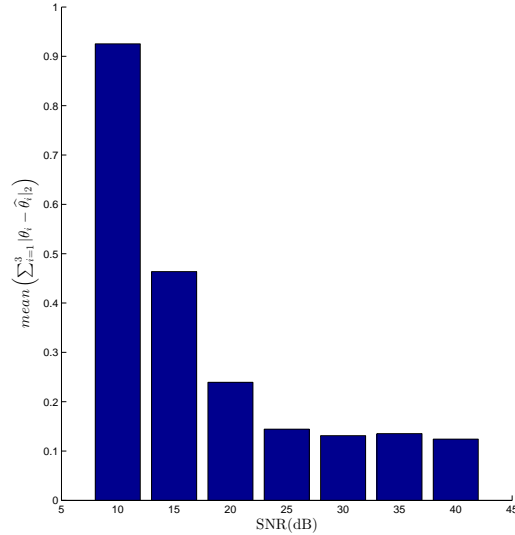


FIGURE 7.6 – La moyenne de l’erreur $mean \left(\sum_{i=1}^3 \|\theta_i - \hat{\theta}_i\|_2 \right)$ en fonction du rapport signal sur bruit.

première étape consiste en la création d’ensembles paramétrés par les vecteurs de paramètres des sous-systèmes. Cette étape permet de transformer le problème d’identification initial en un problème d’identification d’un système commutant linéaire en présence de bruit borné. La deuxième étape consiste en la résolution du problème d’identification d’un système commutant linéaire en présence d’un bruit borné en utilisant un algorithme présent dans la littérature.

Ce chapitre donne des résultats intéressants sur l’identification d’un modèle commutant linéaire à partir de données binaires. Cependant, il reste des points à éclaircir comme la quantification de l’impact du bruit ou le nombre maximal de sous-systèmes discernables.

	$\theta_1(1)$	$\theta_1(2)$	$\theta_1(3)$	$\theta_2(1)$	$\theta_2(2)$	$\theta_2(3)$	$\theta_3(1)$	$\theta_3(2)$	$\theta_3(3)$
Valeurs réelles	0.808	0.505	-0.303	0.809	-0.498	0.311	0.811	0.486	0.324
moyenne avec SNR=40dB	0.807	0.509	-0.294	0.812	-0.491	0.313	0.809	0.451	0.346
variance avec SNR=40dB	0.010	0.036	0.035	0.004	0.010	0.016	0.011	0.069	0.076
moyenne avec SNR=10dB	0.642	0.522	-0.232	0.567	-0.444	0.511	0.805	0.212	0.350
variance avec SNR=10dB	0.183	0.324	0.397	0.245	0.293	0.290	0.147	0.382	0.157

TABLE 7.3 – Paramètres estimés du deuxième exemple

Chapitre 8

Conclusion générale

8.1 Conclusion

Dans ce manuscrit de thèse, le problème de reconnaissance d'activités avec le dispositif eTact a été présenté. La particularité de cette thèse est d'utiliser des données binarisées de l'accéléromètre pour faire de la reconnaissance d'activités.

Pour traiter le problème de reconnaissance activités à partir de données d'un accéléromètre, une bibliographie est présentée dans le chapitre 2 pour illustrer les différentes solutions présentes dans la littérature. La méthode générale la plus souvent utilisée pour faire de la reconnaissance d'activités est un algorithme de classification supervisé pour classer les différentes activités. L'utilisation d'un algorithme d'apprentissage requiert un pré-traitement des données. Pour commencer, les données sont découpées en fenêtres temporelles. Ensuite, des paramètres sont calculés sur ces fenêtres de temps. Ces paramètres sont finalement utilisés comme données d'entrée pour les algorithmes de classification. Bien que la plupart des solutions présentées dans la littérature utilisent cette même méthode, il existe de nombreuses variantes qui diffèrent en fonction des paramètres choisis pour les différentes étapes de la reconnaissance d'activités. On peut citer comme principaux paramètres le positionnement du capteur, la taille de la fenêtre de temps, la fréquence des données de l'accéléromètre, le filtrage des données de l'accéléromètre, le choix des paramètres de la classification et le type d'algorithme de classification supervisé utilisé.

Pour l'élaboration de l'algorithme de reconnaissance d'activités pour eTact, les choix des paramètres des différentes étapes sont motivés par des contraintes matérielles d'eTact, de l'utilisation souhaité d'eTact et des performances de l'algorithme de la reconnaissance

d'activités a proprement dit. Les différentes contraintes sont présentées dans le chapitre 3. Quatre différentes positions du capteur sont étudiées : au torse, au poignet, à la cheville et la taille. L'autonomie d'eTact est un enjeu important qui dépend de la mémoire et de la batterie. Pour optimiser l'autonomie d'eTact, trois différentes approches sont étudiées dans le chapitre 4. Pour la première approche, toutes les données brutes de l'accéléromètre sont stockées, avant d'être renvoyées vers une plateforme externe. Le calcul des paramètres et la mise en oeuvre de la classification est effectuée sur la plateforme externe. Pour la deuxième approche, les paramètres sont calculés directement sur eTact à partir des données brutes de l'accéléromètre. Ces paramètres sont ensuite stockés pour être envoyés sur une plateforme externe sur laquelle sera réalisée la classification. Pour la troisième approche, le calcul des paramètres et la classification sont effectués entièrement sur eTact. Seuls les résultats de la classification sont stockés et envoyés ensuite vers une plateforme externe.

Une solution proposée dans cette thèse est d'utiliser les données binarisées d'un accéléromètre pour faire de la reconnaissance d'activités. L'avantage de cette solution est de pouvoir stocker un plus grand nombre de données par rapport aux cas où les données brutes sont stockées sur eTact. Il est montré dans le chapitre 2 que les paramètres d'un modèle AR peuvent être utilisés comme paramètres pour les algorithmes de classification. C'est pourquoi l'étude du problème d'identification d'un modèle AR à partir de données binaires ou quantifiées est proposée dans le chapitre 5. Il existe une solution dans la littérature dans le cas où les données sont binarisées avec un seuil nul. Une contribution de cette thèse est de proposer une solution dans le cas où le signal est binarisé avec un seuil non nul. L'analyse de la convergence et de la variance sont détaillées dans ce chapitre ainsi que des résultats numériques. Pour continuer, plusieurs extensions de cet algorithme sont proposées dans ce chapitre. La deuxième extension est un algorithme adapté pour le temps réel. La troisième extension est un algorithme pour des données quantifiées. Pour finir, l'algorithme est utilisé dans un problème de reconnaissance d'activités pour montrer l'impact du choix du seuil de binarisation ainsi que l'impact de la taille de la fenêtre de temps.

Dans le chapitre 6, une nouvelle approche est étudiée pour faire de la reconnaissance d'activités avec eTact, à l'image de ce qui a été fait dans le chapitre 4. La particularité de cette approche est d'utiliser les données binarisées de l'accéléromètre pour l'élaboration

de l'algorithme de reconnaissance d'activités. Cette approche consiste à stocker les données binaires, puis à les envoyer vers une plateforme externe. Le calcul des paramètres et la mise en oeuvre de l'algorithme sont effectués sur la plateforme externe. L'avantage principal d'utiliser des données binaires est d'augmenter l'autonomie mémoire d'eTact d'un facteur 16 par rapport à la première approche présentée dans le chapitre 4. Cependant, la binarisation des données entraîne une perte d'information. L'algorithme d'identification des paramètres d'un modèle AR à partir de données binaires présenté dans le chapitre 5 est utilisé. Cependant, les paramètres du modèle AR ne sont pas les seuls paramètres utilisables pour des données binaires un nouvel ensemble de paramètre pour le cas de données binaires est calculé dans ce chapitre. Les résultats de l'approche étudiée dans ce chapitre sont finalement comparés aux résultats des trois approches du chapitre 5

Pour finir, l'étude de série temporelle nous a conduit à l'étude d'autres modèles. Dans le chapitre 7, nous proposons un algorithme pour l'identification d'un modèle commutant à partir de données binaires. L'étude de ce système est motivée par le fait que chaque système peut modéliser un type d'activité et la détection d'activité revient à déterminer quel est le sous-système actif. L'algorithme d'identification est composé de deux étapes. La première étape consiste à construire des sous-ensembles définis par les vecteurs de paramètres. Cette étape a pour but de transformer le problème d'identification initial en un problème d'identification de systèmes commutants linéaires en présence d'un bruit borné. La deuxième étape consiste à résoudre ce nouveau problème d'identification à l'aide d'algorithmes disponibles dans la littérature. L'algorithme est ensuite testé sur des exemples numériques.

8.2 Perspectives

Dans ce manuscrit, la base de données utilisée pour les algorithmes de classification supervisés est issue de la littérature. La prochaine étape va consister à construire une base de données d'apprentissage collectées avec eTact. L'intérêt de construire cette base de données est de collecter des activités ciblées pour une utilisation précise d'eTact.

Malgré les résultats encourageants pour la reconnaissance d'activités avec eTact, il reste encore plusieurs problématiques à traiter. Ainsi, la reconnaissance d'activités a été appliquée que dans le cas d'activités connues. Cependant, il est possible qu'une personne

puisse faire une activité non présente dans la base d'apprentissage et cette activité inconnue sera classée parmi une des activités présentes dans la base de données. Un autre point est le fait que le changement d'activité n'a pas été pris en compte. Pour la totalité de chaque fenêtre glissante, une seule activité est présente. Dans le cas réel, le changement d'activités d'une personne induit la création de fenêtre de temps qui contient la transition entre activités. La classification sur ces fenêtres peut entraîner des erreurs dans le résultats de classification. Pour la suite, il est nécessaire de mesurer l'influence de ces problématiques sur le résultat réel de l'algorithme de reconnaissance d'activités et de proposer des solutions pour réduire leurs impacts sur le résultat.

Un dernier ensemble de perspective concern les problématiques d'identification. En premier lieu, il peut être intéressant d'étendre le résultat de l'identification d'un modèle AR à partir de données binaires à d'autres modèles tel que ARMA. En second lieu, il peut être intéressant d'étendre le résultat de l'identification de système commutant à sortie binaire à l'identification de série temporelles commutantes à sortie quantifié.

Bibliographie

- ANTONSSON, E. K. et R. W. MANN (1985). « The frequency content of gait ». In : *Journal Biomechanics* 18, p. 39–47.
- ATALLAH, L. et al. (2011). « Sensor Placement for Activity Detection Using Wearable Accelerometers ». In : *IEEE Transactions on Biomedical Circuits and Systems* 5.
- ATTAL, F. et al. (2015). « Physical Human Activity Recognition Using Wearable Sensors ». In : *Sensors* 15, p. 31314–31338.
- AUBER, R. et al. (2018a). « Activity Recognition from Binary Data ». In : *UKACC International Conference on Control, Sheffield*.
- AUBER, R. et al. (2018b). « Estimation of Auto-Regressive models for time series using Binary or Quantized Data ». In : *IFAC*.
- AUBER, R. et al. (2019a). « Identification for Switched FIR Linear Systems using Binary Measurements ». In : *58th Conference on Decision and Control, Nice*.
- AUBER, R. et al. (2019b). « Identification of AR Time-series based on binary data ». In : *IET Research Journals*.
- BAGNALL, A.J. et G.J. JANACEK (2004). « Clustering time series from ARMA models with clipped data ». In : *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle*.
- (2005). « Clustering Time Series with Clipped Data ». In : *Machine Learning* 58, p. 151–178.
- BANOS, O. et al. (2014). « Window Size Impact in Human Activity Recognition ». In : *Sensors* 14, p. 6474–6499.
- BAYAT, A., M. POMPLUN et D.A. TRAN (2014). « A Study on Human Activity Recognition Using Accelerometer Data from Smartphones ». In : *Procedia Computer Science* 34, p. 450–457.
- BLACKWELL, T. et al. (2008). « Comparison of Sleep Parameters from Actigraphy and Polysomnography in Older Women : The SOF Study ». In : *Sleep* 31, p. 283–291.

- BOTTEGAL, G., H. HJALMARSSON et G. PILLONETTO (2017). « "A new kernel-based approach to system identification with quantized output data" ». In : *Automatica* 85, p. 145–152.
- BOURGOIS, L. et J. JUILLARD (2012). « "Convergence Analysis of an Online Approach to Parameter Estimation Problems Based on Binary Noisy Observations" ». In : *IEEE Conference on Decision and Control, Hawaii*.
- BOUTEN, C. V. C. et al. (1997). « A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity ». In : *IEEE Transactions on Biomedical Engineering* 44, p. 136–147.
- BREIMAN, L. (1996). « Bagging predictors ». In : *Machine Learning* 24, p. 123–140.
- (2001). « Random Forests ». In : *Machine Learning* 45, p. 5–32.
- BREIMAN, L. et al. (1984). « Classification and regression trees ». In : *Chapman and Hall/CRC*.
- BRESCHI, V., D. PIGA et A. BEMPORAD (2016). « "Piecewise affine regression via recursive multiple least squares and multcategory discrimination" ». In : *Automatica* 73, p. 155–162.
- BRUCE, L. M., C. H. KOGER et J. LI (2002). « Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction ». In : *IEEE Transactions on Geoscience and Remote Sensing* 40, p. 2331–2338.
- BURGES, C.J.C (1998). « A Tutorial on Support Vector Machines for Pattern Recognition ». In : *Data Mining and Knowledge Discovery* 2, p. 121–167.
- BURRUS, C., R. A. GOPINATH et H. T. GUO (1998). « Introduction to Wavelets and Wavelet Transforms : a Primer ». In : *Prentice Hall*.
- CASALE, P., O. PUJOL et P. RADEVA (2011). « Human Activity Recognition from Accelerometer Data Using a Wearable Device ». In : *Iberian Conference on Pattern Recognition and Image Analysis, Las Palmas*.
- CASTRO, D. et al. (2017). « Wearable-Based Human Activity Recognition Using an IoT Approach ». In : *Journal of Sensor and Actuator Networks* 6.
- CERON, J. D., D M. LOPEZ et B.M. ESKOFIER (2018). « Human Activity Recognition Using Binary Sensors, BLE Beacons, an Intelligent Floor and Acceleration Data : A Machine Learning Approach ». In : *International Conference on Ubiquitous Computing and Ambient Intelligence, Punta Cana*.
- CHAPRON, K. et al. (2018). « A More Efficient Transportable and Scalable System for Real-Time Activities and Exercises Recognition ». In : *Sensors* 18.
- CHEN, Y. et C. SHEN (2017). « Performance Analysis of Smartphone-Sensor Behavior for Human Activity Recognition ». In : *IEEE Access* 5, p. 3095–3110.

- CLELAND, I. et al. (2013). « Optimal Placement of Accelerometers for the Detection of Everyday Activities ». In : *Sensors* 13, p. 9183–9200.
- COMPANJEN, B. (2009). « Classification methods for activity recognition ». In : *Twente Student Conference on IT, Twente*.
- DAGHISTANI, T. et R. ALSHAMMARI (2016). « Improving Accelerometer-Based Activity Recognition by Using Ensemble of Classifiers ». In : *International Journal of Advanced Computer Science and Applications* 7, p. 128–133.
- DANDAWATE, A. V. et G. B. GIANNAKIS (1993). « Asymptotic properties and covariance expressions of k th-order sample moments and cumulants ». In : *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, p. 1186–1190.
- DASH, M. et H. LIU (1997). « Feature selection for classification ». In : *Intelligent Data Analysis* 1, p. 131–156.
- DAUBECHIES, I. (1992). « Ten Lectures on Wavelets ». In : *Springer-Verlag*.
- DAVID, F.N. (1953). « A note on the evaluation of the multivariate normal integral ». In : *Biometrika* 40, p. 458–459.
- DAVILA, J.C., A. CRETU et M. ZAREMBA (2017). « Wearable Sensor Data Classification for Human Activity Recognition Based on an Iterative Learning Framework ». In : *Sensors* 17.
- DUDA, R. O., P. E. HART et D. G. STORK (2001). « Pattern Classification ». In : *Wiley-Interscience*.
- DUTTA, A. et al. (2018). « Identifying Free-Living Physical Activities Using Lab-Based Models with Wearable Accelerometers ». In : *Sensors* 18, p. 190–202.
- ENGELBRECHT, A.P. (2007). « Computational Intelligence : An Introduction ». In : *Wiley Publishing*.
- ERMES, M., J. PARKKA et L. CLUITMANS (2008). « Advancing from offline to online activity recognition with wearable sensors ». In : *IEEE International Conference of Engineering in Medicine and Biology Society, Vancouver*.
- ESPINILLA, M. et al. (2018). « Human Activity Recognition from the Acceleration Data of a Wearable Device. Which Features Are More Relevant by Activities? ». In : *International Conference on Ubiquitous Computing and Ambient Intelligence, Punta Cana*.
- EYOBU, O.S. et D. HAN (2018). « Feature Representation and Data Augmentation for Human Activity Classification Based on Wearable IMU Sensor Data Using a Deep LSTM Neural Network ». In : *Sensors* 18.
- FAWCETT, T. (2004). « ROC Graphs : Notes and Practical Considerations for Researchers ». In : *Pattern Recognition Letters* 31, p. 1–38.

- FERRARI-TRECATE, G. et al. (2003). « "A clustering technique for the identification of piecewise affine systems" ». In : *Automatica* 39.2, p. 250–217.
- FIGO, D. et al. (2010). « Preprocessing Techniques for Context Recognition from Accelerometer Data ». In : *Personal Ubiquitous Computing* 14, p. 645–662.
- FULLERTON, E., B. HELLER et M. MUNOZ-ORGANERO (2017). « Recognizing Human Activity in Free-Living Using Multiple Body-Worn Accelerometers ». In : *IEEE Sensors Journal* 17, p. 5290–5297.
- GHAZALI, N.F. et al. (2018). « Common sport activity recognition using inertial sensor ». In : *IEEE International Colloquium on Signal Processing Its Applications, Batu Feringghi*.
- GOUDJIL, A. et al. (2016a). « "A real-time identification algorithm for switched linear systems with bounded noise" ». In : *European Control Conference, Aalborg*.
- (2016b). « "Convergence analysis of a real-time identification algorithm for switched linear systems with bounded noise" ». In : *IEEE Conference on Decision and Control, Las Vegas*.
- GU, F. et al. (2015). « User-Independent Motion State Recognition Using Smartphone Sensors ». In : *Sensors* 15, p. 30636–30652.
- GUO, J. et Y. ZHAO (2013). « Recursive projection algorithm on FIR system identification with binary-valued observations ». In : *Automatica* 49.11, p. 3396–3401.
- GUO, J. et al. (2015). « Recursive identification of FIR systems with binary-valued outputs and communication channels ». In : *Automatica*, p. 165–172.
- GUPTA, P. et T. DALLAS (2014). « Feature Selection and Activity Recognition System Using a Single Triaxial Accelerometer ». In : *IEEE Transactions on Biomedical Engineering* 61, p. 1780–1786.
- HASHAMBHOY, Y. et R. VIDAL (2005). « "Recursive identification of switched ARX models with unknown number of models and unknown orders" ». In : *IEEE Conference on Decision and Control, Seville*.
- HE, Z. et L. JIN (2008). « Activity Recognition from Acceleration Data Using AR Model Representation and SVM ». In : *IEEE International Conference on Machine Learning and Cybernetics, Kunming*.
- (2009). « Activity Recognition from acceleration data Based on Discrete Cosine Transform and SVM ». In : *IEEE International Conference on Machine Learning and Cybernetics, San Antonio*.
- HO, T. (1998). « The random subspace method for constructing decision forests ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, p. 832–844.

- JAFARI, K., J. JUILLARD et M. ROGER (2012). « "Convergence analysis of an online approach to parameter estimation problems based on binary observations" ». In : *Automatica* 48.11, p. 2837–2842.
- JAIN, A. et V. KANHANGAD (2018). « Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensor ». In : *IEEE Sensors Journal* 18, p. 1169–1177.
- JANIDARMIAN, M. et al. (2017). « A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition ». In : *Sensors* 17.
- JASIEWICZ, J.M. et al. (2006). « Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals ». In : *Gait Posture* 24, p. 502–509.
- JOLLIFFE, I.T. (2002). « Principal Component Analysis ». In : *Springer*.
- JULOSKI, A., S. WEILAND et W. P. M. H. HEEMELS (2005). « "A Bayesian approach to identification of hybrid systems" ». In : *IEEE Transactions on Automatic Control* 50.10, p. 1520–1533.
- KEDEM, B. (1980). « Estimation of the Parameters in Stationary Autoregressive Processes after Hard Limiting ». In : *Journal of The American Statistical Association* 75, p. 146–153.
- KEDEM, B. et E. SLUD (1981). « On Goodness of Fit of Time Series Models : An Application of Higher Order Crossings ». In : *Biometrika* 68, p. 551–556.
- KHAN, A.M., Y.K. LEE et T. KIM (2008). « Accelerometer Signal-based Human Activity Recognition Using Augmented Autoregressive Model Coefficients and Artificial Neural Nets ». In : *IEEE International Conference of Engineering in Medicine and Biology Society, Vancouver*.
- KHAN, A.M. et al. (2010). « A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer ». In : *IEEE Transactions on Information Technology in Biomedicine* 14, p. 1166–1172.
- KIM, T., J. CHO et J.T. KIM (2012). « Mobile Motion Sensor-Based Human Activity Recognition and Energy Expenditure Estimation in Building Environments ». In : *International Conference in Sustainability in Energy and Buildings, Stockholm*.
- KIRA, K. et L.A. RENDELL (1992). « A Practical Approach to Feature Selection ». In : *International Workshop on Machine Learning, Aberdeen*.
- KONONENKO, I. (1994). « Estimating attributes : Analysis and extensions of RELIF ». In : *European Conference on Machine Learning, Catania*.
- LARA, O.D. et M.A. LABRADOR (2013). « A Survey on Human Activity Recognition using Wearable Sensors ». In : *IEEE Communications Surveys & Tutorials* 15, p. 1192–1209.

- LAUER, F. (2013). « "Estimating the probability of success of a simple algorithm for switched linear regression" ». In : *Nonlinear Analysis : Hybrid Systems* 8, p. 31–47.
- LE, V.L., F. LAUER et G. BLOCH (2013). « "Identification of linear hybrid systems : A geometric Approach" ». In : *IEEE American Control Conference, Washington*, p. 830–835.
- LEE, M. et al. (2011). « A single tri-axial accelerometer-based real-time personal life log system capable of activity classification and exercise information generation ». In : *Personal and Ubiquitous Computing* 15, p. 887–898.
- LI, F. et al. (2018). « Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors ». In : *Sensors* 18.
- LIU, H. et R. SETIONO (1996). « A Probabilistic Approach to Feature Selection - A Filter Solution ». In : *International Conference on International Conference on Machine Learning, Bari*, p. 319–327.
- MA, Y. et R. VIDAL (2005). « "Identification of deterministic switched ARX systems via identification of algebraic varieties" ». In : *International Conference on Hybrid Systems : Computation and Control, Zurich*.
- MEHRANG, S., J. PIETILA et I. KORHONEN (2018). « An Activity Recognition Framework Deploying the Random Forest Classifier and A Single Optical Heart Rate Monitoring and Triaxial Accelerometer Wrist-Band ». In : *Sensors* 18.
- MITCHELL, E., D. MONAGHAN et N. E. O'CONNOR (2013). « Classification of Sporting Activities Using Smartphone Accelerometers ». In : *Sensors* 13, p. 5317–5337.
- MITCHELL, T. M. (1997). « Machine Learning ». In : *McGraw-Hill, Inc.*
- NAKADA, H., K. TAKABA et T. KATAYAMA (2005). « "Identification of piecewise affine systems based on statistical clustering technique" ». In : *Automatica* 41.5, p. 905–913.
- NGUYEN, A., D. MOORE et I. MCCOWAN (2007). « Unsupervised Clustering of Free-Living Human Activities using Ambulatory Accelerometry ». In : *IEEE International Conference of Engineering in Medicine and Biology Society, Lyon*.
- NI, Q., L. ZHANG et L. LI (2018). « A Heterogeneous Ensemble Approach for Activity Recognition with Integration of Change Point-Based Data Segmentation ». In : *Applied Sciences* 8.
- OZAY, N., C. LAGOA et M. SZNAIER (2015). « "Set membership identification of switched linear systems with known number of subsystems" ». In : *Automatica* 51.1, p. 180–191.
- PANNURAT, N. et al. (2017). « Analysis of Optimal Sensor Positions for Activity Classification and Application on a Different Data Collection Scenarios ». In : *Sensors* 17.

- PITTNER, S. et S. KAMARTHI (1999). « Feature Extraction From Wavelet Coefficients for Pattern Recognition Tasks ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, p. 83–88.
- PRASERTSUNG, P. et T. HORANONT (2016). « A classification of accelerometer data to differentiate pedestrian state ». In : *International Computer Science and Engineering Conference, Chiang Mai*.
- PREECE, S.J. et al. (2009a). « A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data ». In : *IEEE Transactions on Biomedical Engineering* 56, p. 871–879.
- PREECE, S.J. et al. (2009b). « Activity identification using body-mounted sensors—a review of classification techniques ». In : *Physiological Measurement* 30.
- PUDIL, P., J. NOVOVICOVA et J. KITTLER (1994). « Floating search methods in feature selection ». In : *Pattern Recognition Letters* 15, p. 1119–1125.
- QUINLAN, J.R. (1986). « Induction of Decision Trees ». In : *Machine Learning* 1, p. 81–106.
- (1993). « C4.5 : Programs for Machine Learning ». In : *Morgan Kaufmann Publishers*.
- RAHIM, K.N Ku Abd et al. (2018). « Classification of Human Daily Activities Using Ensemble Methods Based on Smartphone Inertial Sensors ». In : *Sensors* 18.
- RAILEANU, L.E. et K. STOFFEL (2004). « Theoretical Comparison between the Gini Index and Information Gain Criteria. » In : *Annals of Mathematics and Artificial Intelligence* 41.
- REISS, A. et D. STRICKER (2012a). « Introducing a New Benchmarked Dataset for Activity Monitoring ». In : *2012 16th International Symposium on Wearable Computers*.
- REISS, Attila et Didier STRICKER (2012b). « Creating and Benchmarking a New Dataset for Physical Activity Monitoring ». In : *The 5th Workshop on Affect and Behaviour Related Assistance*.
- ROBNIK-SIKONJA, M. et I. KONONENKO (2003). « Theoretical and Empirical Analysis of Relief and RRelief ». In : *Machine Learning*, p. 23–69.
- ROKACH, L. et O. MAIMON (2014). « Data Mining With Decision Trees : Theory and Applications ». In : *World Scientific Publishing*.
- ROSATI, S., G. BALESTRA et M. KNAFLITZ (2018). « Comparison of Different Sets of Features for Human Activity Recognition by Wearable Sensors ». In : *Sensors* 18.
- RUEDA, F.M. et al. (2018). « Convolutional Neural Networks for Human Activity Recognition Using Body-Worn Sensors ». In : *Informatics* 5.
- SAEEDI, S. et N. EL-SHEIMY (2015). « Activity Recognition Using Fusion of Low-Cost Sensors on a Smartphone for Mobile Navigation Application ». In : *Micromachines* 6, p. 1100–1134.

- SAEZ, Y., A. BALDOMINOS et P. ISASI (2017). « A Comparison Study of Classifier Algorithms for Cross-Person Physical Activity Recognition ». In : *Sensors* 17.
- SAGI, O. et L.R. ROKACH (2018). « Ensemble learning : A survey ». In : *Wiley Data Mining and knowledge discovery* 8.
- SAIF, S., S. TEHSEEN et S. KAUSAR (2018). « A Survey of the Techniques for The Identification and Classification of Human Actions from Visual Data ». In : *Sensors* 18.
- SAN-SEGUNDO, R. et al. (2018). « Robust Human Activity Recognition using smartwatches and smartphones ». In : *Engineering Applications of Artificial Intelligence* 72, p. 190–202.
- SEKINE, M. et al. (1998). « Classification of acceleration waveform in a continuous walking record ». In : *IEEE International Conference of Engineering in Medicine and Biology Society, Hong Kong*.
- SHEPPARD, W. F. (1899). « On the application of the theory of error to cases of normal distribution and normal correlation ». In : *Philosophical Transactions of the Royal Society A* 192, p. 101–167.
- SHOAIB, M. et al. (2015). « A Survey of Online Activity Recognition Using Mobile Phones ». In : *Sensors* 15, p. 2059–2085.
- SKURICHINA, M. et R. DUIN (2002). « Limited Bagging, Boosting and the Random Subspace Method for Linear Classifiers ». In : *Formal Pattern Analysis & Application* 5, p. 121–135.
- STEINWART, I. et A. CHRISTMANN (2008). « Support Vector Machines ». In : *Springer*.
- THARWAT, A. et al. (2017). « Linear discriminant analysis : A detailed tutorial ». In : *AI Communications* 30, p. 169–190.
- TWOMEY, N. et al. (2018). « A Comprehensive Study of Activity Recognition Using Accelerometers ». In : *Informatics* 5.
- URBANOWICZ, R. et al. (2017). « Relief-Based Feature Selection : Introduction and Review ». In : *Journal of Biomedical Informatics*.
- VELTINK, P. H. et al. (1996). « Detection of static and dynamic activities using uniaxial accelerometers ». In : *IEEE Transactions on Rehabilitation Engineering* 4, p. 375–385.
- VIDAL, R. (2008). « "Recursive identification of switched ARX systems" ». In : *Automatica* 44.9, p. 2274–2287.
- VYAS, V.V., K. H. WALSE et R. V. DHARASKAR (2017). « A Survey on Human Activity Recognition using Smartphone ». In : *International Journal of Advance Research in Computer Science and Management Studies* 5, p. 118–125.
- WANG, G. et al. (2018). « Impact of Sliding Window Length in Indoor Human Motion Modes and Pose Pattern Recognition Based on Smartphone Sensors ». In : *Sensors* 18.

- WANG, J. et al. (2011a). « Recognizing Human Daily Activities From Accelerometer Signal ». In : *Procedia Engineering* 15, p. 1780–1786.
- WANG, W. et al. (2011b). « Analysis of filtering methods for 3D acceleration signals in body sensor network ». In : *International Symposium on Bioelectronics and Bioinformatics, Suzhou*.
- WHITNEY, A. W. (1971). « A Direct Method of Nonparametric Measurement Selection ». In : *IEEE Transactions on Computers*.
- YANG, J., J. WANG et Y. CHEN (2008). « Using acceleration measurements for activity recognition : An effective learning algorithm for constructing neural classifiers ». In : *Pattern Recognition Letters* 29, p. 2213–2220.
- YURTMAN, A., B. BARSHAN et B. FIDAN (2018). « Activity Recognition Invariant to Wearable Sensor Unit Orientation Using Differential Rotational Transformations Represented by Quaternions ». In : *Sensors* 18.
- ZHANG, M. et A. A. SAWCHUK (2011a). « A feature selection-based framework for human activity recognition using wearable multimodal sensors ». In : *BODYNETS*.
- ZHANG, M. et A. A. SAWCHUK (2011b). « A Feature Selection-Based Framework for Human Activity Recognition Using Wearable Multimodal Sensors ». In : *International Conference on Body Area Networks, Beijing*.
- ZHEN, X. et I. V. BASAWA (2010). « Estimation for binary models generated by Gaussian autoregressive processes ». In : *Journal of Statistical Computation and Simulation* 80, p. 1041–1051.

Résumé

Ce manuscrit porte sur la reconnaissance d'activités à partir de données accélérométriques. Le dispositif utilisé pour collecter les données de l'accéléromètre est eTact, dispositif développé par la société Bodycap. Plusieurs solutions sont proposées afin d'optimiser l'autonomie de l'objet connecté. Ces solutions sont mises en oeuvre et comparées sur différentes séries de données. L'originalité d'une de ces solutions consiste à binariser les données de l'accéléromètre avant de les transférer vers une plateforme externe où elles sont analysées. L'utilisation de données binaires entraîne la perte de nombreuses informations, cependant il est montré dans ce manuscrit qu'il est possible d'estimer, entre autres, les paramètres d'un modèle Auto Régressif d'une série temporelle à partir de l'information binaire sur cette série. A ce titre, un algorithme d'identification est proposé et analysé.

Mots clés : données binaires, identification de modèle autorégressif, reconnaissance d'activités, classification, systèmes commutants linéaires.

Abstract

This manuscript deals with the recognition of activities from accelerometric data. The device used to collect the accelerometer data is eTact, a device developed by Bodycap. Several solutions are proposed to optimize the autonomy of the connected object. These solutions are implemented and compared on different data sets. The originality of one of these solutions is to binarize the data of the accelerometer before transferring them to an external platform where they are analyzed. The use of binary data induces the loss of a lot of information, however it is shown in this manuscript that it is possible to estimate, among other things, the parameters of an Auto Regressive model of a time series from the binary information on this series. In this respect, an identification algorithm is proposed and analyzed.

Keys words : binary data, identification of an autoregressive model, activity recognition, machine learning, switched linears systems.