

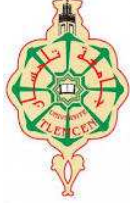
الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

Université Abou Bekr Belkaid
Tlemcen Algérie



جامعة أبي بكر بلقايد

تلمسان الجزائر



FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE



Mémoire présenté en vue de l'obtention du diplôme de
Magister en informatique

Option : *Systemes et Technologies de l'Information
et de la Communication*

Thème : Apprentissage en ligne pour le
développement d'un classifieur de données médicales

Présenté par :
Mr. IMAD GAOUAR

Devant le jury composé de :

Président :	Mr. Bessaid Abdelhafid	Professeur
Directeur de mémoire :	Mr. Chikh Med Amine	Professeur
Examineur:	Mr. Benmamar Badr	MCA
Examinatrice :	Mme Didi Fedoua	MCA
Invité :	Mr. Benazzouz Mortada	MCB

Année : 2014-2015

Dédicaces

Je dédie ce mémoire :

A mes parents

A mes frères et sœurs

A mes beaux-frères et belles sœurs

A mes chers neveux et nièces

A tous mes amis (es)

A tous les membres de ma famille

A tous ceux qui ont contribué à ma formation

Imad

Remerciements

*Je tiens tout d'abord à remercier le directeur de ce mémoire, **Mr. Chikh Mohamed Amine**, Professeur à l'Université de Tlemcen pour m'avoir guidé, encouragé, conseillé et fait confiance tout au long de ce travail.*

*Je tiens à exprimer ma grande considération et ma vive reconnaissance à **Mr. Bessaid Abdelhafid**, Professeur à l'Université ABB de Tlemcen, d'avoir accepté d'assurer la présidence du jury de mon mémoire de magister.*

*Mes vifs remerciements vont à **Mme Didi Fedoua**, Maître de Conférences à l'Université ABB de Tlemcen, d'avoir accepté d'examiner ce mémoire.*

*Je remercie **Mr. Benmamar Badr**, Maître de Conférences à l'Université ABB de Tlemcen, d'avoir accepté de faire partie du jury de mon mémoire de magister.*

*Je tiens à remercier également **Mr. Benazzouz Mortada** pour le grand honneur qu'il me fait en étant présent dans ce jury.*

*Mes remerciements vont également à **Mr. Bechar Mohamed Amine**, doctorant à l'Université de Tlemcen, pour sa gentillesse, ses conseils ainsi que sa disponibilité, je remercie également **Mr Behedada Omar** doctorant à l'Université de Tlemcen pour m'avoir aidé à mettre mes idées en œuvre sur MATLAB. Je remercie aussi **Mme Baba Ahmed née Boublenza Amina** doctorante à l'Université de Tlemcen pour m'avoir orienté et conseiller durant ce travail.*

Enfin, merci à mes parents pour le soutien et l'encouragement qu'ils m'ont apportés tout au long de ce travail, mes frères et sœurs ainsi qu'à tous mes ami(e)s pour leurs encouragements et leurs conseils.

Je remercie toute personne ayant contribué de près ou de loin à l'élaboration de ce mémoire.

IMAD

Table des matières

Introduction générale	1
Chapitre1: Principes de l'apprentissage en ligne	
1. Introduction	3
2. Modèles intelligents	4
2.1 L'IA Symbolique :.....	5
2.2 L'IA Numérique (Réseaux Connexionnistes) :.....	7
2.3 Besoin en apprentissage :.....	8
3. Les modes d'apprentissage artificiel	8
3.1 Apprentissage supervisé.....	9
3.1.1 Apprentissage non-supervisé.....	9
3.1.2 L'apprentissage par renforcement.....	9
4. Différents problèmes en apprentissage artificiel	9
4.1 Problèmes en apprentissage supervisé.....	9
4.1.1 Classification.....	9
4.1.2 Régression.....	10
4.1.3 Séries temporelles.....	10
4.2 Problèmes en apprentissage non-supervisé :.....	10
4.2.1 Estimation de densité.....	10
4.2.2 Partitionnement.....	10
4.2.3 Réduction de dimensionnalité.....	10
5. Relations entre les techniques d'apprentissage	11
6. Apprentissage hors-ligne et Apprentissage en-ligne	11
6.1 Apprentissage hors-ligne.....	11
6.2 Apprentissage en-ligne.....	11
7. Apprentissage statique et apprentissage incrémental	12
7.1 Apprentissage statique.....	12
7.2 Apprentissage incrémental.....	12
8. Système adaptatif et système évolutif	13
8.1 Système adaptatif.....	13
8.2 Système évolutif.....	13

9. Mémoire des données et mémoire des concepts	14
9.1 Mémoire des données.....	14
9.2 Mémoire des concepts.....	14
10. les principaux modèles incrémentaux évolutifs	15
10.1 Cas de mono classifieurs évolutifs.....	15
10.1.1 Les Réseaux de neurones multicouche.....	15
10.1.2 Machines à vecteurs supports (SVM) évolutives	15
10.2 Cas de multi-classifieurs évolutifs.....	16
10.2.1 Systèmes à ensemble évolutifs	16
10.2.1.1 L'algorithme Learn++	16
10.2.1.2 Growing Negative Correlation Learning (GNCL)	17
10.2.2 Les systèmes multi agents	17
10.3 Systèmes d'inférence floue évolutifs.....	18
10.3.1 Exemples de SIF dans la littérature.....	18
10.3.1.1 FLEXible Fuzzy Inference System (FLEXFIS)	18
10.3.1.2 Evolving Fuzzy Neural Networks (EFuNN)	18
10.3.1.3 Dynamic Evolving Neural-Fuzzy Inference System (DENFIS)	19
10.3.1.4 Evolving Takagi-Sugeno (eTS) et eClass (notre sujet)	19
10.3.2 Principes des Systèmes d'inférence floue (SIF) incrémental évolutif	19
10.3.2.1 Théorie de la logique floue.....	19
<i>a) Théorie des ensembles flous</i>	20
<i>b) Inférence floue</i>	20
10.3.2.2 Architecture d'un SIF	21
<i>a) Structure des prémisses</i>	21
<i>b) Structure des conclusions</i>	22
10.3.3 Apprentissage incrémental d'un SIF	24
10.3.3.1 Évolution des prémisses	24
10.3.3.2 Évolution des conclusions	25
10.3.3.3 Stabilité de l'apprentissage.....	25
10.3.3.4 Ajout de classes.....	25
11. Discussion	25
12. Conclusion	26
 Chapitre2: Les classifieurs évolutifs littérature et théorie	
1. Introduction :	27

2. Besoins en classifieurs évolutifs en ligne	28
3. Les classifieurs évolutifs « Evolvingfuzzyclassifiers » :	28
3.1. Comparaison entre les classifieurs évolutifs et les classifieurs génétiques:	29
3.2. Comparaison entre les classifieurs évolutifs et les classifieurs incrémentales :.....	29
4. L’algorithme e-class :	30
4.1 Eclass0 :.....	31
4.1.1 Architecture d’eclass0 :	31
4.1.2 Apprendre la partie antérieure des règles:	34
a) Partitionnement de l’espace des données	34
4.1.3Algorithme eclass0: [Angelovet Zhou, 2008]	37
4.2 Eclass1 :.....	38
4.2.1 Architecture d’eclass1 :	38
4.2.2 Apprentissage de eclass1 :	43
4.2.3 Algorithme d’eclass1: [Angelovet Zhou, 2008]	44
4.3Simpl_eclass : Simplification de l’algorithme E-class	46
4.3.1Introduction	46
4.3.2 Algorithme simplifié pour e-class0:	46
4.3.3Algorithme simplifié pour e-class1 :	50
4.3 Une méthode de pondération de règles en ligne pour classer les flux de données	53
4.4.1 Méthode de réglage de poids :	54
4.4.2 Le mécanisme de réglage de poids:	55
4.4.3 L’ajout d’un nouvel échantillon de donnée:	56
4.4.4 Suppression de l’échantillon de données le plus ancien :	57
4.4.5 L’algorithme proposé par Shahparast et al. [2012] :	57
5. Les domaines d’application des eclass :	58
5.1 Diagnostic médical :.....	59
5.2 Robotique :	63
5.3 Reconnaissance en temps réel des activités humaines :	64
6. Conclusion :	66

Chapitre3: Le cœur et l’électrocardiographie

1. Introduction	67
2. Le système cardiovasculaire	67
3. Anatomie du cœur	68
4. Activité mécanique cardiaque	70

5. Activité électrique du cœur	71
6. L'Electrocardiogramme (ECG).....	73
6.1 Les dérivations.....	75
6.2 Conditions d'enregistrement	77
6.3 Le Holter.....	77
6.54 Ondes et intervalles de L'ECG.....	77
7. Le rythme cardiaque sinusal	79
8. Troubles du rythme et de la conduction cardiaque	80
8.1 Extrasystole ventriculaire	80
8.2 Arythmies à base de fréquence	81
8.2.1 Bradycardie	81
a. Bradycardie sinusale.....	81
b. Bradycardie jonctionnelle.....	81
8.2.2 Tachycardie	82
a. Tachycardie sinusale.....	82
b. Tachycardie auriculaire et nodale AV	83
c. Tachycardie ventriculaire (TV)	83
8.3 Les arythmies à base de régularité.....	84
8.3.1 Foyer ectopique auriculaire ou nodal AV.....	84
8.3.2 Foyer ectopique ventriculaire	84
8.4 Les blocs cardiaques.....	87
9. Parasites et Artefacts visibles sur l'électrocardiogramme	88
9.1 Bruits techniques	89
9.2 Bruits physiques	90
10. Reconnaissance automatique des arythmies : Etat de l'art	91
10.1 Paramètres descriptifs d'un battement cardiaque.....	91
10.2 Algorithmes d'apprentissage et de décision.....	92
10.3 Problèmes liés à la reconnaissance automatique des arythmies	93
11. Conclusion	93
Chapitre4: Expérimentation, Résultats et discussion	
1. Introduction	94
2. Description de la base de données Arrhythmia	95
3. Traitements effectués sur la base de données :.....	96

3.1 L'évaluateur d'attributs CFS (Correlation-based Feature Selector):.....	96
3.2 La recherche Heuristique « Best-First » :.....	96
3.2.1 WEKA 3.6 : le logiciel d'extraction de données.....	97
4. Choix de L'algorithme :	98
4.1 Algorithme eclass0:.....	99
4.2 Description de l'algorithme :.....	100
4.2.1 Phase de classification :.....	100
4.2.2 Phase de mise à jour :.....	100
4.3 La boîte à outils IRooTlab :	101
5. Résultats Obtenus :	103
5.1 mesures de performances :	103
5.2 Expérimentation:	104
5.3 Comparaison des résultats avec l'état de l'art :	105
5.4 Résultats obtenus par l'algorithme eclass dans d'autres études dans le domaine médical :.....	106
6. Conclusion :.....	107
Conclusion générale	108
Références Bibliographiques:.....	109

Liste des illustrations :

Liste des figures :

Figure I-1 : Exemple d'inférence floue sur un système à trois règles.....	20
Figure I-2 : Système d'inférence floue d'ordre zéro.....	22
Figure I-3 : Système d'inférence floue d'ordre un.....	23
Figure II-1 : Classification basée-prototype contre classification basée-moyenne.....	33
Figure II-2 : Potentiel de deux échantillons de données d'ensemble des données du PIMA ...	35
Figure II-3 : Évolution de limite de la classification dans eClass1-MISO.....	38
Figure II-4 : L'évolution d'âge (données ionosphère).	41
Figure II- 5 : Bonnet EEG dans sa forme physique monté sur le cuir chevelu d'un sujet.....	59
Figure II-6 : Modèle général du système eClass.....	60
Figure II-7 : Pioneer 3DX.	64
Figure II-8 : L'enceinte expérimentale bureau avec 8 coins	64
Figure II-9 : Sunspot (sur une main et attaché au bras, la taille 63x38x25mm).....	65
Figure II-10 : Capteur Porcupine v2.5 avec capteur modulaire et boîtiers en plastique	65
Figure III-1 : Principales veines et artères.....	68
Figure III-2 : schéma de la circulation sanguine.....	68
Figure III-3 : Anatomie du cœur.....	69
Figure III-4 : Vue postérieure du cœur.....	69
Figure III-5 : Grande et petite circulation sanguine.....	70
Figure III-6 : Les différentes valves du cœur	70
Figure III-7 : Le cycle cardiaque.....	71
Figure III-8 : Parcours de l'onde électrique et phases de la contraction cardiaque	73
Figure III-9 : Electrophysiologie cellulaire	74
Figure III-10 : Triangle D'Einthoven.....	75

Figure III-11 : Dérivations unipolaires.....	76
Figure III-12 : Dérivations thoraciques.....	76
Figure III-13 : Le Holter.....	77
Figure III-14 : ECG normal.....	79
Figure III-15 : Rythme sinusal, segment de l'enregistrement AHA : 0201.....	80
Figure III-16: Exemple d'extrasystole ventriculaire.....	81
Figure III-17: Exemple de Bradycardie sinusale.....	81
Figure III-18: Exemple de Bradycardie jonctionnelle.....	82
Figure III-19: Tachycardie sinusale.....	82
Figure III-20: Comparaison entre un ECG normal, ECG d'une Tachycardie Ventriculaire et ECG d'une fibrillation ventriculaire.....	83
Figure III-21 : Emplacement des foyers ectopiques.....	84
Figure III-22 : Exemple de rythme Bigéminisme.....	85
Figure III-23 : Exemple de rythme Trigéminisme.....	85
Figure III-24 : Comparaison entre un rythme régulier et une fibrillation auriculaire.....	86
Figure III-24 : Comparaison entre un rythme régulier et une fibrillation ventriculaire.....	87
Figure III-25 : Exemple de Bloc de Branche Gauche	88
Figure III-26: Exemple de Bloc de Branche Droite	88
Figure III-27 : Signal électrocardiographique perturbé par le secteur.	89
Figure III-28 : Bruit dû au mauvais contact électrode-peau	89
Figure III-29 : Ondulation de la ligne de base	90
Figure III-30 : Exemple de bruits musculaires sur un segment de l'enregistrement MITDB...91	91
Figure IV-1 : interface du logiciel Weka 3.6.11.....	97
Figure IV-2 : interface de Weka 3.6.11 pour la sélection d'attributs.....	98
Figure IV-3 : les différents concepts et classifieurs proposés dans la boîte à outils IRootLab.....	102
Figure IV-4 : vitrine de démonstrations disponibles sur IRootLab.....	102
Figure IV-5 : Schéma de la classification en-ligne.....	103

Figure IV-6 : représentation de deux paramètres à l'instant $K=38$105

Liste des tableaux :

Tableau II-1: Les résultats de classification pour les systèmes IM-HMM-D et DM-HMM-D...60

Tableau II-2 : Résultats de l'utilisation d'eClass sur WBCD par rapport aux neuf caractéristiques...61

Tableau II-3 : Notes de fonction et de classement pour les deux modèles.....61

Tableau II-4 : Précision des deux modèles utilisant différents sous-ensembles de caractéristiques....62

Table IV-1 : Caractéristiques de la base de données UCI Arrhythmia.....95

Table IV-2 : performances des différents classifieurs appliqués.....104

Table IV-3 : performances de différentes méthodes utilisées dans la reconnaissance d'arythmies cardiaques.....106

Table IV-4 : performances de la méthode eclass dans le domaine du diagnostic médical....106

Introduction générale

Les maladies cardiovasculaires constituent un problème majeur de santé publique. Elles viennent en tête des causes médicales de décès en Algérie. En effet, près de la moitié des décès dans le pays (44.5%) sont dus aux maladies cardiovasculaires. Ce chiffre alarmant a été révélé par le Pr. Djamel Eddine Nibouche, en marge des journées de formation continue à l'école de formation paramédicale de la ville de Jijel, le 26 juin 2012. Les facteurs de risques sont multiples : tabac, sédentarité, obésité, hypertension artérielle, diabète et facteurs génétiques [Algerie1, 2012].

Le cœur, organe central du système cardiovasculaire, peut être affecté par de nombreuses pathologies qui peuvent être soit bénignes, comme certaines tachycardies par exemple, soit très sérieuses, comme l'infarctus du myocarde qui cause 10% de décès dans le monde. En raison de la gravité du problème, le suivi des patients à risque devient primordial. Les arythmies mineures informent le médecin sur l'état cardiaque du patient. Elles doivent être détectées pour prévenir notamment une dégénérescence possible en arythmies sévères.

Avec l'évolution des techniques, les médecins disposent aujourd'hui d'outils performants pour observer le fonctionnement du cœur et poser leur diagnostic. Parmi les examens cardiologiques possibles, l'électrocardiogramme (ECG) est l'examen le plus couramment effectué, car il est rapide à mettre en place, peu coûteux et surtout non contraignant pour le patient.

Une manière de détecter les troubles cardiaques consiste à recueillir le signal électrique cardiaque (ECG) par des capteurs, puis l'analyser. Cette analyse présente des enjeux à la fois pratiques et théoriques, pour les recherches actuelles en reconnaissance de formes, en apprentissage artificiel et en médecine. L'ECG est très souvent complété par un examen similaire d'une durée de 24 heures appelé "Holter", examen au cours duquel le patient peut exécuter ses occupations habituelles. Le principal avantage de l'enregistrement Holter par rapport à l'ECG de courte durée est qu'il permet la détection d'événements sporadiques qui n'interviennent pas nécessairement au cours des quelques secondes d'enregistrement ECG, réalisé en milieu hospitalier, lorsque le patient est au repos.

L'analyse de tels enregistrements nécessite l'utilisation d'outils de lecture automatique du signal, car la quantité d'informations enregistrées en 24 heures est très importante : elle correspond à environ 100 000 battements cardiaques sur 2 ou 3 voies d'enregistrements.

Ces outils de lecture doivent permettre le repérage d'informations pertinentes, d'éventuelles arythmies ainsi que leur fréquence d'occurrence. Cette reconnaissance automatique est rendue

possible grâce aux outils de traitement du signal ainsi qu'aux techniques d'apprentissage [Zidelmal, 2012].

L'objectif poursuivi à travers ce mémoire est de proposer de nouvelles méthodes en ligne, pour la reconnaissance d'arythmies cardiaques, afin d'aider le médecin à lire les enregistrements de longue durée et de déceler d'éventuelles arythmies en temps réel.

Une technique de classification relativement nouvelle, appelé eClass, a été introduite récemment par **Xydeas et al. [2005]**, afin de classer les signaux électroencéphalogrammes (EEG) en deux catégories douleur / non douleur. C'est une méthode récursive, non itérative, fait qui la rend appropriée pour les applications en temps réel, en raison de la faible complexité des calculs et du petit nombre de variables. Elle est basée sur l'algorithme de classification floue en ligne, pour un partitionnement du domaine d'entrée (Paramètres) et pour générer la base de règles. Le processus de défuzzification est celui du " winner-takes - all" pour mapper l'espace de données en entrée vers une étiquette de classe.

Cet algorithme permet aussi de préserver la transparence et l'interprétation des règles. Dans [Angelov et al., 2007b], différentes architectures possibles sont également considérées. En général, il s'agit d'un système très flexible qui peut traiter les données dont la classe n'est pas du tout connue (apprentissage non supervisé) ou connue (l'apprentissage semi-supervisé).

Ce mémoire de magistère comporte quatre chapitres :

- Le premier chapitre est une introduction aux principes de l'apprentissage artificiel en ligne ;
- Le deuxième chapitre présente les fondements des classifieurs évolutifs en général, puis un état de l'art de l'algorithme que nous allons appliquer pour la reconnaissance d'arythmies cardiaques (eClass) ;
- Le chapitre 3 est consacré à la présentation générale du fonctionnement du système cardiovasculaire, puis à celle du principe de l'électrocardiogramme (ECG) ainsi qu'une brève description des différentes anomalies et arythmies cardiaques ;
- Le dernier chapitre présente les différents résultats de classification des battements (normaux ou anormaux) obtenus après application de l'algorithme, ainsi que l'interprétation de ces résultats.

Chapitre I :

Principes de l'apprentissage en-Ligne

Principes de l'apprentissage en ligne

1. Introduction

Depuis bientôt un demi-siècle, les chercheurs en intelligence artificielle travaillent à programmer des machines capables d'effectuer des tâches qui requièrent de l'intelligence. Nous citerons *l'aide à la décision*, par exemple l'aide au diagnostic médical (notre domaine d'application) ; *la reconnaissance de formes*, comme la reconnaissance de la parole ou la vision artificielle ; *le contrôle de processus* tel que la conduite de procédés industriels ; *la prédiction*, telle la prédiction de consommation électrique ou la prédiction de cours boursiers; *la conduite de robots*, y compris d'équipes de robots comme dans la RoboCup; *l'exploration de grandes bases de données* (appelée aussi fouille de données) [Cornuéjols et Miclet, 2002]. Chacune de ces tâches et bien d'autres, ont stimulé l'inventivité des chercheurs et donné lieu à de nombreuses réalisations impressionnantes. Cependant, programmer des machines capables de s'adapter à toutes les situations et éventuellement d'évoluer en fonction de nouvelles contraintes est difficile. L'enjeu est de contourner cette difficulté en dotant les machines de capacités d'apprentissages, leur permettant de tirer profit de l'expérience d'experts humains, créant ainsi des systèmes autonomes.

En même temps, Le XXI^e siècle nous confronte à une série de nouveaux défis qui exigent des approches entièrement nouvelles. Comme a si bien dit John Naisbitt " aujourd'hui, nous sommes noyés dans l'information, mais affamés de connaissances " [Naisbitt, 1988]. Nous sommes au milieu d'une révolution de l'information et témoins d'une croissance exponentielle de la quantité, et la vitesse d'apparition de nouvelles informations par les utilisateurs de l'Internet, les consommateurs, les capteurs dans les procédés industriels, de systèmes autonomes etc.

Il est rapporté que chaque année plus de 1 Exabyte (= 10^{18} octets) de données sont générées dans le monde entier, la plupart de celles-ci sous forme numérique. Toshiba a récemment inventé l'expression « de l'obésité numérique » pour illustrer la quantité toujours croissante de données qui sont générées, transmises et consommées par les utilisateurs. Dans cet océan de données, les informations et connaissances utiles sont très souvent difficiles à extraire sous une forme claire, complète et intelligible.

Les nouveaux défis qui ne peuvent pas bien être traités avec succès par les techniques existantes, en particulier dans leur complexité et interconnexion, peuvent être résumés comme suit :

1. faire face à d'énormes quantités de données ;
2. traiter les données en streaming en ligne et en temps réel ;
3. s'adapter à l'environnement changeant et au modèle de données de manière autonome;
4. être efficace dans le calcul (c'est à dire, utiliser des approches récursives, en un seul passage, non itératives) ;
5. préserver l'intelligibilité et la transparence dans un sens dynamique.

La disponibilité de méthodes pratiques à utiliser et efficaces, des algorithmes, des techniques et des outils qui peuvent aider à extraire des connaissances à partir des données [Martin, 2005] est une demande pressante au niveau individuel et collectif, surtout si cela peut se faire en ligne, en temps réel.

Dans cette section, nous définirons précisément les différentes catégories de systèmes d'apprentissage qui existent, et en particulier les types de systèmes qui peuvent convenir à notre contexte applicatif.

2. Modèles intelligents

Il existe beaucoup de définitions pour l'Intelligence Artificielle, et c'est souvent un objet de débats et de malentendus. Mais en général, l'intelligence artificielle est la recherche de moyens susceptibles de doter les systèmes informatiques de capacités intellectuelles comparables à celles des êtres humains ; c'est le domaine de développement de modèles dits intelligents.

La tentative de comprendre l'intelligence et la construction de modèles de substitution du cerveau et de l'esprit, a été un problème central dans la philosophie, dès les premiers écrits de l'Inde et de la Grèce. L'avènement de l'ordinateur numérique dans les années 1950 en a fait une préoccupation centrale des informaticiens comme Turing [1950]. Le développement parallèle de la théorie de calcul (par John Von Neumann ; Alan Turing ; Emil Post ; Alonzo Church ; Steven Kleene ; Markov et autres) a fourni un nouvel ensemble d'outils de travail pour aborder ce problème par l'analyse, la conception et l'évaluation des ordinateurs et des

programmes, qui montrent les aspects de comportement intelligent comme la capacité à reconnaître et à classer des modèles; de raisonner à partir de prémices aux conclusions logiques et d'apprendre à partir de ses expériences.

Il est souvent suggéré que deux grandes approches ont émergé, l'intelligence artificielle symbolique (SAI) et l'IA (numérique) des réseaux de neurones artificiels (NANN ou réseaux connexionnistes). Certains auteurs tels que **Norman(1986) et Schneider(1987)** ont même suggéré qu'ils sont fondamentalement et peut-être irrémédiablement différents. En effet, c'est cette dichotomie apparente entre les deux approches de la modélisation de la cognition et l'ingénierie des systèmes intelligents qui est responsable de l'intérêt actuel pour les architectures de calcul, pour l'intégration de neurones et les processus symboliques ; ce qui soulève les questions suivantes: c'est quoi L'IA symbolique? C'est quoi L'IA numérique?

2.1 L'IA Symbolique :

Il existe de nombreuses approches différentes pour créer l'intelligence artificielle dans les ordinateurs et les systèmes intelligents. Depuis la création du champ de l'intelligence artificielle (IA), les approches symboliques ont persisté. Ces dernières représentent les problèmes comme des propositions logiques entre objets et utilisent les règles de la logique pour effectuer des déductions. On pense que par la représentation de ces relations et leur raisonnement, nous pouvons rendre les ordinateurs capables de réaliser des exploits mentaux, similaires à ceux des êtres humains. Cela diffère des techniques non-symboliques telles que les réseaux de neurones où, souvent, on ne tient pas compte de la façon dont un problème est représenté ou résolu tant qu'il est résolu.

L'IA symbolique est fondée sur la modélisation du raisonnement logique, sur la représentation et la manipulation de la connaissance par des symboles formels. Approche top-down, s'appuyant sur l'intersection entre logiques philosophiques et mathématiques (computational logic).

Un premier courant de l'approche symbolique est directement issu de la logique. Dans la logique propositionnelle, les symboles représentent des propositions et des connecteurs (ET, OU, NON, ...). La manipulation des symboles s'appuie, par exemple dans la logique déductive, sur un ensemble de règles permettant, à partir de propositions vraies, d'engendrer d'autres propositions vraies. Dans la théorie des modèles, une proposition est une représentation d'un état de choses du monde réel, elle est vraie si cet état est réalisé dans le

monde réel, sinon elle est fausse. La théorie de la démonstration ne s'intéresse pas aux relations entre les propositions et les choses qu'elles représentent, mais seulement aux relations entre les propositions elles-mêmes. Pour le philosophe français Descartes (1596-1650) et le philosophe et mathématicien allemand Leibniz (1646-1716), les symboles sont des idées combinées par des règles.

L'ordinateur apparut au milieu du XX^{ème} siècle comme un instrument permettant d'actualiser des systèmes de logiques formelles: Les symboles sont des chaînes binaires (de 0 et de 1, correspondant à des phénomènes physiques à deux états) rangées en mémoire. Un programme n'est qu'une suite de tels symboles, mais interprétés par la machine comme des opérations à exécuter sur d'autres symboles (le code et les datas). Comme en logique formelle, ces programmes ne s'occupent pas de sémantique. Mais à la différence des systèmes formels abstraits, un ordinateur est une machine physique réalisant des opérations concrètes. En particulier, l'intelligence artificielle (IA), née en 1956, montra que des heuristiques, comme la suppression de voies de recherches non prometteuses, pouvaient être plus utiles que les algorithmes garantissant le résultat en un nombre fini de pas, mais souvent inefficaces pour des problèmes complexes. L'IA professait que des programmes étaient capables de simuler n'importe quel comportement intelligent. Pour l'IA, les causes des comportements intentionnels résident dans des représentations internes instanciées physiquement dans le cerveau, sous forme de symboles. Ceux-ci présentent deux faces: L'une, matérielle, (état d'activité d'un groupe de neurones) et l'autre, abstraite, qui est une représentation. Le traitement des symboles correspond à un niveau syntaxique. Les comportements résultent des états représentationnels et de l'architecture fonctionnelle de l'organisme.

Un deuxième courant de l'approche symbolique est issu de la linguistique: Le linguiste américain Chomsky (né en 1928), élaborait les grammaires génératives (sorte d'automates capables d'engendrer des ensembles infinis de phrases au moyen de règles récursives), fort différentes des grammaires à états finis employées jusqu'alors. Il critiqua le béhaviorisme, impuissant à expliquer comment un locuteur pouvait comprendre et produire des phrases entièrement nouvelles.

Ces deux courants, le logique et le linguistique, de l'approche symbolique, conduisent tous deux à la conclusion qu'il serait possible de modéliser la cognition au moyen de programmes d'ordinateur. Mais la cognition humaine réside-t-elle seulement dans la

manipulation de symboles ? Il appartient aux neurosciences et au connexionnisme d'apporter un éclairage différent en attirant l'attention sur la très importante notion de réseau.

2.2 L'IA Numérique (Réseaux Connexionnistes) :

Dans les années soixante et soixante-dix, l'approche symbolique prévalait dans le domaine de l'IA et celui de la psychologie, l'approche par des réseaux neuronaux ayant été stoppée par **Papert et Minsky** en **1969**. Mais l'approche symbolique commençait à montrer ses limites, en particulier quant à sa robustesse défectueuse et son manque de souplesse, qui la firent échouer dans les tâches d'apprentissage et de reconnaissance de formes. Jusqu'en 1980, des recherches furent menées pour améliorer les systèmes d'IA en ce sens, puis à partir de cette date on recommença à s'intéresser aux réseaux: il s'agissait de remplacer les symboles à grain large par des micro-symboles à grain fin, de remplacer les structures linéaires et déterministes par de grands réseaux à comportements statistiques et à représentations distribuées.

Dans les années soixante, le connexionnisme avait défini la cognition comme une propriété émergente des nombreuses interactions d'un réseau neuronal dans lequel l'information est traitée de façon parallèle. Les systèmes symboliques et connexionnistes sont tous deux computationnels, mais dans des sens différents: pour le premier, le calcul est un ensemble de manipulations symboliques obéissant à des règles écrites dans un programme, alors que le deuxième s'intéresse aux processus causaux, réglant l'échange d'informations au sein d'un réseau, mais sans recourir ni aux symboles ni à des règles prédéfinies. En 1969, Papert et Minsky, avec leur ouvrage *Perceptrons*, donnèrent un coup d'arrêt à la recherche dans le domaine: on savait déjà que des problèmes non linéairement séparables, tels que le XOR logique, ne pouvaient pas être implémentés dans un réseau à deux couches et qu'une couche supplémentaire, dite couche cachée, était nécessaire. Mais **Papert et Minsky (1969)** montrèrent qu'aucun apprentissage d'un réseau multicouche n'était garanti de converger vers une solution en un nombre fini de passes et donc, même avec des réseaux de grande taille, il était vain de poursuivre les recherches. Cependant des scientifiques tels qu'**Anderson (1972), Kohonen (1972) et Grossberg (1976)** continuèrent la recherche dans le domaine avec des moyens limités. C'est à Hopfield et Rumelhart que l'on doit le regain d'intérêt pour les méthodes connexionnistes grâce, en particulier, à la méthode de la "rétropropagation de l'erreur" qui permet à un réseau de neurones multicouche de résoudre des problèmes non linéairement séparables. Simultanément, l'approche symbolique montrait ses limites, en

particulier quant à la fragilité de ses procédures et à son incapacité à résoudre des problèmes, simples pour le vivant, et très complexes pour une machine (comme la reconnaissance de forme). De plus, les neurosciences voyaient dans le connexionnisme une modélisation efficace, leur permettant de mieux étudier et comprendre le système nerveux, le cerveau et donc la cognition.

L'IA Numérique vise l'émergence de phénomènes relevant de la conscience ou de l'intelligence, à partir d'un agencement de mécanismes élémentaires (réaction simple à une perception), inspirés du vivant. Approche Bottom-up. Réseaux de neurones ; Systèmes d'agents réactifs (fourmis artificielles) ; automates cellulaires ...

2.3 Besoin en apprentissage :

Afin de construire un modèle intelligent, il faut trouver un moyen de le doter d'une certaine intelligence, pour cela on a recours à ce qu'on appelle l'apprentissage automatique.

Un des champs d'étude de l'intelligence artificielle est la discipline scientifique concernée par le développement, l'analyse et l'implémentation de méthodes automatisables qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.

L'apprentissage artificiel est défini par **Cornuéjols et Miclet [2002]**, comme une notion englobant toute méthode permettant de construire un modèle réel à partir d'un ensemble de données, soit en améliorant un modèle partiel (ou moins général), soit en créant complètement le modèle.

3. Les modes d'apprentissage artificiel

L'apprentissage automatique est divisé en trois modes:

- apprentissage supervisé ;
- apprentissage non supervisé ;
- apprentissage semi-supervisé (par renforcement).

3.1 Apprentissage supervisé

En apprentissage supervisé, on cherche à prédire la catégorie ou la valeur d'individus inconnus à partir d'un ensemble de référence, pour lesquels on connaît la vraie classe ou valeur. Cette catégorie d'appartenance, ou valeur associée, étant renseignée par un expert, on parle alors d'apprentissage supervisé.

3.2 Apprentissage non-supervisé

Au contraire, en apprentissage non-supervisé, on ne connaît pas la classe ou la valeur des individus de notre ensemble d'entraînement. L'apprentissage non-supervisé consiste à chercher une structure ou des motifs dans les données. L'objectif est d'obtenir un partitionnement des données en ensembles d'individus ayant des caractéristiques assez similaires entre eux, mais relativement différentes des autres individus.

3.3 L'apprentissage par renforcement

La particularité et la difficulté du cadre de l'apprentissage par renforcement sont que les décisions prises par l'algorithme influent sur l'environnement et les observations futures. L'exemple typique est celui d'un robot autonome qui évolue et effectue des actions dans un environnement totalement inconnu initialement. Il doit constamment apprendre de ses erreurs et succès passés, et décider de la meilleure politique à appliquer pour choisir sa prochaine action.

Notre contexte applicatif correspond donc à de l'apprentissage semi-supervisé puisque le système doit être en mesure de se corriger lui-même, on ne définit pas le nombre de classes ni leurs étiquettes, elles doivent être générées automatiquement par le système.

4 Différents problèmes en apprentissage artificiel

Les problèmes d'apprentissage peuvent être divisés en plusieurs catégories, selon le mode d'apprentissage :

4.1 Problèmes en apprentissage supervisé

On peut les subdiviser en 3 catégories :

4.1.1 Classification

Un problème de classification est un problème où l'on cherche à prédire la catégorie, ou classe, des individus. En classification supervisée, on dispose d'un ensemble d'apprentissage

composé d'individus pour lesquels on connaît d'une part, les valeurs de leurs caractéristiques, et d'autre part, leur catégorie d'appartenance. On cherche ensuite à ranger les individus inconnus dans les différentes catégories à l'aide des valeurs de leurs caractéristiques.

4.1.2 Régression

Un problème de régression est un problème où l'on cherche à prédire la valeur d'une variable inconnue en fonction d'autres variables descriptives. En régression, on cherche donc à modéliser la relation entre cette variable inconnue et les variables d'entrées.

4.1.3 Séries temporelles

Dans les problèmes de séries temporelles, il s'agit typiquement de prédire les valeurs futures d'une certaine quantité connaissant ses valeurs passées ainsi que d'autres informations. Par exemple le rendement d'une action en bourse. . . Une différence importante avec les problèmes de régression ou de classification est que les données suivent typiquement une distribution non stationnaire.

4.2 Problèmes en apprentissage non-supervisé :

Il y a également 3 types de problèmes pour ce type d'apprentissage :

4.2.1 Estimation de densité

Dans un problème d'estimation de densité, on cherche à modéliser convenablement la distribution des données. L'estimateur obtenu $f(x)$ doit pouvoir donner un bon estimé de la densité de probabilité à un point de test X issu de la même distribution (inconnue) que les données d'apprentissage.

4.2.2 Partitionnement

Le problème du partitionnement « clustering » est le pendant non supervisé de la classification. Un algorithme de partitionnement tente de partitionner l'espace d'entrée en un certain nombre de "classes", en se basant sur un ensemble d'apprentissage fini, ne contenant aucune information de classe explicite. Les critères utilisés pour décider si deux points devraient appartenir à la même classe ou à des classes différentes, sont spécifiques à chaque algorithme, mais sont très souvent liés à une mesure de distance entre points.

4.2.3 Réduction de dimensionnalité

Le but d'un algorithme de réduction de dimensionnalité est de parvenir à "résumer" l'information présente dans les coordonnées d'un point en haute dimension ($x \in R^n, n \text{ grand}$) par un nombre plus réduit de caractéristiques ($y = f(x), y \in R^m, m > n$). Le but

espéré est de préserver l'information "importante", de la mettre en évidence, en la dissociant du bruit, et possiblement, de révéler une structure sous-jacente qui ne serait pas immédiatement apparente dans les données d'origine en haute dimension. L'exemple le plus classique d'algorithme de réduction de dimensionnalité est l'Analyse en Composantes Principales (ACP).

Notre contexte applicatif correspond à un problème de classification, puisque nous cherchons à prédire la classe d'appartenance d'arythmies inconnues.

5 Relations entre les techniques d'apprentissage

Bien entendu, les frontières entre les tâches présentées sont souples. Ainsi nous appliquons couramment, et avec succès, des algorithmes conçus pour faire de la régression à des problèmes de classification, ou bien on estime des densités dans le but de faire de la classification comme dans le contexte de la présente étude.

Notons qu'une bonne estimation de densité permet en théorie, de prendre la décision optimale concernant un problème de classification ou de régression.

6 Apprentissage hors-ligne et Apprentissage en-ligne

Un système de reconnaissance peut être utilisé soit de manière en-ligne soit de manière hors-ligne [Almaksour, 2011]. On parle bien ici de la manière dont est utilisé le système et non pas de la manière dont est réalisé l'apprentissage.

6.1 Apprentissage hors-ligne

Le mode de fonctionnement hors-ligne sépare les phases d'apprentissage et d'utilisation. Dans un premier temps, le système est entraîné de manière statique ou incrémentale à partir d'un ensemble d'entraînement. Ensuite, dans un deuxième temps, le système préalablement appris, est utilisé.

6.2 Apprentissage en-ligne

Le mode de fonctionnement en-ligne ne sépare pas l'apprentissage et l'utilisation du système, les deux tâches sont effectuées de manière conjointe. Tout au long de son utilisation, le

système continue d'apprendre dès qu'une nouvelle donnée est disponible afin d'améliorer ses performances.

L'avantage des systèmes en-ligne par rapport aux systèmes hors-ligne est qu'ils vont pouvoir s'ajuster très précisément à l'utilisateur final, en continuant à apprendre tout au long de leur utilisation. L'ensemble d'apprentissage initial n'a plus besoin d'être aussi conséquent et diversifié puis que le système s'adaptera pendant son utilisation.

L'apprentissage d'un système fonctionnant en-ligne est le plus souvent réalisé incrémentalement, même s'il pourrait également être fait de manière statique, en reconstruisant tout le système (en stockant toutes les données).

7 Apprentissage statique et apprentissage incrémental

L'entraînement d'un système d'apprentissage peut se faire de deux manières : de manière statique ou de manière incrémentale. Nous allons voir les principales différences entre ces deux approches [Almaksour, 2011].

7.1 Apprentissage statique

En apprentissage statique (*batch learning*), le jeu de données d'entraînement est prédéfini et disponible lors de l'entraînement du système. L'algorithme d'apprentissage utilise l'ensemble du jeu de données pour minimiser le taux d'erreur.

7.2 Apprentissage incrémental

En apprentissage incrémental, le jeu de données n'est pas forcément disponible dès le début de l'entraînement. Les données sont introduites au fur et à mesure et le système doit être capable d'apprendre à partir de chacun de ces exemples séparément. Le système doit donc être en mesure de se modifier et d'ajuster ses paramètres après l'observation de chaque exemple, pour apprendre à partir de celui-ci ; mais néanmoins sans oublier la connaissance acquise à partir des exemples précédents.

Ce mode d'apprentissage est utilisé soit lorsque le jeu de données est trop grand pour être utilisé en une seule fois ; soit lorsque l'ensemble d'apprentissage n'est pas disponible dans son intégralité et que les données d'entraînement arrivent de manière incrémentale.

L'apprentissage incrémental peut facilement être réalisé en-ligne et permet d'utiliser le système pendant son apprentissage, ou formulé autrement, d'améliorer le système tout au long de son utilisation.

Nous souhaitons ici que notre système apprenne en-ligne et qu'il soit utilisable en temps réel, il faut donc que l'apprentissage soit incrémental.

8 Système adaptatif et système évolutif

Les systèmes d'apprentissage incrémental sont donc dynamiques car ils apprennent au fur et à mesure que les données arrivent, contrairement aux systèmes d'apprentissage classique, dits statiques.

Néanmoins, ces systèmes incrémentaux peuvent encore être séparés en deux catégories : les systèmes adaptatifs et les systèmes évolutifs [Lughoferet Angelov, 2011].

8.1 Système adaptatif

On appelle systèmes adaptatifs les systèmes d'apprentissage incrémental qui vont apprendre de manière incrémentale leurs paramètres mais dont la structure est fixe. Cet apprentissage des paramètres peut être considéré comme un algorithme « d'adaptation ». La structure de ces systèmes est initialisée au départ et reste ensuite inchangée pendant l'apprentissage.

8.2 Système évolutif

Les systèmes évolutifs sont une sous-catégorie des systèmes incrémentaux, capables de changer leur structure. Ils vont non seulement apprendre leurs paramètres de manière incrémentale, mais également modifier leur structure, comme par exemple lors de l'apparition d'une nouvelle classe.

Un système évolutif est donc bien plus flexible qu'un système adaptatif puisqu'il permet, en cours d'utilisation, d'ajouter des classes supplémentaires lorsque cela s'avère nécessaire.

Dans notre travail, nous souhaitons mettre en œuvre un classifieur évolutif en ligne.

9 Mémoire des données et mémoire des concepts

Un système d'apprentissage a besoin de mémoriser l'information apprise. Pour cela, il peut soit mémoriser les données, soit mémoriser les concepts extraits des données ; ou encore combiner les deux [Maloof et Michalski, 2004].

9.1 Mémoire des données

Mémoriser toutes les données est une méthode simple qui permet de mettre à jour l'ensemble d'apprentissage facilement. Cette méthode est par exemple utilisée par les classifieurs de type plus proches voisins. Elle peut également être utilisée pour reconstruire tout un système à l'arrivée de nouvelles données.

Cependant, cette méthode possède plusieurs inconvénients. Le nombre de données à mémoriser augmente tout au long de l'utilisation du système et cette méthode va nécessiter beaucoup d'espace mémoire. De plus, si l'on reconstruit tout le système, son apprentissage va être de plus en plus long et complexe, au fur et à mesure que la taille du jeu de données d'entraînement augmente.

Une approche dérivée, limitant ces inconvénients, consiste à sélectionner et maintenir un sous-ensemble des données d'apprentissage sur lequel reconstruire le système. Cette méthode est appelée mémoire partielle des données, à opposer à la mémoire complète des données.

9.2 Mémoire des concepts

Une autre méthode consiste non pas à mémoriser les données, mais plutôt à mémoriser les concepts, c'est-à-dire la connaissance extraite de ces données. Cette méthode est par exemple utilisée par les classifieurs de type réseau connexionnistes. La connaissance, le modèle, doivent alors être mis à jour à l'arrivée de nouveaux exemples, de manière à prendre en compte ces nouvelles informations, mais sans pour autant oublier celles issues des exemples précédents.

C'est ce qui nous intéresse dans notre travail, car dans la classification en ligne, on a besoin d'avoir un système avec un coût computationnel, le plus faible possible.

10 les principaux modèles incrémentaux évolutifs

Dans cette partie, nous allons présenter les principaux systèmes incrémentaux évolutifs.

Plusieurs systèmes à ensembles évolutifs ont été proposés dans la littérature, nous citons en particulier :

10.1 Cas de mono classifieurs évolutifs

10.1.1 Les Réseaux de neurones multicouche

Le Perceptron multicouche est un Classifieur linéaire de type réseau neuronal formel, organisé en plusieurs couches, au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement. Chaque couche est constituée d'un nombre variable de neurones, les neurones de la couche de sortie correspondant toujours aux sorties du système.

Le Perceptron a été inventé en 1957 par Frank Rosenblatt au Cornell Aeronautical Laboratory, inspiré par la théorie cognitive de Friedrich Hayek et celle de Donald Hebb. Dans cette première version le Perceptron était alors mono-couche et n'avait qu'une seule sortie à laquelle toutes les entrées sont connectées.

Les premiers réseaux de neurones n'étaient pas capables de résoudre des problèmes non linéaires ; cette limitation fut supprimée au travers de la rétropropagation¹ du gradient de l'erreur dans les systèmes multicouches, proposé par Paul J Werbos en 1984 et mis aux point deux années plus tard, en 1986 par David Rumelhart.

Dans le Perceptron multicouche à rétropropagation, les neurones d'une couche sont reliés à la totalité des neurones des couches adjacentes. Ces liaisons sont soumises à un coefficient altérant l'effet de l'information sur le neurone de destination. Ainsi, le poids de chacune de ces liaisons est l'élément clef du fonctionnement du réseau : la mise en place d'un Perceptron multicouche pour résoudre un problème, passant donc par la détermination des meilleurs poids applicables à chacune des connexions inter-neurales. Ici, cette détermination s'effectue au travers d'un algorithme de rétropropagation.

10.1.2 Machines à vecteurs supports (SVM) évolutives

Les machines à vecteurs supports (SVM) sont des systèmes de classification très puissants qui vont déterminer la séparatrice maximisant la marge entre les classes. Cette marge est définie

comme la distance entre la séparatrice et les individus les plus proches, appelés vecteurs supports.

Les premiers algorithmes d'apprentissage incrémental de machine à vecteurs supports ont été proposés dans [Syed et al., 1999] et [Ruping, 2001]. Cette approche vise à gérer l'ensemble de vecteurs supports de manière incrémentale. Lorsqu'une nouvelle donnée d'apprentissage est mal classée ou est à l'intérieur de la marge, la séparatrice est recalculée à partir des vecteurs supports et de ce nouveau point.

Le principal inconvénient des SVM incrémentales est leur difficulté à gérer l'ajout de nouvelles classes. Une classe récemment ajoutée ne possèdera que peu d'individus, et en maximisant la marge, une SVM aura tendance à isoler ces quelques points sans vraiment généraliser la représentation de cette nouvelle classe.

10.2 Cas de multi-classifieurs évolutifs

10.2.1 Systèmes à ensemble évolutifs

Les systèmes à ensemble sont composés d'un groupe de classifieurs faibles, dont les résultats sont ensuite combinés, le plus souvent par un système de vote majoritaire. Ces classifieurs faibles peuvent être construits avec un sous-ensemble de données d'apprentissage, de caractéristiques, de classes ou une combinaison des trois.

L'ajout de nouveaux classifieurs, à l'arrivée de nouvelles données, permet l'apprentissage incrémental et l'ajout de classes. Néanmoins, une attention particulière doit être portée à la diversité de cet ensemble de classifieurs afin que la combinaison soit représentative de toutes les données.

Nous citons quelques exemples de ce type de systèmes :

10.2.1.1 L'algorithme *Learn++*

L'algorithme *Learn++*, proposé par [Polikar et al., 2001], permet de construire un système à ensemble incrémental où de nouveaux classifieurs faibles sont créés à l'arrivée de nouvelles données.

L'implémentation proposée utilise des Perceptrons multi-couches (MLP) comme classifieurs faibles, mais d'autres choix sont également possibles. *Learn++* nécessite une mémoire partielle des données, afin de maintenir un ensemble d'apprentissage.

Cet ensemble d'apprentissage est ensuite échantillonné, suivant une distribution de probabilités, correspondant à la difficulté des individus, afin d'obtenir des sous-ensembles sur lesquels seront appris les classifieurs faibles. Ensuite, les résultats de l'ensemble de classifieurs faibles générés, sont combinés par une méthode de vote pondéré spécifique.

10.2.1.2 Growing Negative Correlation Learning (GNCL)

GNCL [Minku et al., 2009] est une méthode d'apprentissage d'ensemble de classifieurs, également basée sur des réseaux de neurones. GNCL est une méthode de bagging, et d'autres types de classifieurs faibles sont également utilisables.

Cette méthode vise à produire un ensemble de classifieurs faibles, le plus varié possible, en insérant une pénalité dans la fonction d'erreur de chaque classifieur faible. Cette pénalité négative de corrélation augmente lorsqu'un classifieur se « rapproche » d'un autre, assurant ainsi la diversification de l'ensemble en pénalisant les classifieurs corrélés.

10.2.2 Les systèmes multi agents

Les systèmes multi-agents cherchent à appréhender la coordination de processus autonomes. Pour Weiss (1999), un agent est une "entité computationnelle", comme un programme informatique ou un robot, qui peut être vu comme percevant et agissant de façon autonome sur son environnement. On peut parler d'autonomie parce que son comportement dépend au moins partiellement de son expérience. Un système multi-agents (SMA) est constitué d'un ensemble de processus informatiques se déroulant en même temps, donc de plusieurs agents vivant au même moment, partageant des ressources communes et communiquant entre eux. Le point clé des systèmes multi-agents réside dans la formalisation de la coordination entre les agents. La recherche sur les agents est ainsi une recherche sur :

- la décision -quels sont les mécanismes de la décision de l'agent? quelle est la relation entre les perceptions, les représentations et les actions des agents? Comment décomposent-ils leurs buts et tâches ? comment construisent-ils leurs représentations ?
- le contrôle - quelles sont les relations entre les agents? comment sont-ils coordonnés ? Cette coordination peut être décrite comme une coopération pour accomplir une tâche commune ou comme une négociation entre des agents ayant des intérêts différents.
- la communication - quels types de message s'envoient-ils? à quelle syntaxe obéissent ces messages? Différents protocoles sont proposés en fonction du type de coordination entre les agents.

Les systèmes multi-agents ont des applications dans le domaine de l'intelligence artificielle où ils permettent de réduire la complexité de la résolution d'un problème, en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents [Ferber, 1995]. On parle ainsi d'intelligence artificielle distribuée.

10.3 Systèmes d'inférence floue évolutifs

Dans un système d'inférence floue, un individu active plus ou moins des règles d'inférence. Les sorties de ces règles vont ensuite être combinées, en fonction de leur activation, pour prédire la classe de cet individu. Les systèmes d'inférence floue sont, sous certaines conditions, équivalents à des réseaux de neurones à fonction radiale [Hunt et al., 1996].

L'apprentissage incrémental évolutif d'un système d'inférence floue se fait par modification de ces règles d'inférence. En particulier, de nouvelles classes peuvent facilement être ajoutées en créant de nouvelles règles.

Quelques exemples de ce type de systèmes :

10.3.1 Exemples de SIF dans la littérature

10.3.1.1 FLEXible Fuzzy Inference System (FLEXFIS)

FLEXFIS est un système d'inférence floue de Takagi-Sugeno d'ordre un [Lughofer, 2008b], c'est-à-dire avec une structure de conclusion linéaire. L'idée principale de cet algorithme est d'utiliser les principes de quantification vectorielle, de manière incrémentale, pour mettre à jour les prémisses des règles d'inférence.

Le principal inconvénient de cette méthode de mise à jour des prémisses vient du fait qu'elle nécessite un seuil de distance. Ce seuil contrôle l'équilibre entre la mise à jour des clusters existants et la création de nouveaux. Ce paramètre doit être ajusté pour chaque problème de classification, ce qui est difficilement réalisable de manière incrémentale.

10.3.1.2 Evolving Fuzzy Neural Networks (EFuNN)

EFuNN est un système de classification floue, composé de règles associant une hypersphère de l'espace d'entrée à une hypersphère de l'espace de sortie [Kasabov, 2001]. L'auteur propose une application de cette méthode à la reconnaissance de phonèmes.

Dans ce système, une nouvelle règle est créée lorsqu'une nouvelle donnée d'apprentissage n'active pas de règle au-dessus d'un certain seuil. Une nouvelle règle est également créée

lorsqu'une donnée d'apprentissage active une règle, mais que l'erreur dépasse un autre seuil. Ces principes d'évolution dépendent fortement de seuils, qui sont des paramètres qu'il faut ajuster pour chaque problème.

10.3.1.3 Dynamic Evolving Neural-Fuzzy Inference System (DENFIS)

DENFIS est un système d'inférence floue de Takagi-Sugeno d'ordre un, où la fonction d'appartenance floue est la fonction triangulaire [Kasabov and Song, 2002]. L'auteur utilise ce système pour faire de la prédiction dynamique de séries temporelles.

L'évolution des prémices des règles est faite à l'aide d'une méthode de clustering évolutif (ECM), qui nécessite un paramètre spécifiant la taille des clusters. DENFIS partage donc l'inconvénient des systèmes précédents, c'est à dire l'ajustement d'un paramètre dépendant du problème. Les conclusions des règles sont apprises par la méthode des moindres carrés récursifs.

10.3.1.4 Evolving Takagi-Sugeno (eTS) et eClass (notre sujet)

eTS est un système d'inférence floue de Takagi-Sugeno d'ordre un, qui ajuste les règles d'inférences, pour suivre l'évolution du flux de données [Angelov and Filev, 2004]. eClass est une extension de eTS, appliquée au domaine de la classification [Angelov and Zhou, 2008]. L'ajustement du système se fait à plusieurs niveaux ; tout d'abord, les prémisses des règles existantes sont mises à jour à l'arrivée de nouvelles données, ensuite, de nouvelles règles sont créées si les nouvelles données ne correspondent pas aux prémices des règles existantes. Enfin, les paramètres des conclusions des règles d'inférence sont également mis à jour.

Il est important de préciser que la mise à jour des prémisses et la création de règles sont faites par une méthode de *clustering* basée sur la densité.

10.3.2 Principes des Systèmes d'inférence floue (SIF) incrémental évolutif

Dans cette partie, nous allons commencer par aborder les principes de la logique floue. Ensuite, nous étudierons les différentes structures de SIF. Enfin, nous verrons les différents niveaux de fonctionnement de l'apprentissage incrémental d'un SIF.

10.3.2.1 Théorie de la logique floue

La théorie de la logique floue a été formalisée dans [Zadeh, 1965].

a) Théorie des ensembles flous

En logique floue, l'appartenance d'un individu à un ensemble n'est pas binaire, mais floue. Cette appartenance n'est plus limitée aux deux valeurs {0; 1}, mais peut varier dans tout l'intervalle [0; 1].

Un individu peut ainsi appartenir, à différents degrés, à plusieurs ensembles flous. Différentes fonctions peuvent être utilisées comme fonction d'appartenance, les plus utilisées sont la fonction triangulaire [Kasabovet Song, 2002], la fonction trapézoïdale et la fonction gaussienne[AngelovetFilev, 2004] ; [Lughofer, 2008].

b) Inférence floue

Un système d'inférence floue est composé de règles structurées selon la forme suivante :

$$IF \text{ (prémisse) } THEN \text{ (conclusion)} \tag{I.1}$$

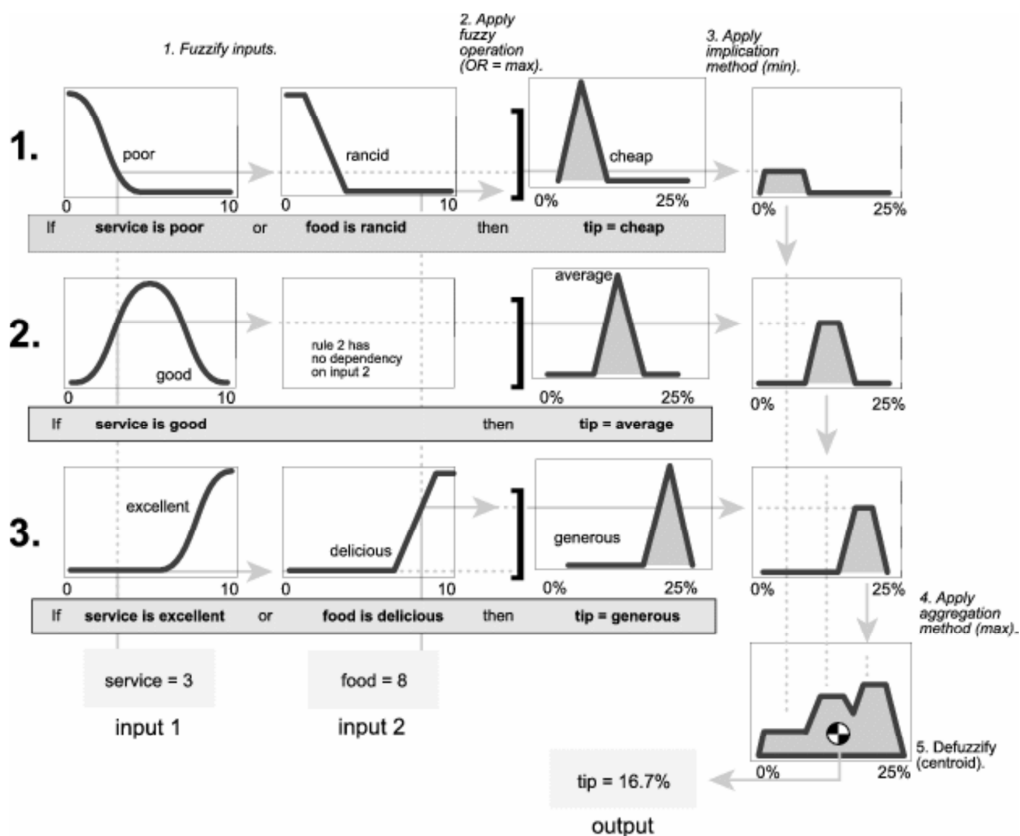


Figure I-1 : Exemple d'inférence floue sur un système à trois règles, avec deux entrées (service, food) et une sortie (tip) – source : mathworks.com

Le processus d'inférence floue peut être détaillé en cinq étapes (voir figure I.1) :

Fuzzyfication : les données « classiques » sont rendues floues;

Unification : les données floues, en entrée, sont unifiées avec les prémisses des règles ;

Implication : en fonction de leur activation, les règles produisent les sorties floues ;

Agrégation : les sorties floues des règles sont agrégées, pour produire la sortie floue du système ;

Défuzzyfication : la sortie floue est rendue « classique ».

La structure des prémisses et la structure des conclusions sont détaillées dans la section suivante.

10.3.2.2 Architecture d'un SIF

a) Structure des prémisses

La prémisses d'une règle d'inférence floue est une agrégation d'ensembles flous :

$$IF(x_1 \text{ is } A_1), \dots, (x_k \text{ is } A_k) THEN(\text{conclusion}) \quad (I.2)$$

où x_1, \dots, x_n sont les variables d'entrée et A_1, \dots, A_k sont les différents ensembles flous d'entrée.

La zone d'influence d'une règle floue est appelée le prototype de cette règle. Les prototypes des règles d'inférence peuvent être calculés à l'aide d'algorithmes de *clustering*. Un algorithme de regroupement basé sur un seuil de distance a l'inconvénient d'introduire un paramètre dépendant du problème [Lughofer, 2008], ce qui est évité avec une méthode basée sur la densité [Angelov et Filev, 2004].

Dans le cas de la fonction d'appartenance gaussienne, les prototypes sont de formes hyper elliptiques. Le prototype d'une règle est caractérisé par son centre $\mu = (\mu_1, \dots, \mu_n)$ et son rayon $\sigma = (\sigma_1, \dots, \sigma_n)$.

Le degré d'activation β d'une règle par un individu $x = (x_1, \dots, x_n)$ est l'union – opérateur produit – des degrés d'activation des différentes composantes de cet individu, selon les différents axes. Ils sont obtenus en appliquant la fonction d'appartenance, paramétrée par le rayon, à la distance entre les composantes de cet individu et celles du centre du prototype de la règle.

Pour la fonction d'appartenance gaussienne, le degré d'activation β est donné par la formule :

$$\beta(x) = \prod_{i=1}^n \exp\left(-\frac{\|x_i - \mu_i\|}{2\sigma_i}\right) \quad (\text{I.3})$$

b) Structure des conclusions

La sortie floue d'une règle est un vecteur regroupant les degrés d'appartenance aux k classes :

$$s^{(i)} = (s_1^{(i)}, \dots, s_k^{(i)}) \quad (\text{I.4})$$

La sortie floue \hat{y} du système est la somme des sorties floues des r règles pondérées par leur activation

$$\hat{y} = (s_i, \dots, s_k) = \sum_{i=1}^r \beta_i (s_1^{(i)}, \dots, s_k^{(i)}) \quad (\text{I.5})$$

La classe prédite \tilde{y} est celle avec le plus fort degré d'appartenance :

$$\tilde{y} = \operatorname{argmax}_{1 \leq i \leq k} \hat{y} = \operatorname{argmax}_{1 \leq i \leq k} (s_i, \dots, s_k) \quad (\text{I.6})$$

Les conclusions des règles, qui produisent leur sortie floue, peuvent être de deux types : d'ordre zéro ou d'ordre un.

Conclusion d'ordre zéro : Les SIF avec une structure de conclusion d'ordre zéro ont été proposés dans [Mamdani, 1977]. EFuNN [Kasabov, 2001] est un exemple de SIF de Mamdani. Dans un tel système, les règles sont de la forme :

$$IF \text{ (prémisse) THEN } \hat{y}^{(i)} \text{ is } (s_i, \dots, s_k) \quad (\text{I.7})$$

Un SIF de Mamdani va donc créer un partitionnement flou de l'espace d'entrée et relié à un partitionnement de l'espace de sortie. Le principal avantage des SIF d'ordre zéro est leur transparence : on peut facilement interpréter les règles produites en langage naturel.

La figure (I-2) représente un SIF d'ordre zéro sous la forme d'un réseau de neurones.

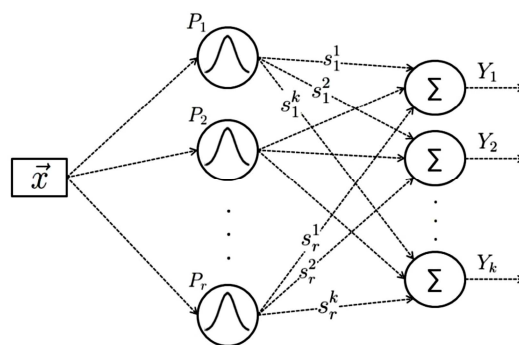


Figure I-2 : Système d'inférence floue d'ordre zéro. [Bouillon, 2012]

Pour chacune des r règles, la conclusion est un vecteur regroupant les degrés d'appartenance à chacune des k classes :

$$s^{(i)} = (s_1^{(i)}, \dots, s_k^{(i)}) \quad \text{Pour } 1 \leq i \leq k \quad (\text{I.8})$$

Conclusion d'ordre un : Les systèmes d'inférence floue, avec une structure de conclusion d'ordre un, ont été proposés dans [Takagi et Sugeno, 1985]. eTS [Angelov et Filev, 2004] est un exemple de SIF de Takagi-Sugeno.

Dans un tel système, les règles sont de la forme :

$$\mathbf{IF} \text{ (prémisse) } \mathbf{THEN} \hat{y}^{(i)} = (\Pi_1^{(i)}(x_1, \dots, x_n), \dots, \Pi_k^{(i)}(x_1, \dots, x_n)) \quad (\text{I.9})$$

Où $\Pi_1^{(i)}, \dots, \Pi_k^{(i)}$ sont des fonctions linéaires des variables d'entrées.

Les SIF d'ordre un permettent généralement d'obtenir une meilleure précision que les SIF d'ordre zéro, mais sont moins facilement interprétables.

La figure (I-3) représente un SIF d'ordre un sous la forme d'un réseau de neurones.

Pour chacune des r règles, la conclusion est une matrice regroupant les coefficients des fonctions linéaires $\Pi_1^{(i)}, \dots, \Pi_k^{(i)}$:

$$\Pi^{(i)} = \begin{pmatrix} \pi_{1,1}^{(i)} & \dots & \pi_{1,k}^{(i)} \\ \vdots & \vdots & \vdots \\ \pi_{n,1}^{(i)} & \dots & \pi_{n,k}^{(i)} \end{pmatrix} \quad \text{Pour } 1 \leq i \leq r \quad (\text{I.10})$$

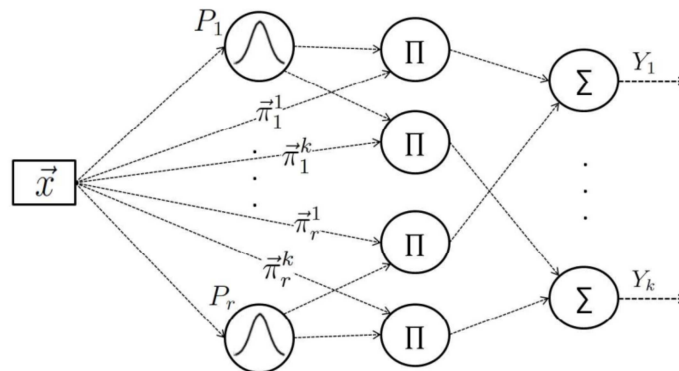


Figure I-3 : Système d'inférence floue d'ordre un. [Bouillon, 2012]

La sortie floue d'une règle est alors obtenue en multipliant x^t par la matrice de conclusion $\Pi^{(i)}$:

$$s^{(i)} = (s_1^{(i)}, \dots, s_k^{(i)}) = x^t \Pi^{(i)} \quad (\text{I.11})$$

10.3.3 Apprentissage incrémental d'un SIF

10.3.3.1 Évolution des prémisses

Lorsqu'une nouvelle donnée d'apprentissage arrive, il est nécessaire de faire évoluer les prémisses de certaines règles.

Ajustement des prototypes : Lorsqu'une nouvelle donnée arrive, il faut alors mettre à jour les prototypes des différentes règles en fonction de leur activation. Leur centre μ et leur rayon σ sont alors recalculés récursivement à l'aide des formules statistiques suivantes :

$$\mu_t = \frac{(\alpha_t - 1)}{\alpha_t} \mu_{t-1} + \frac{\beta_t}{\alpha_t} x_t \quad (\text{I.12})$$

$$\sigma_t = \frac{(\alpha_t - 1)}{\alpha_t} \sigma_{t-1} + \frac{\beta_t}{\alpha_t} (\mu_{t-1} - x_t) \quad (\text{I.13})$$

Où β_t est l'activation du prototype et $\alpha_t = \sum_{i=1}^t \beta_i$ est le cumul des activations de ce prototype à l'instant t .

Création de nouvelles règles : Si la nouvelle donnée ne correspond à aucun prototype de règle déjà existante, il est alors nécessaire de créer une nouvelle règle autour de cette donnée. Cette décision peut être prise sur différents critères :

- si l'activation de toutes les règles est inférieure à un certain seuil ;
- à l'aide d'un algorithme de *clustering* basé sur un seuil de distance [Kasabov et Song, 2002] ;
- à l'aide d'un algorithme de *clustering* basé sur la densité [Angelov et Filev, 2004].

Les deux premières solutions ont l'inconvénient d'introduire un paramètre qui est dépendant du problème et qui est difficilement estimable de manière incrémentale, nous préférons donc la troisième solution.

On peut notamment retenir l'algorithme RepStream [LühretLazarescu, 2009], basé sur une représentation des données sous forme de graphe, ainsi que l'algorithme eClustering qui procède par estimation récursive de la densité [Ramezani et al., 2008].

10.3.3.2 Évolution des conclusions

Lorsqu'une nouvelle donnée d'apprentissage arrive, il est également nécessaire de faire évoluer les conclusions de certaines règles. Cette optimisation est le plus souvent réalisée de manière supervisée à l'aide de l'algorithme des moindres carrés récurrents [Lughofer, 2008], mais peut aussi être réalisée de manière non supervisée par *clustering* [Kasabov, 2001].

Cette optimisation des conclusions par moindres carrés récurrents peut être réalisée de manière globale ou de manière locale. Il est montré dans [Almaksour, 2011] que les deux méthodes donnent des résultats similaires, mais que l'optimisation locale est de complexité plus faible. L'optimisation locale est préférable dans le cas d'une utilisation en-ligne où l'apprentissage est réalisé en arrière-plan.

Enfin, toujours de par sa plus faible complexité, l'optimisation locale des conclusions permet de travailler dans un espace d'entrée de plus grande dimension (nombre de caractéristiques plus important).

10.3.3.3 Stabilité de l'apprentissage

Lors de l'apprentissage incrémental d'un SIF, l'évolution des prémisses et des conclusions est réalisée de manière simultanée. Cette évolution des prémisses pendant l'optimisation des conclusions crée des instabilités [Lughofer, 2008]. En effet, lorsque les prototypes changent, les anciennes modifications apportées aux conclusions ne correspondent plus tout à fait aux nouveaux prototypes.

Une méthode pour limiter ces instabilités est de maintenir une mémoire partielle des données et de réaliser la mise à jour des prémisses de manière ponctuelle ; une seconde est de concentrer les mises à jour sur les individus mal ou difficilement classés [Almaksour et Anquetil, 2009].

10.3.3.4 Ajout de classes

L'ajout de classes est relativement facile dans un SIF. L'arrivée d'un individu d'une nouvelle classe va mettre à jour les prémisses de manière classique, ou créer une nouvelle règle si besoin, puis la nouvelle classe sera ajoutée aux règles activées.

11 Discussion

Si les SVM incrémentales sont très performantes, leur utilisation de manière évolutive est beaucoup moins aisée. L'algorithme de calcul des SVM cherche à séparer deux classes avec la plus grande marge, ce qui réduit la généralisation des classes récentes sous représentées.

Le principal inconvénient des méthodes à ensembles est le problème d'*outvoting* lorsque l'on introduit de nouvelles classes. Ce problème est dû au fait que les anciens classifieurs votent pour des individus appartenant à des classes sur lesquelles ils n'ont pas été entraînés. Cela va donc fortement défavoriser les dernières classes introduites.

Par contre Les systèmes d'inférence floue permettent généralement d'obtenir de bonnes performances, et se comportent bien à l'ajout de nouvelles classes grâce à leur représentation floue des ensembles. De plus, ces systèmes ne nécessitent pas de mémoire des instances, mais seulement une mémoire des concepts. Les systèmes d'inférence floue semblent donc bien correspondre à nos attentes.

12 Conclusion

L'intelligence artificielle est un domaine très vaste. Dans ce chapitre, nous avons vu les concepts importants en apprentissage artificiel ; évidemment, il reste d'autres concepts et d'autres techniques d'apprentissage, nous avons mis l'accent sur ce qui est essentiel pour notre travail.

Notre contexte applicatif correspond à une utilisation en ligne d'un système de classification semi supervisée. Si l'on veut que notre système soit utilisable sur un ordinateur standard, voire une tablette (le système ne doit pas être gourmand en espace mémoire ainsi qu'en calculs), il est indispensable que ce système ne soit pas entièrement reconstruit à l'arrivée de nouvelles données. Ce système doit donc être incrémental, c'est-à-dire satisfaire les critères suivants :

- pouvoir apprendre de nouvelles connaissances à partir de nouvelles données ;
- utiliser une mémoire des concepts (ne pas nécessiter le stockage de toute les anciennes données) ;
- préserver la connaissance préalablement acquise à partir des anciennes données, lors de l'apprentissage de nouvelles connaissances à partir de nouvelles données.

Ensuite, nous souhaitons que ce système soit évolutif et en particulier qu'il puisse :

- apprendre en partant de zéro (sans connaissance a priori) ;
- permettre l'ajout de classes ;
- prendre en considération la structure ouverte du modèle évolutif.

Chapitre II

Les classifieurs évolutifs :
Littérature et théories

Les classifieurs évolutifs flous : littérature et théorie

1. Introduction :

Au cours des dernières années, les observations montrent une utilisation accrue de la théorie des ensembles flous dans le domaine des diagnostics médicaux. Steimann [Steimann, 2001] a observé et analysé profondément la croissance rapide de leur utilisation dans ce domaine. En effet, la théorie des ensembles flous répond à l'incertitude non-statistique qui entoure les problèmes du domaine médical. Zadeh, l'instigateur et le créateur de la théorie des ensembles flous, a indiqué en 1969 que "...les ensembles flous sont beaucoup pour être appliqués dans le domaine des diagnostics médicaux..." [Zadeh, 1969]. Actuellement, les modèles flous ont augmenté de manière significative l'interprétabilité et la transparence du diagnostic médical.

Beaucoup d'approches dites intelligentes sont utilisées dans le diagnostic médical. L'inconvénient principal de ces approches qui ne sont pas basées sur les ensembles flous, est qu'ils ne fournissent aucune explication sur la façon dont leur résultat d'inférence a été acquis. Les ensembles flous au contraire, comblent cette lacune en soutenant des descriptions linguistiques et des expressions, comme indiqué par Kuncheva et Steimann [Kuncheva et Steimann, 1999]. En effet, Les systèmes flous produisent habituellement des règles interprétables par l'humain qui prennent la forme : SI...alors...des déclarations [Lekkas et Mikhailov, 2010]

La grande majorité des méthodologies existantes pour les diagnostics médicaux flous exigent des données d'être traitées en mode en différé (offline), comme groupe. Malheureusement, ceci permet seulement à un aperçu du domaine réel d'être analysé; il n'existe aucune preuve que si d'autres échantillons conditionnels, qui appartiennent au même domaine de données, deviennent disponibles, alors les exactitudes modèles demeurent inchangées. Pour cette raison, Kuncheva et Steimann [Kuncheva et Steimann, 1999] caractérisent les systèmes basés sur les règles classiques et fixes comme insuffisants.

Les classifieurs évolutifs flous '*evolving fuzzy classifier*' sont une approche relativement nouvelle pour développer les systèmes basés sur les règles floues, qui incluent plusieurs architectures et méthodes d'apprentissage online, qui résolvent ces problèmes. e-class '*evolving classifier*' est un algorithme qui permet de mettre en œuvre un classifieur évolutif flou en ligne. Dans ce chapitre nous allons expliquer sa méthodologie, ses différentes architectures.

2. Besoins en classifieurs évolutifs en ligne

Actuellement, nous assistons à une révolution de l'information, de grandes quantités d'informations sont produites à un rythme rapide par les capteurs dans les processus industriels, systèmes autonomes, par les utilisateurs de l'Internet, de l'industrie, de la finance, des marchés de consommation etc... Les défis auxquels sont confrontés les traitements de l'information et la classification en particulier, sont liés à :

1- la nécessité de faire face à d'énormes quantités de données ;

2-traiter les données qui arrivent en streaming, en ligne et en temps réel [Fayyad et al., 1996] ; [Domingos et Hulten, 2001].

Stocker l'ensemble des données et analyser les flux de données dans un mode hors ligne (batch mode) est souvent impossible ou peu pratique ; les flux de données sont très souvent non stationnaires. En même temps, la plupart des classifieurs classiques sont conçus pour fonctionner en mode de traitement par lots et ne changent pas leurs structures en ligne. Un exemple: la détection d'intrusions à un réseau, l'arrivée de nouvelles menaces inédites (les pirates sont très inventifs, ils utilisent de nouvelles façons de perturber le fonctionnement normal des serveurs et des utilisateurs). Les Classificateurs en hors ligne pré-entraîné peuvent être bons pour certains scénarios, mais ils ont besoin d'être repensés ou recyclés pour de nouvelles circonstances.

Les classificateurs conventionnels, appliqués aux flux de données, permettent d'extraire un «aperçu» du flux de données et exigent de stocker toutes les données antérieures, ce qui implique plus d'exigences en mémoire et en calculs. En revanche, les classifieurs incrémentales (ou en ligne) travaillent sur une base par-échantillon et ne nécessitent que les caractéristiques de l'échantillon courant, plus une petite quantité d'informations agrégées (une base de règles, et un petit nombre de variables nécessaires aux calculs récursifs); mais ils ne nécessitent pas tous l'historique du flux de données (tous les échantillons de données précédemment lus).

Ils sont aussi en un seul passage (chaque échantillon est traité une seule fois à un moment et est ensuite enlevé de la mémoire). Pour relever le défi de classer les données en streaming en ligne et en temps réel avec un classifieur qui s'adapte / évolue sa structure (sa base de règle), la famille eClass a été introduite.

3. Les classifieurs évolutifs « Evolving fuzzyclassifiers » :

C'est une nouvelle approche pour la classification en ligne qui est basée sur l'auto-développement (evolving) de la base de règles floues. Le terme « evolving classifier » est souvent utilisé dans le sens de 'développement graduel' de la structure du classifieurs (la base de règles floues).

Les classifieurs évolutifs flous s'adaptent aux nouvelles données entrante en auto-développant leur base de règles floues, la base n'est pas fixée comme dans les méthodes en hors ligne. Cette adaptation ne nécessite pas de réapprendre à nouveau, elle est basée sur des calculs récursifs. Cette approche est transparente, linguistiquement interprétable et applicable à l'apprentissage non supervisé ainsi qu'à l'apprentissage semi-supervisé.

Ce nouveau paradigme a été introduit pour les réseaux de neurones dans [Fritzke, 1994] et [Kasabov, 2001] ; pour les arbres de décision dans [Jin et Agrawal, 2003] et pour les systèmes à base de règles floues dans [Angelov et Buswell, 2001]. Ce paradigme émergent imite l'évolution d'individus dans la nature pendant leur cycle de vie, spécifiquement le développement mental autonome, typique des êtres humains: apprendre d'expériences, héritage, changement graduel, génération de la connaissance à partir d'opérations habituelles, et extraction des règles à partir des données.

3.1. Comparaison entre les classifieurs évolutifs et les classifieurs génétiques:

Angelov et Zhou ont fait une comparaison en 2008, entre les classifieurs évolutifs et les autres types de classifieurs ; ils indiquèrent que les algorithmes génétiques et évolutionnaires sont des techniques computationnelles qui empruntent à la nature les concepts *évolution génétique* ou *évolution des populations*, qui sont basés sur des mécanismes computationnels et qui imitent quelques paradigmes comme *la mutation, le croisement des chromosomes, la reproduction et la sélection* [Holland, 1975] ; [Goldberg, 1989].

Alors que les systèmes à base de règles floues génétiques et évolutionnaires génèrent les nouvelles règles à partir de la mutation ou du croisement des anciennes règles conduites par un processus aléatoire "dirigé" [Ishibuchi et al., 1995] ; [Cordon et al., 2004], les systèmes à base de règles floues évolutives génèrent les nouvelles règles à partir des nouvelles données entrantes progressivement, tout en préservant les règles déjà apprises. Les auteurs comparent cette manière d'apprendre à celle d'un enfant. Les classifieurs évolutifs peuvent apprendre à partir d'une base de règles initiales comme ils peuvent apprendre à partir de zéro.

Les algorithmes génétique / évolutionnaire ont été appliqués avec succès comme techniques d'apprentissage pour développer des classifieurs à base de règles floues, en hors ligne [Ishibuchi et al., 1995] ; [Cordon et al., 2004], et les arbres de décision [Gomez et al., 2002].

3.2. Comparaison entre les classifieurs évolutifs et les classifieurs incrémentales :

Après avoir comparé les classifieurs évolutifs aux classifieurs génétiques et évolutionnaires Angelov et Zhou (2008) ont également comparé avec les classifieurs évolutifs et les classifieurs incrémentales qui sont considérés comme l'une des techniques qui s'est intéressée au problème des trains de données (data Stream).

Les classifieurs incrémentales ont été implémentés dans différentes structures : arbres de décision [Ferrer-Troyano et al., 2005], réseaux de neurones [e.g, théorie de la résonance adaptative (ART) [Carpenter et al., 1992], quantificateurs du vecteur de l'érudition croissant (iLVQs)], probabilistes [e.g, versions incrémentales de classifieurs Bayésiens [Chai, et Chieu, 2002], analyse du discriminant linéaire (iLDA) [Pang et al., 2005], analyse en composante principale (iPCA) [Pang et al., 2005].

La structure des classifieurs incrémentales est fixe. Dans le cadre du problème des trains de données, de tels classifieurs ne peuvent traiter le problème du *drift* and *shift*. Dans la littérature de l'apprentissage automatique, par *drift*, on fait référence à la modification du concept à travers le temps. Il s'agit d'une transition en douceur de la distribution des données d'une région locale de l'espace des caractéristiques (décrit dans un classifieur flou par une règle floue) à une autre région [Widmer et Kubat, 1996], [Klinkenberg et Joachims, 2000]; par *shift*, on fait référence à la soudaine apparence d'une faute ou bien au changement brusque du régime d'opération [Angelov et al., 2006]. Pour pallier à ces changements brusques et soudains, on ne doit pas modifier les paramètres du classifieur seulement mais aussi changer sa structure.

4. L'algorithme e-class :

e-class est un algorithme de classification relativement nouveau qui a été introduit par Xydeas et al. [Xydeas et al., 2005] pour classer les signaux d'électroencéphalogramme (EEG) en deux catégories (douleur et non-douleur). C'est une méthode récursive, non itérative. Sa faible complexité de calculs et son nombre de variables limitées la rend appropriée pour les applications en temps réel telles que le diagnostic, la navigation robotique, la détection de fautes etc. Elle est basée sur l'algorithme de clustering en ligne (e-clustering) pour partitionner le domaine des paramètres en entrée et pour générer la base de règles [Lekkas et Mikhailov, 2010].

A noté que cet algorithme a la propriété de pouvoir commencer l'apprentissage à partir de zéro, ce qui est un fait unique de cette approche. Dans le cas où il existerait une base initiale développée en hors ligne par un autre classifieur ou initialisée par un expert, alors e-class peut faire évoluer ce classifieur en mode en-ligne, en incorporant de nouvelles règles et/ou en supprimant certaines anciennes règles [Angelov et al., 2007].

En 2010, **Lekkas et Mikhailov** ont démontré que les performances de ce type de classifieurs dépendent aussi de l'ordre des données en entrée. Pour chaque ordre de données différent, résulte une base de règles différente. Néanmoins, il est montré que les modèles d'e-Class peuvent être améliorés en arrangeant l'ordre des nouvelles données, en utilisant une simple stratégie d'optimisation.

Cette méthode inclut deux types d'architectures : *eClass0* avec étiquettes de classe comme sortie et *eClass1* qui est basée sur le principe de régression sur les paramètres.

4.1 ECLASS0 :

4.1.1 Architecture d'eClass0 :

Un classifieur est une projection topographique (mapping) de l'espace des paramètres à l'espace de l'étiquette de la classe. Un classifieur à base de règles floues (BRF) décrit, avec sa partie des antécédents, le partitionnement flou de l'espace du paramètre $x \in R^n$, et avec la partie conséquente, l'étiquette de classe L_i , $i = [1, K]$. La structure d'eClass0 suit la construction typique d'un classifieur BRF [**Ishibuchi et al., 1995**] ; [**Kuncheva, 2000**]:

R_i : Si (X_1 est au tour de X_1^{i*}) **et** (X_2 est au tour de X_2^{i*}) **et...et**

(X_n est au tour de X_n^{i*}) **Alors** (L^i) (1)

Où $X=[x_1, x_2, \dots, x_n]^T$ est le vecteur des paramètres, R_i dénote la i ème règle floue, $i = [1, N]$, N est le nombre de règles floues. (X_j est au tour de X_j^{i*}) dénote le j ème ensemble flou de la i ème règle floue. $j = [1, n]$, x^{i*} est le point focal de la i ème règle (le prototype de la i ème règle, l'échantillon de données). Et L^i est l'étiquette de la classe de l' i ème prototype (point focal).

L'inférence dans *eClass0* est produite en utilisant le principe de « winner takes all » [**Ishibuchi et al., 1995**] ; [**Klawonn et Klement, 1997**] ; [**Kuncheva, 2000**].

$$L = L^{i^*} \quad i^* = \arg \max_{i=1}^N (\tau^i) \quad (2)$$

Où τ^i dénote le degré de confiance dans la i ème règle floue, lequel est déterminé comme un produit (une t-norme qui représente l'opérateur ET logique [**Casillas et al., 2002**] ; [**Cordon et al., 2004**], des valeurs appartenant μ_j^i au j ème paramètre, $j = [1, n]$ et l'ensemble flou (x_j est au tour de x_j^{i*}) :

$$\tau^i = \prod_{j=1}^n \mu_j^i(x_j), \quad i = [1, N]. \quad (3)$$

Les fonctions d'appartenance, qui décrivent le degré d'association avec un prototype spécifique, sont de forme Gaussienne (caractérisée par de bonnes capacités de généralisation et de couverture de l'espace du paramètre):

$$\mu_j^i = \exp(-1/2 (d_j^i / r_j^i)^2) \quad , \quad i = [1, N], j = [1, n] \quad (4)$$

Où d_j^i est la distance euclidienne entre l'échantillon x et le prototype (point focal) de la i ème règle floue et r_j^i est l'étendue de la fonction d'appartenance qui représente aussi le rayon de la zone d'influence du j ème prototype de l' i ème règle.

A noter que cette représentation ressemble à la distribution normale [Hastie et al., 2001], [Duda et al., 2000] et l'étendue de la fonction d'appartenance peut aussi être représentée par la déviation standard. Le r_j^i de l'étendue est basé sur la dispersion σ_j^i [Angelov et Filev, 2005] des données par cluster / règle ($i = [1, N]$) et par paramètre ($j = [1, n]$).

$$\sigma_{jk}^i = \sqrt{\frac{1}{s_k^i} \sum_{l=1}^{s_k^i} d^2(x_j^l, x_j^{i*})}, \quad \sigma_{j1}^i = 1 \quad (5)$$

Où s_k^i dénote le support de l' i ème cluster / règle à l'instant K (quand les échantillons de données k sont lus).

A noter aussi que dans (4) et (5), la projection de la distance d_j^i sur l'axe formé par le j ème paramètre est utilisée. Cela mène aux clusters hyper-ellipsoïdaux avec des étendues différentes pour les différents paramètres (voir la Figure II-1, facilitant l'interprétation. Le support est le nombre de points qui sont dans la zone d'influence d'un cluster [Angelov et Filev, 2005]. Il est initialisé à 1 quand un prototype est produit, ensuite il est augmenté de façon incrémental par 1 pour chaque échantillon, qui est plus proche de ce prototype qu'à tout autre:

$$s_{k+1}^i = s_k^i + 1, \quad i = \text{arg}_{i=1}^N \min \|x_k - x^{i*}\|, K = 2, 3, \dots \quad (6)$$

Quand un nouveau cluster / règle est formé, $N \leftarrow N + 1$, sa dispersion initiale est rapprochée par la moyenne de la dispersion pour les règles floues existantes pour cette classe [Angelov et Filev, 2005] et son support est mis initialement à 1:

$$\sigma_k^{N+1} = \frac{1}{N} \sum_{i=1}^n \sigma_k^i, \quad s_k^{N+1} \leftarrow 1, \quad k = 2, 3, \dots \quad (7)$$

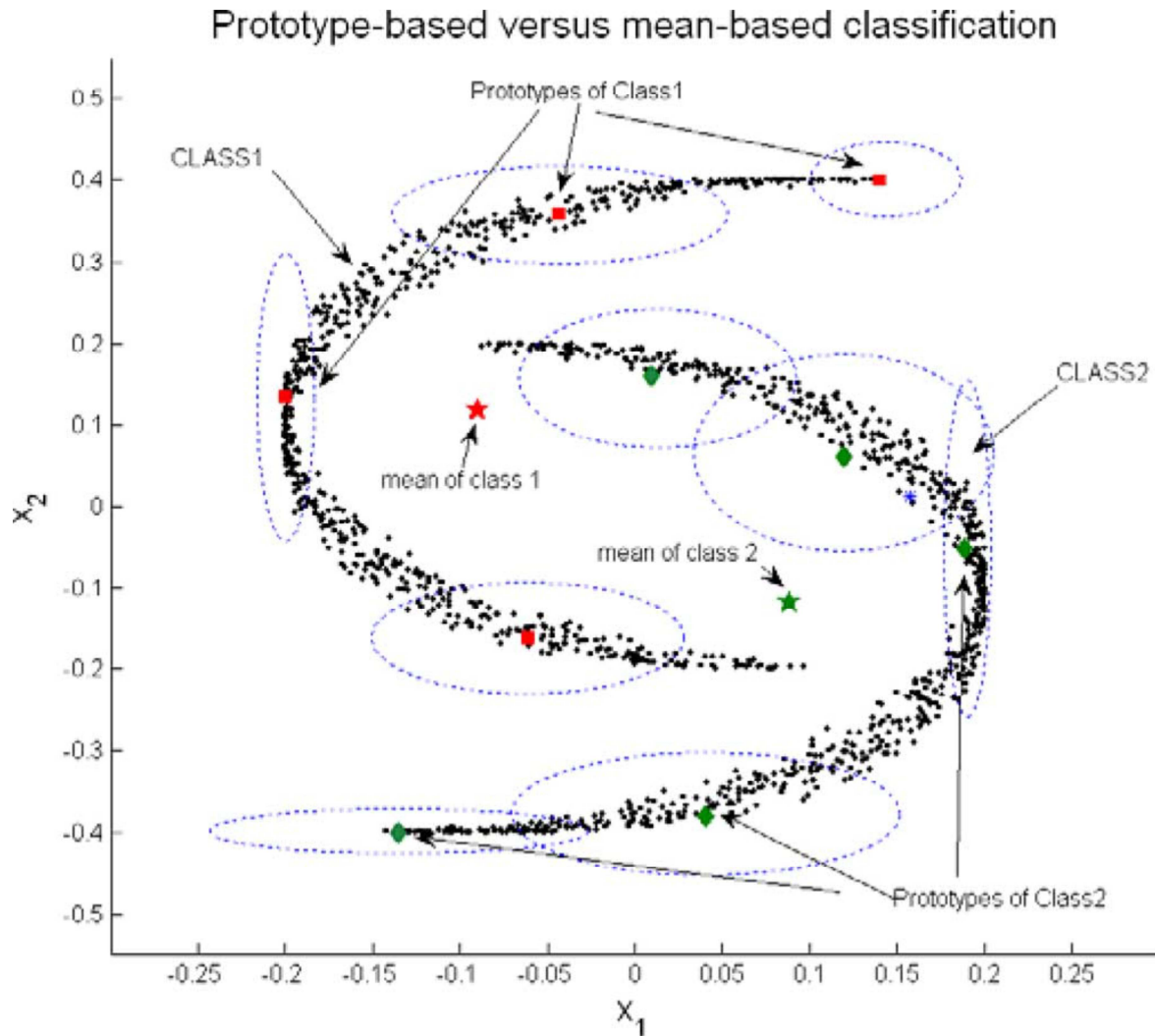


Figure II-1 : Classification basée-prototype contre classification basée-moyenne.

Ici, nous introduisons aussi le support par classe Q_k^l qui représente le nombre d'échantillons des données qui sont associés aux clusters, qui forment les parties antérieures des règles floues, qui ont toutes le même conséquent N^l :

$$Q_k^l = \sum_{j=1}^{N^l} S_k^j. \quad (6a)$$

La distance d dans (5) peut avoir une forme euclidienne [Angelov, 2002] ; [Angelov et Buswell, 2002] ou de cosinus. La distance du cosinus peut faire face aux problèmes tels qu'un nombre de paramètres différents et les valeurs zéro, elle est définie par [Duda et al., 2000] comme :

$$d_{cos}(x_k, x_l) = 1 - \frac{\sum_{j=1}^n x_{kj} x_{lj}}{\sqrt{\sum_{j=1}^n x_{kj}^2 \sum_{j=1}^n x_{lj}^2}} \quad (8)$$

Où le x_k et x_l sont deux vecteurs n-dimensionnels.

4.1.2 Apprendre la partie antérieure des règles :

Typiquement, les classifieurs BRF sont entraînés en offline en utilisant des algorithmes évolutionnaires [Ishibuchi et al., 1995] ; [Klawonn et Klement, 1997] ; [Kuncheva, 2000] ; [Roubos et al., 2003], ou des schémas basé-gradient tels que la rétro-propagation combinée avec les réseaux de neurones [Nauck et Kruse, 1997]. Cependant, la famille d'eClass est conçue pour les applications en ligne avec une structure à base de règles floues évolutives (auto-développement). Pour ce faire, les parties antécédentes des bases de règles floues sont formées depuis les flux de données autour des très descriptifs points focaux (prototypes) dans l'espace entrée-sortie $z = [x^T, l]^T$ par classe.

La différence principale entre eClass0 et un classifieur à base de règles floues conventionnel est :

- la structure ouverte de la base de règles (il s'auto-développe en mode en ligne "à partir de zéro", pendant que dans un classifieur à BRF conventionnel, il est déterminé en hors ligne et donc a une structure fixe) ;
- le mécanisme d'apprentissage en ligne qui prend en considération cette structure flexible de la base de règles.

En général, la base de règles floues est composée de K bases de sous-règles afin que, dans chaque base de sous-règles, les conséquents de toutes les règles soient les mêmes, mais le nombre de règles N ne devrait être en aucun cas plus faible que le nombre de classes ($N \geq K$). Chaque nouvel échantillon de données avec une étiquette de classe, qui n'a pas été vu précédemment, devient automatiquement un prototype. Il y a un remplacement du prototype et un mécanisme d'enlèvement, c'est habituellement temporaire (ce prototype est souvent remplacé plus tard par des prototypes plus descriptifs). Dans ce sens, le classifieur apprend "à partir de zéro" et le nombre de classes ne doit pas être su en avance [Duda et al., 2000].

Les points focaux des règles floues sont produits à la suite d'une projection de fonctions d'appartenance Gaussienne, centrée prototypes sur les axes qui représentent des paramètres différents. Ce processus est utilisé largement dans les approches données-dirigées [Roubos et al., 2003], pour la conception des systèmes flous et est facilité par l'usage de la mesure de distance Euclidienne standard ou de cosinus. La différence avec la plupart des autres approches de classification est que les ensembles flous dans l'eClass sont formés autour d'échantillons des données existants (prototypes), pas autour de moyens (Figure II-1). Notez que ce dernier peut être souvent infaisable, car les moyens sont des points abstraits habituellement inexistant.

a) *Partitionnement de l'espace des données*

La notion de base de l'algorithme de partitionnement est celui du potentiel, qui est défini comme une fonction Cauchy de la somme des distances entre certains échantillons de données et tous les autres échantillons des données, dans l'espace des données des paramètres [Angelov, 2004]. L'interprétation du potentiel est comme une mesure de la densité des données :

$$p_{k(z_k)} = \frac{1}{1 + (\sum_{i=1}^{k-1} d(z_k, z_i)) / (k-1)} \quad (9)$$

L'équation (9) définit le potentiel global [Angelov et Filev, 2005].

On définit le potentiel local (par classe) comme une mesure de densité des données pour la classe spécifique l :

$$p_k^l(z_k) = \frac{1}{1 + (\sum_{i=1}^{Q_k^l-1} d(z_k, z_i)) / (Q_k^l-1)} \quad (10)$$

Le partitionnement utilisant le potentiel est basé sur le principe suivant: "Le point avec le plus haut potentiel est choisi pour être le point focal de l'antécédent d'une règle floue" [Yager et Filev, 1993]; [Chiu, 1994]. Dans ce sens, des règles floues avec haut pouvoir descriptif et capacités de généralisation sont produites. Cela est illustré dans la Figure II-2.

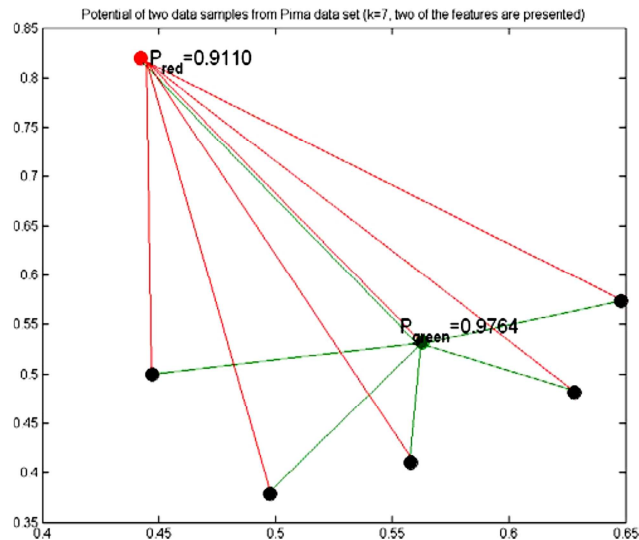


Figure II-2 : potentiel de deux échantillons de données d'ensemble des données du PIMA (k=7, deux des paramètres sont présentés).

Le calcul récursif qui est introduit a besoin du point de donnée courant z_k^j et (n + 1) quantités de données mémorisées seulement (β_k et x_k^j , $j = [1, n]$).

$$p_k^l(z_k) = \frac{1}{2 - \left[1 / \sqrt{\sum_{j=1}^{n+1} (z_k^j)^2} \right] \sum_{j=1}^{n+1} z_k^j b_{k-1}^j} \quad k=2,3,\dots, \quad p_1(z_1)=1 \quad (11)$$

Où

$$b_k^i = b_{k-1}^i + \frac{(z_k^i)^2}{\sqrt{\sum_{j=1}^{n+1} (z_k^j)^2}}, \quad b_1^i = \frac{(z_1^i)^2}{\sqrt{\sum_{j=1}^{n+1} (z_1^j)^2}}, \quad i = [1, n + 1].$$

Chaque fois qu'un nouvel échantillon des données est lu, il affecte la densité des données; par conséquent, les possibilités des centres existants doivent être mises à jour ; cette mise à jour peut aussi être récursive :

$$p_k^l(z_k^{i*}) = \frac{(Q_k^l - 1)p_{k-1}^l(z_l^{i*})}{Q_k^l - 1 + \left[(Q_k^l - 2) \left(\frac{1}{p_{k-1}^l(z_l^{i*})} - 1 \right) + d_{\cos}(z_l^{i*}, z_k) \right]} \quad k=2,3,\dots, \quad (12)$$

Une fois que le potentiel du nouvel échantillon des données est arrivé et calculé récursivement utilisant (11) et le potentiel de chacun des prototypes existants ont été mis à jour récursivement utilisant (12), ils sont comparés. Si le nouvel échantillon des données a un plus haut potentiel que ceux des prototypes existants de cette classe L, alors c'est un bon candidat pour devenir un point focal d'une nouvelle règle dans cette base de sous-règles, puisqu'il a un haut pouvoir descriptif et un potentiel de généralisation:

$$p_k^l(z_k) > p_k^l(z_l^{i*}) \quad \forall i^* \in N^l. \quad (13)$$

Former une nouvelle règle floue autour des prototypes ajoutés récemment mène à une augmentation graduelle de la dimension de la base des sous-règles, c'est pourquoi l'approche est appelée "évolutive":

$$z^{(N^l+1)*} \leftarrow z_k. \quad (14)$$

Le potentiel de la règle nouvellement produite est mis temporairement à 1 (il sera mis à jour pour prendre en considération l'influence de chaque nouvel échantillon de données, sur la généralisation du potentiel de ce point focal particulier par (12) avec chaque nouvel échantillon de données déjà lu):

$$p_k^l(z^{(N^l+1)*}) \leftarrow z_k. \quad (15)$$

Pour augmenter l'interprétabilité et mettre à jour la base de règles, on doit enlever les règles précédemment existantes, qui deviennent ambiguës après l'insertion d'une nouvelle règle. Par conséquent, chaque fois qu'une nouvelle règle floue est ajoutée, on vérifie également si l'un des prototypes déjà existant dans cette base de sous-règles est décrit par cette règle à un degré plus haut que e^{-1} (c'est une analogie au soi-disant état one-sigma [Duda et al., 2000]:

$$\exists i, i = [1, N^l], \quad \mu_i^j(z^{(N^l+1)}) > e^{-1} \quad \forall j, \quad j = [1, n] \quad (16)$$

Si l'un des prototypes précédemment existant remplit cette condition, les règles qui lui correspondent sont enlevées de cette base de sous-règles (en fait, remplacées par la règle récemment formée).

Les étendues des fonctions d'appartenance de la base de sous-règles de la classe respective sont également périodiquement mises à jour sur la base de la distribution de données [Angelov et Zhou, 2006]:

$$r_k^i = pr_k^i + (1 - p)\sigma_{k-1}^i, \quad i = [1, N^l] \quad (17)$$

Où p est un paramètre d'apprentissage; il détermine à quelle rapidité la diffusion des fonctions d'appartenance convergera à la dispersion locale de ce cluster. La valeur par défaut de 0.5 simplifie l'expression :

$$r_k^i = \frac{r_{k-1}^i + \sigma_{k-1}^i}{2}, \quad i = [1, N^l] \quad (18)$$

4.1.3 Algorithme eclass0: [Angelov et Zhou, 2008]

Begin eclass0

Initialisez eclass0 par le premier échantillon de données $z_1 = [x_1, l_1]$; $(p_1)_1 \leftarrow 1$ (ou par une classe initial si elle existe).

Tant que le flux de données n'est pas fini, faire pour l'échantillon de données.

Lire le vecteur de paramètres de l'échantillon de données x_k ;

Calculer l'appartenance à chacun des ensembles flous par (4);

Calculer la force d'activation de la règle par (3) ;

Classer l'échantillon en utilisant l'étiquette de classe gagnante \hat{L}_k par (2) ;

Au prochain pas ($K \leftarrow K + 1$) ou toutes les fois après cela

Obtenir le vrai label L_k de l'échantillon ;

Calculer son potentiel $p_k(z_k)$ en utilisant (11) ;

Mettre à jour les potentiels des prototypes existants par (12) ;

Si (13) influant alors

Ajouter un nouveau cluster basé centre au nouveau point de donnée(14) ;

Initialiser son potentiel à 1 comme indiqué par (15) ;

Mettre à jour les étendues de fonctions d'appartenance de la règle dans la base de sous-règles de la classe correspondante par (17) ou (18) ;

Si (16) influant alors

Enlever les règles qu'il influe ;

Sinon

Ignorer (ne pas changer la structure du cluster) ;

Mettre à jour les étendues de fonctions d'appartenance de la règle dans la base de sous-règles de la classe correspondante par (17) ou (18) ;

Fin si

Fin (tan que...faire)

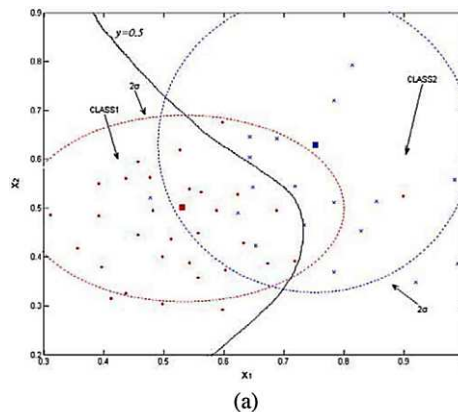
Fin (eclass0).

4.2 Eclass1 :

4.2.1 Architecture d'eClass1 :

L'architecture d'eClass1 diffère de manière significative de l'architecture d'eClass0 et des classifieurs à base de règles floues typiques [Ishibuchi et al., 1995] ; [Klawonn et Klement, 1997] ; [Nauck et Kruse, 1997] ; [Kuncheva, 2000] ; [Roubos et al., 2003] ; [Cordon et al., 2004] , parce qu'elles régressent le vecteur de paramètres utilisant le modèle en évolution Takagi-Sugeno de premier ordre à entrées multiples et à sorties multiples (multiple- input-multiple-output MIMO-ETS) [Angelov et al., 2004]. Noter que la surface de classification dans un flux de données est changeante dynamiquement (voir la figure II-3) ; eClass1 vise à évoluer sa base de règle de sorte qu'il réagisse à ceci en adaptant dynamiquement des paramètres du classificateur (diffusions, paramètres conséquents) aussi bien que les points focaux et la taille de la base de règle.

Évolution de la limite de classification dans eclass1-miso (ensemble des données indiens de Pima, 2 paramètres sélectionnés, $K=50$)



Évolution de la limite de classification dans eclass1-miso (ensemble des données indiens de Pima, 2 paramètres sélectionnés, $K=80$)

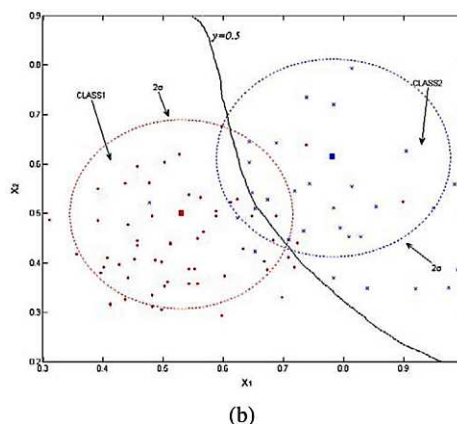


Figure II-3 : Évolution de limite de la classification dans eClass1-MISO (ensemble des données indiens de Pima, deux des paramètres) (a) après $k = 50$ les échantillons des données sont lus, et (b) après $k = 80$ les échantillons des données sont lus.

EClass1 est défini par un ensemble de règles de type suivant:

$$R^i = \mathbf{SI}(x_1 \text{ est au tour de } x_1^{(i)}) \text{ et } \dots \text{ et } (x_n \text{ est au tour de } x_n^{(i)})$$

$$\text{Alors}(Y^i = \bar{x}^T \theta) \tag{19}$$

Où $\bar{x} = [1, x_1, x_2, x_3 \dots x_n]^T$ dénote le $(n+1) \times 1$ vecteur de paramètres étendu et Y^i est la sortie.

Les sorties des règles particulières peuvent être normalisées de sorte qu'elles se résument à 1 :

$$\bar{y}^i = \frac{y^i}{\sum_{i=1}^N y^i}. \tag{20}$$

On peut vérifier facilement cela : $\sum_{i=1}^N \bar{y}^i = 1$

Les sorties normalisées par règle peuvent être interprétées [Zadeh, 1978] comme la possibilité des données d'appartenir à une certaine classe si nous employons des étiquettes cible pour les classes : 0 pour ceux qui n'appartiennent pas à la classe et 1 pour ceux qui lui appartiennent. À cet égard, cette approche a une similitude avec la matrice d'indicateur [Hastie et al., 2001], utilisée dans des classifieurs crisp en différé. Noter qu'un classifieur à BRF conventionnel a une étiquette de classe comme sortie de chaque règle. Dans le cas où MISO eTS est utilisé (pour les problèmes à 2 classes), les sorties par règles floues sont des scalaires. Dans ce cas, les paramètres du modèle local linéaire Θ sont une colonne de vecteurs

$\Theta = [\theta_{01}^i, \theta_{11}^i, \dots, \theta_{n1}^i]^T$. Dans le cas où MIMO eTS est utilisé, il peut former une matrice de $(n+1) \times K$

$$\Theta = \begin{bmatrix} \theta_{01}^i, \theta_{12}^i, \dots, \theta_{0k}^i \\ \theta_{11}^i, \theta_{12}^i, \dots, \theta_{1k}^i \\ \vdots \\ \theta_{n1}^i, \theta_{n2}^i, \dots, \theta_{nk}^i \end{bmatrix}.$$

Dans ce cas, les sorties des règles floues forment une rangée de dimension K-vectorel. Une sortie normalisée pour chaque classe, $\bar{y}^i = [\bar{y}_1^i, \bar{y}_2^i, \dots, \bar{y}_k^i]$.

La sortie globale de eclass1 est formée comme une somme pondérée des sorties normalisées de chaque règle, aussi appelé le principe «centre de gravité» [Klir et Folger, 1988], ce qui est typique pour la régression, prédiction de séries temporelles et la modélisation, mais pas pour la classification :

$$y = \sum_{i=1}^N \frac{t^i}{\sum_{j=1}^N t^j} \bar{y}^i. \tag{21}$$

La sortie est alors utilisée pour discriminer entre les classes.

Dans un problème à deux classes en utilisant eTS MISO, on peut avoir des cibles de 0 pour la classe 1 et 1 pour la classe 2, ou vice versa. La discrimination dans cette affaire sera à un seuil de 0,5 avec des sorties supérieures à 0,5, classées comme classe 2, et les sorties en dessous de 0,5, classées dans la classe 1, ou vice versa.

Dans les problèmes avec plus de deux classes, on peut appliquer eTS MIMO où chacune des K sorties correspond aux possibilités de l'échantillon de données d'appartenir à une certaine classe (comme indiqué précédemment). Noter que l'on peut utiliser eTS MIMO pour un problème à deux classes aussi. Les sorties cibles dans ce cas seront 2-D, par exemple $y = [1 \ 0]$ pour la classe 1 et $y = [0 \ 1]$ pour la classe 2, ou vice versa. L'étiquette de eclass1-MISO (qui est utilisée dans des problèmes à deux classes) est déterminée par :

$$\mathbf{SI} \ (y > 0.5) \ \mathbf{Alors} \ (\text{classe1}) \ \mathbf{Sinon} \ (\text{classe2}). \quad (22)$$

La figure 3 illustre la classification d'un ensemble de données de référence (PIMA) [Asuncion et Newman, 2007] en deux classes (signes de diabète ou aucun signe de diabète) après 50 et après 80 échantillons de données. Les deux règles floues formées sont représentées par leurs points focaux dans une illustration 2-D pour deux des neuf caractéristiques. La classification surface non linéaire (courbe en trait plein sur la Fig. 3) est dérivée de ce problème à deux classes avec $y = 0,5$. Cette figure illustre la dynamique lorsque l'on compare les deux parcelles (après que 50 échantillons soient lus, et après 80 échantillons), ce qui illustre la nécessité d'avoir des systèmes de classification en ligne et en constante évolution.

Dans eclass1-MIMO, l'étiquette est déterminée par la valeur la plus élevée du discriminateur \bar{y}_l :

$$L = L_{i^*}, \quad i^* = \mathit{arg}_{l=1}^k \max \bar{y}_l. \quad (23)$$

Noter que (23) n'est pas le principe typique "winner takes all" en terme de force d'activation des règles appliquées dans les problèmes de classification. En ce sens, l'approche proposée ressemble plus à la LDA [Duda et al., 2000] et machine à vecteurs de support [Vapnik, 1998], plutôt que de classifieurs flous typiques [Ishibuchi et al., 1995] ; [Nauck et Kruse, 1997] ; [Kuncheva, 2000] ; [Roubos et al., 2003] ; [Cordon et al., 2004] ; ; [Ishibuchi et al., 2004]. Noter également que les problèmes à deux classes peuvent être résolus par eclass1-MISO et par eclass1-MIMO, mais alors que le nombre de paramètres de la partie antécédente dans les deux architectures est le même, le nombre de paramètres qui en découlent dans eclass1-MISO est réduit de moitié ($n + 1$ car $\Theta = [\theta_{01}^i \ \theta_{02}^i \ \dots \ \theta_{n1}^i]^T$) par rapport à eclass1-MIMO, soit $2 \times (n + 1)$ car

$$\Theta = \begin{bmatrix} \theta_{01}^i, \theta_{12}^i, \dots, \theta_{n1}^i \\ \theta_{02}^i, \theta_{12}^i, \dots, \theta_{n2}^i \end{bmatrix}^T.$$

En eclass1, la qualité des règles floues est constamment surveillée par le calcul de leur «âge». L'âge d'une règle floue est particulièrement important pour les flux de données [Angelov et

Filev, 2005]. Il donne des informations accumulées sur le moment, où un échantillon a été affecté à un cluster ou règle respective. Il est bien connu que les approches progressives sont dépendantes de l'ordre. Avec l'âge d'une règle, on peut faire usage de cette particularité des flux de données. Il est proposé que l'âge soit calculé comme suit :

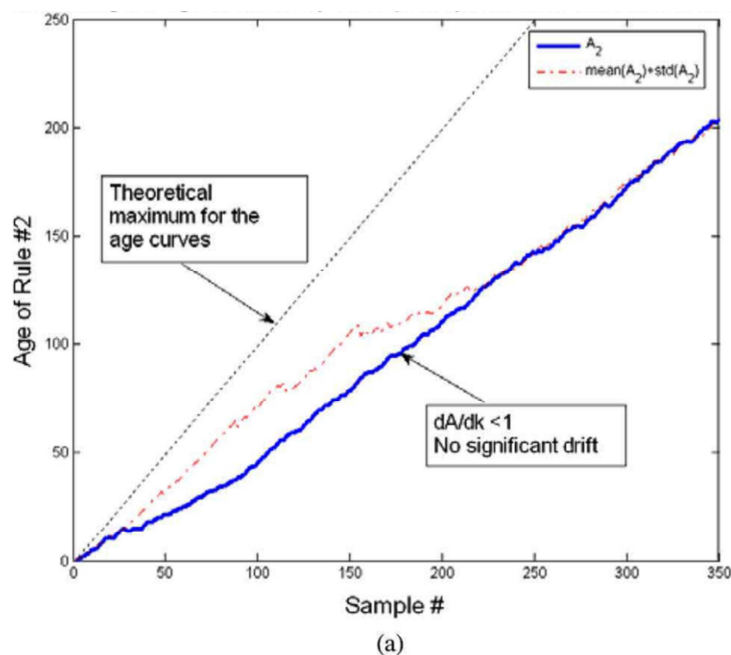
$$A^i = k - \frac{\sum_{l=1}^{s_k^i} k_l}{s_k^i} \quad (24)$$

Où k_l est le moment ou l'indice où l'échantillon de données est lu.

Noter que le facteur $\sum_{l=1}^{s_k^i} k_l / s_k^i$ peut facilement être calculé de manière récursive, et donc, A lui-même peut être calculé récursivement. Quand un nouvel échantillon de données crée une nouvelle règle, son âge est initié par l'indice de l'échantillon (l'instant de temps, si dans un flux relatif au temps). Chaque fois qu'un nouvel échantillon de données est simplement affecté à une règle existante, l'âge de cette règle diminue [voir (24)].

Si aucun échantillon n'est affecté à une règle, il prend de l'âge par 1. Noter que la gamme de A est $[0, k]$. L'âge de règles floues (taux de vieillissement est un dérivé de l'âge en fonction de la période d'échantillonnage k) s'est avéré très utile pour l'analyse en ligne du concept *drift* (mouvement) dans le flux de données. Alors que les classifieurs de la famille eClass visent à réagir au concept *shift* (changements) en évoluant leur structure en ligne, générer de nouvelles règles et en supprimant les règles dépassées, l'analyse en ligne du (premier et deuxième) gradient de l'âge des règles peuvent identifier le concept *drift*. Dans la Fig. 4, nous illustrons ceci pour l'exemple d'un ensemble de données ionosphère de l'Université de Californie, Irvine (UCI) [Asuncion et Newman, 2007].

Suivi des règles floues (ensemble de données ionosphère, la règle n ° 2, pas de dérive (drift))



La première image [Figure 4 (a)] illustre le cas où les règles prennent de l'Age avec un taux normal (le gradient de l'âge est inférieur mais proche de 1, et plus important encore, il est à peu près constant).

Suivi des règles floues (ensemble de données ionosphère, la règle n ° 3)

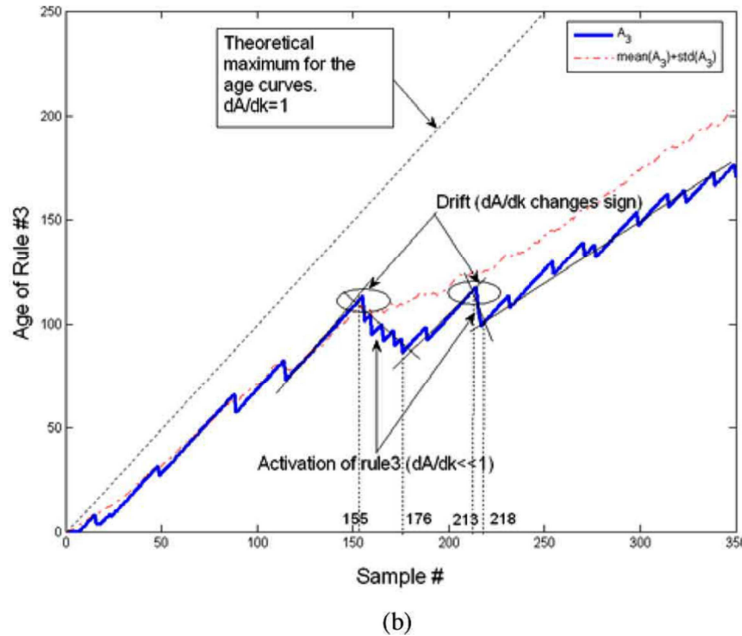


Figure II-4 : L'évolution d'âge (données ionosphère). (a) L'article 2, pas de dérive (drift). (b) L'article 3, drift détecté.

La deuxième image [Fig. 4 (b)] illustre un cas, où il y a des périodes (environ échantillons 155 et 213), où les données provoquent une forte mise à jour d'une règle spécifique (n ° 3). Dans ce cas, le gradient de l'âge diminue et devient même négatif, ensuite son âge augmente à nouveau (la règle n'est pas mise à jour aussi souvent ou pas du tout). L'explication est que lorsqu'une dérive (concept *drift*) se produit (comme dans la deuxième image), la fréquence d'utilisation de certaines règles (comme pour la règle 3) diminuera en raison d'une transition d'un état de fonctionnement, ce qui affecte la densité des données dans une région locale (autour du point focal de la règle 3) à un autre, ce qui affecte la densité des données dans une autre région locale.

La règle simple suivante, pour la mise à jour de la base de règle, est introduite en enlevant les règles âgées (règles dont l'âge est supérieur à l'âge moyen de cette règle de plus que la valeur "one-sigma" calculée de manière récursive jusqu'à ce moment):

$$IF A^i > \bar{A}^i + std(A^i) THEN (\lambda^i \leftarrow 0, N \leftarrow (N - 1)) \quad (25)$$

Où \bar{A}^i représente l'âge moyen et $std(A^i)$ désigne l'écart-type de l'âge de la règle i.

4.2.2 Apprentissage d'eClass1 :

L'apprentissage de eClass1 est basé sur l'approche de l'apprentissage en ligne des modèles TS avec structure en évolution, introduite dans [Angelov et Filev, 2004] et étendue pour le cas MIMO dans [Angelov et al., 2004]. L'apprentissage est basé sur la décomposition du problème d'identification dans : 1) conception de la structure à Base de Règles Floues et 2) identification des paramètres. Ces deux sous-problèmes peuvent être effectués en mode en ligne pendant un pas de temps unique (par échantillon).

Le premier sous-problème, celui de l'identification de la structure, est adressé à l'aide de dispersion à base de partitionnement flou comme décrit pour eClass0. La principale différence est que, dans eClass1, le potentiel est calculé globalement et non par classe. La raison en est que le but du partitionnement dans eClass0 est différent de celui dans eClass1. En eClass0, elle est réalisée dans le but d'identifier les prototypes représentatifs qui sont élevés au sein de la classe de densité. Noter que c'est au sein de la classe, mais pas nécessairement intra-classe (suivant la règle), la densité est exprimée mathématiquement par le potentiel local [voir (9)]. Cependant, dans eClass1, le regroupement sert à un rôle différent, il est l'une des deux phases de l'identification de modèle TS et est combiné avec le paramètre d'identification, en utilisant une version conséquente des moindres carrés récursifs (RLS). Une approche similaire est utilisée pour les systèmes flous TS hors ligne conçus dans [Chiu, 1994] pour faire évoluer en ligne TS (ETS) [Angelov et Filev, 2004]. Dans cette étude, nous utilisons l'apprentissage en ligne de l'ÉTS à régresser sur les caractéristiques, afin d'estimer la possibilité d'un échantillon de données appartenant à une certaine classe. Depuis, dans eClass1, nous avons un FRB qui n'est pas divisé par classe, le potentiel global est utilisé pour identifier les prototypes qui sont représentatifs de toute catégorie. Il est récursivement calculé par (11), mais appliqué au niveau global et non pas par classe. L'équation de la mise à jour du potentiel de tous les prototypes antérieurs, à chaque fois un échantillon de données nouvelle lecture est donnée par:

$$p_k(z^{i*}) = \frac{(k-1)p_{k-1}(z^{i*})}{k-1 + [(k-2)\{1/p_{k-1}(z^{i*})\} - 1] + d_{cos}(z^{i*}, z_k)} \quad k = 2, 3, \dots \quad (12a)$$

Alors, les potentiels du nouvel échantillon de données (11) et le potentiel mis à jour de tous les points focaux précédents sont comparés en appliquant (13). L'algorithme suit alors la logique de l'algorithme décrit dans la section II-B avec les seules différences étant que le potentiel est Global au lieu de local et N^1 est remplacé par N dans toutes les équations.

Une fois que les antécédents sont formés, l'estimation des paramètres conséquents (le deuxième sous-problème de la conception d'eClass1) est abordée comme un problème d'estimation floue pondérée (RLS) par règle [Angelov et Filev, 2004] :

$$\Theta_k^i = \Theta_{k-1}^i + C_k^i \lambda^i \bar{x}_k (y_k - \bar{x}_k^T \Theta_{k-1}^i), \quad \Theta_1^i = 0 \quad (26)$$

$$C_k^i = C_{k-1}^i - \frac{\lambda^i C_{k-1}^i \bar{x}_k \bar{x}_k^T C_{k-1}^i}{1 + \lambda^i \bar{x}_k^T C_{k-1}^i \bar{x}_k}, \quad C_k^i = \Omega I, \quad k = 2, 3 \dots \quad (27)$$

Où $C \in R^{N(n+L) \times N(n+L)}$ désigne la matrice de covariance, Ω est un nombre positif important, et I est l'identité de la matrice.

EClass1 est optimale (au sens LS) [Angelov et Filev, 2004] ; [Angelov et Zhou, 2006] en termes de paramètres conséquents Θ . En termes de ses antécédents et de la structure de base des règles, il est basé sur l'approche robuste de partitionnement en ligne [Angelov, 2004]. La condition (13) est très difficile à satisfaire, par conséquent, le plus souvent un petit nombre de règles sont formées ; en outre, la condition (16) supprime automatiquement les règles avec signification ambiguë. L'objectif de l'estimation optimale [vu du facteur entre parenthèses dans (26)] est de générer des valeurs aussi proches que possible de 1, pour les échantillons de certaines classes, et des valeurs proches de 0 pour toutes les autres classes par régression sur l'espace des paramètres [voir (19)]. De cette façon, eclass1 est unique comme classifieur BRF robuste non linéaire et en évolution, adapté pour les applications en ligne nécessaires pour les flux de données.

Les principales nouveautés d'eclass1 peuvent être résumées comme suit :

- 1) Il est en évolution (la base de règles n'est pas pré-entraînée et fixée, l'apprentissage peut commencer à "partir de zéro" avec le tout premier échantillon de données) ;

- 2) Il peut avoir une structure MIMO et donc construire un modèle de régression distinct pour chaque classe (sortie). Noter que si un échantillon avec une étiquette de classe inédite est remplie, la structure MIMO augmente naturellement par l'initialisation de l'apprentissage de la nouvelle (L + 1) ième classe à partir de ce point de la même manière que pour les classes L restantes ;

- 3) Il peut atteindre des taux de classification élevés par rapport aux bien connus classifieurs offline et incrémentale ;

- 4) Il s'agit d'un un-passe, récursif, donc il a des exigences de mémoire extrêmement basses ;

- 5) Il est utile pour l'analyse en ligne et le suivi du concept *drift* à l'aide de règles floues vieillissantes (voir Figure II-4) [Angelov et Zhou, 2008].

4.2.3 Algorithme d'eclass1: [Angelov et Zhou, 2008]

Begin Eclass1

Initialiser Eclass1 par le premier échantillon de donnée $z_1 = [x_1, l_1], (p_1)_1 \leftarrow 1$ (ou par une classe initiale si elle existe).

Tant que le flux de données n'est pas fini, faire pour l'échantillon de données.

Lire le vecteur de paramètres de l'échantillon de données x_k ;

Calculer l'appartenance à chacun des ensembles flous par (4) ;

Calculer la force d'activation de la règle par (3) ;

Classer l'échantillon en utilisant l'étiquette de classe gagnante \hat{L}_k par (22) pour eclass1-MISO ou par (23) pour eclass1-MIMO ;

Au prochain pas ($K \leftarrow K + 1$) ou toutes les fois après cela ;

Obtenir le vrai label L_k de l'échantillon ;

SI (dans le cas d'utilisation de la version MIMO et que l'étiquette de classe n'a jamais été vue précédemment) ;

Alors (initialiser le potentiel par 1 : $(p_k)_k \leftarrow 1$, augmenter la matrice des paramètres conséquents en ajoutant une colonne de $(N+1)$ zéros

$$\Theta_k^i = [\Theta_k^i \dots 0_{(n+1) \times 1}] \text{ et } K \leftarrow K + 1$$

Sinon (Calculer son potentiel $p_k(z_k)$ par (11) mais appliqué au potentiel global) ;

Mettez à jour les potentiels des prototypes existant par (12a) ;

SI (13) influant

Alors

Ajouter un nouveau cluster basé-centre au nouveau point de donnée (14) ;

Initialiser son potentiel global à 1 comme indiqué par (15) ;

Mettez à jour les étendues de fonctions d'appartenance par (17) ou (18) ;

SI (16) influant

Alors Enlever les règles qu'il influe ;

Fin (si) ;

Sinon

Ignorez (ne changez pas la structure du cluster) ;

Mettez à jour les étendues de fonctions d'appartenance par (17) ou (18) ;

Optionnel : enlevez les règles qui sont influencées par (25) ;

Fin (Si Alors Sinon) ;

Mettez à jours les paramètres conséquents par (26)-(27).

Fin (Tant que...Faire)

Fin (eclass1)

Outre les modèles *eclass0* et *eclass1*, on peut trouver dans la littérature d'autres variantes de l'algorithme *eclass* qui ont été proposées par différents auteurs, dans différents domaines.

4.3 Simpl_eclass : Simplification de l'algorithme E-class

4.3.1 Introduction

En 2011, Baruah, Angelov et Andreu ont publié une nouvelle version plus simplifiée d'*eclass* nommé *Simpl_eclass*, similaire au *e-class* classique, cette nouvelle version comprend aussi deux différents classifieurs de zéro et premier ordre (*Simpl_eclass0* et *Simpl_eclass1*). Les deux classifieurs diffèrent l'un de l'autre en terme de partie conséquente des règles floues et la stratégie d'apprentissage adoptée. La conception de *simpl_eClass* est basée sur le principe de l'augmentation de la densité, introduit récemment dans l'approche *simpl_eTS+*. L'apprentissage des règles dans *Simpl_eclass* n'implique pas le calcul de valeurs potentielles, ce qui lui permet d'avoir une phase de mise à jour moins coûteuse en termes de calculs, comparé à l'*eclass* classique.

Comparé aux autres classifieurs à base de règles floues, *Simpl_eclass* garde tous les avantages d'*eclass*, comme le fait d'être online et évolutif, ayant deux types différents (zéro et premier ordre). Comparé aux autres classifieurs non flous, il a l'avantage d'être interprétable et transparent (spécialement d'ordre zéro).

Enfin, cette nouvelle version simplifie le processus d'apprentissage d'*eclass*, tout en gardant la même précision sinon meilleure qu'un *eclass* classique.

4.3.2 Algorithme simplifié pour e-class0:

Le premier type de classifieur évolutif simplifié, *Simpl_eclass0* évolue en un ensemble de règles floues de type Takagi-Sugeno d'ordre zéro présenté comme suit: [Angelov et Zhou, 2008] :

$$\text{Règle}^i = \mathbf{SI}(x_1 \text{ est au tour de } x_1^{i*}) \mathbf{et} \dots \mathbf{et}(x_n \text{ est au tour de } x_n^{i*})$$

$$\mathbf{Alors Classe}^i \quad (28)$$

Où *Règleⁱ* est la *i*ème règle floue, $i=[1, N]$, N est le nombre de règles floues. $x = [x_1, x_2, \dots, x_n]$ Sont les variables en entrée représentées comme un vecteur de paramètres, $(x_j \text{ est au tour de } x_j^{i*})$ est le *j*ème ensemble flou de la *i*ème règle floue, $j= [1, n]$, x^{i*} est le point focal (le centre du cluster) de la *i*ème règle, et *Classeⁱ* est l'étiquette de classe du *i*ème point focal.

La stratégie bien connue « Winner takes all » (le vainqueur prend tout) est utilisée pour décider la classe d'une entrée donnée. Pour une entrée donnée x , la force d'activation (τ^i) de chaque règle est déterminée par la fonction d'appartenance gaussienne (μ^i) qui représente le degré de

rapprochement de x avec un point focal spécifique (x^{i^*} , $[i= 1, N]$). L'étiquette de classe (la partie conséquente), associée à la règle qui détient la plus haute force d'activation, est assignée à la variable en entrée x (29) :

$$Classe = Classe^{i^*}, \quad i^* = \operatorname{argmax}_{i=1}^n (\tau^i) \quad (29)$$

$$\tau^i = \prod_{j=1}^n \mu_j^i(x_j), \quad i = [1, N] \quad (30)$$

$$\mu_j^i = e^{-\frac{\|x-x^{i^*}\|^2}{2(r_j^i)^2}} \quad (31)$$

Où $r_j^i, i = [1, N], j = [1, n]$ est l'étendue de la fonction d'appartenance et représente la zone d'influence de la règle floue.

Simpl_eClass0 classe en ligne, chaque échantillon utilisant les règles existantes, quand la classe actuelle de l'échantillon est disponible à une étape plus tardive, il met à jour automatiquement la base de règles; donc, il exécute une séquence de classification et mise à jour du modèle consécutivement.

Les règles sont formées autour des points focaux [Angelov, 2002] ; [Angelov et Buswell, 2002] ; [Angelov et Filev, 2004] où un point focal est l'échantillon des données qui représente le plus convenablement tous les autres échantillons dans sa zone d'influence. Le point focal et la zone d'influence associés sont souvent désignés comme «centre» du cluster et "rayon" du cluster respectivement. Dans simpl_eClass0, la base de règles est formée d'un certain nombre de sous-bases de règles qui est le même que le nombre de classes, c . Chaque $i^{\text{ème}}$ base de sous-règles peut se composer de nombre de règles R_i avec la même étiquette de classe (partie conséquente) (32).

En termes de modèle d'identification floue, chaque base de sous-règles peut être considérée comme un ensemble de clusters représentant une classe et chaque cluster dispose d'un point focal associé.

$$\sum_{i=1}^c R_i = N, \quad R_i > 0, \quad \text{ainsi } N \geq c \quad (32)$$

Après la phase de classification pour un nouvel échantillon de données, quand et si son étiquette de classe actuelle devient disponible, il est mis en correspondance avec la partie conséquente des règles existantes dans la base de règles. Si aucune correspondance n'est trouvée, cela signifie qu'il s'agit d'une étiquette de classe qui n'a pas été déjà vue, alors le point d'échantillonnage devient un nouveau point focal. Cependant, à un stade ultérieur, ce point focal pourrait être remplacé par un autre, comme expliqué dans le paragraphe suivant.

Le principe de base qui souligne la décision de considérer un nouvel échantillon de données ($x_k, k^{\text{ème}}$ échantillon) en tant que point focal, est basé sur l'accroissement de la densité. Si la

densité des données autour du nouvel échantillon de données augmente ou diminue à l'égard de tous les points focaux existants, il est alors considéré comme un nouveau point focal et une règle est formée (condition A). Si ce point focal nouvellement formé n'est pas dans la zone d'influence de tous les points focaux existants, alors cette condition indique que le nouvel échantillon se trouve dans une région qui n'est pas encore couverte par les règles existantes (augmentation de la densité négative). D'autre part, si elle se trouve dans la zone d'influence de tous les points focaux existants (ou très proche d'un point focale existant) et a une densité plus élevée (augmentation de densité positive), alors l'échantillon de données remplace celle-ci (condition B). Cette condition permet à un seul échantillon, l'échantillon avec la plus forte densité, d'être un point focal au sein d'une région, évitant ainsi des règles redondantes, en fournissant une bonne généralisation des données sous-jacentes. L'augmentation de la densité dans la condition A peut être déterminée en calculant la distance entre l'échantillon actuel et la moyenne globale (moyenne de tous les échantillons de données). En `simpl_eClass0`, cependant, la valeur moyenne par classe / base de sous-règles est nécessaire au lieu de la moyenne globale.

Condition A :

Si $(\delta_k^i = R^i)$ Alors x_k devient le nouveau point focal de l' $i^{\text{ème}}$ classe.

Où δ_k^i est l'augmentation de densité dans l' $i^{\text{ème}}$ classe lorsque $x_k(z_k = [x_k y_k]) y_k = \text{class}_k^i$ arrive et peut être déterminée en utilisant (33), R^i est le nombre de règles / points focaux dans l' $i^{\text{ème}}$ classe.

Condition B : $\exists i, i = [1, R^i], \mu_i^j(x_k) > e^{-1}, \forall j, j = [1, n]$

$$\delta_k^i = \left| \sum_{j=1}^{R^i} \text{sign}(\gamma_k^{j*}) \right|, \gamma_k^{ij*} = (x_k^{ij*})^2 - (x_k)^2 + 2(x_k - x_k^{ij*})\bar{x}_k^{ij*} \quad (33)$$

Où x_k^{ij*} est le $j^{\text{ème}}$ point focal dans la $i^{\text{ème}}$ classe, \bar{x}_k^{ij*} est la moyenne de tous les points sauf le $j^{\text{ème}}$ point focal dans la $i^{\text{ème}}$ classe, ainsi peut être calculée en soustrayant la valeur du $j^{\text{ème}}$ centre depuis la moyenne de tous les points, la moyenne de tous les points appartenant à la $i^{\text{ème}}$ classe peut être calculée de manière récursive comme montré par (34).

$$\bar{x}_k = \frac{k-1}{k} \bar{x}_{k-1} + \frac{1}{k} x_k; \bar{x}_{k1} = x_1 \quad (34)$$

Contrairement à `eClass0`, la détermination des points focaux dans `simpl_eClass0` utilise l'approche introduite dans `simpl_eTS +`. La méthode utilisée dans `eClass0` nécessite le calcul du potentiel pour chaque échantillon et met à jour le potentiel de tous les points focaux existants. Cependant, `simpl_eClass0` nécessite seulement de mettre à jour la valeur moyenne ramenant ainsi la complexité de calcul par échantillon de $O(N)$ à $O(1)$.

`Simpl_eClass0` hérite de la propriété de rayon / la propagation de la fonction d'appartenance mise à jour depuis la famille `eClass`, qui dépend de la distribution sous-jacente des données et est donnée par:

$$r_k^{ij} = \rho r_{k-1}^{ij} + (1 - \rho) \sigma_k^{ij} \quad (35)$$

$$(\sigma_k^{ij})^2 = \frac{1}{s_k^{ij}} \sum_{i=1}^{s_k^{ij}} \|x^{i*} - x_i^i\|^2 \quad (36)$$

Où r_k^{ij} est le rayon / propagation de la fonction d'appartenance du $j^{\text{ème}}$ point focal du $i^{\text{ème}}$ cluster. ρ Est le taux d'apprentissage (la valeur par défaut est 0,5), σ ressemble à la variance autour du point focal j , s_k^{ij} est le support (nombre de points de données à l'intérieur de la zone d'influence du $j^{\text{ème}}$ point focal) [Angelov et Zhou, 2008]. σ Peut être calculé récursivement comme suit :

$$\sigma_k^{ij} = \frac{s_k^{ij} - 1}{s_k^{ij}} \sigma_{k-1}^{ij} + \frac{1}{s_k^{ij}} \|x^{j*} - x_k\|^2; \quad \sigma_0^{ij} = 1 \quad (37)$$

L'algorithme suivant résume l'algorithme simpl_eClass0 :

Algorithme

1. Lire l'échantillon (suivant), x_K ($k = 1, 2, \dots$)

2. Classification:

Si $k = 1$ (x_K est le premier échantillon) **Alors**

Si $N = 0$ (pas de règles encore) **Alors**

la classification ne peut pas être effectuée

Sinon (base de règles existante)

calculer le degré d'appartenance en utilisant (31), la

force d'activation de la règle en utilisant (30), et attribuer une étiquette de

Classe à l'échantillon en utilisant (29).

3. Mise à jour du modèle:

Obtenir l'étiquette de classe actuelle (si disponible) $y_K = \text{classe}_k^i$

Si $N = 0$ **ou** $k = 1$ **ou** classe_k^i n'a pas été vu jusqu'à présent **Alors**

Attribuer x_K comme nouveau point focal dans une nouvelle classe / base de sous-règle i et créer une règle de la forme donnée dans l'expression (28).

Fixer \bar{x}_k^i (moyenne) = x_K ; $s_k^i = 1$; $R^i = 1$.

Initialiser la propagation de la fonction d'appartenance pour le nouveau point focal dans la $i^{\text{ème}}$ classe / base de sous-règle.

Sinon

Considérant que l'échantillon, x_K appartient à la $i^{\text{ème}}$ classe, calculer l'augmentation de densité en utilisant (33).

Si les deux conditions (A) et condition (B) sont vraies Alors

Remplacer le point focal respectif dans l'ième classe avec x_K .

Mettre à jour la propagation de la fonction d'appartenance en utilisant (35).

Sinon *si seule condition (A) est satisfaite Alors*

Ajouter x_K comme un nouveau point focal dans la classe i.

$$R^i = R^i + 1$$

Initialiser la propagation de la fonction d'appartenance pour le nouveau point focal.

Sinon

x_K n'influence pas la structure du classifieur pour la classe i.

Mettre à jour la propagation de la fonction d'appartenance du point focale le plus proche de x_K dans la i^{ème} classe.

4.Si *il n'y a plus aucun échantillon Alors on s'arrête*

Sinon *passez à l'étape 1.*

4.3.3 Algorithme simplifié pour e-class1 :

Le simpl_eClass1 est construit sur un modèle de régression qui est utilisé pour les tâches de classification. L'idée est de transformer l'espace de paramètres en un espace linéairement séparable. La structure de simpl_eClass1 est basée sur le premier ordre multi-entrées multi-sorties (MIMO) de système de commande-BRF avec une structure et des paramètres adaptatifs. Les règles de simple_eClass1 sont de la forme suivante :

$$\text{Règle}^i = \mathbf{SI}(x_1 \text{ est au tour de } x_1^{i*}) \mathbf{et} \dots \mathbf{et}(x_n \text{ est au tour de } x_n^{i*})$$

$$\mathbf{Alors} \ y^i = x_e \theta^i \tag{38}$$

Où $y^i = [y_1^i, y_2^i, \dots, y_m^i]$ est la sortie de dimension m de la règle i.

$x_e = [1 \ x_1 \ \dots \ x_n]$ est le vecteur d'entrée étendue, θ^i est la matrice de sous-système de paramètres locale donnée comme suit:

$$\theta^i = \begin{bmatrix} a_{01}^i & a_{02}^i & \dots & a_{0m}^i \\ a_{11}^i & a_{12}^i & \dots & a_{1m}^i \\ \vdots & & & \\ a_{n1}^i & a_{n2}^i & \dots & a_{nm}^i \end{bmatrix} \tag{39}$$

Ainsi chaque élément $(y_1^i, y_2^i, \dots, y_m^i)$ de la sortie de dimension m, y^i peut être représenté comme suite : $y_1^i = a_{01}^i + x_1 a_{11}^i + \dots + x_n a_{n1}^i$; $y_2^i = a_{02}^i + x_1 a_{12}^i + \dots + x_n a_{n2}^i$ etc.

En simpl_eClass1, avant la phase d'apprentissage de la règle, l'étiquette de classe d'un échantillon donné est convertie sous forme numérique afin que la conséquence d'une règle puisse être représentée comme dans l'expression (38). Par exemple, la classification binaire de la classe 1 peut être représentée sous la forme $y = [1 \ 0]$, et celle de la classe 2 comme $y = [0 \ 1]$. De même, pour un problème à trois classes, la classe 1 peut être représentée sous la forme $y = [1 \ 0 \ 0]$, la classe 2 comme $y = [0 \ 1 \ 0]$, et la classe 3 comme $y = [0 \ 0 \ 1]$. Ici, la dimension m de y est équivalente au nombre de classes.

De même que pour simpl_eClass0, la partie antécédente d'une règle dans la base de règles simpl_eClass1 est basée sur la valeur de l'augmentation de densité. Cependant, simpl_eClass1 ne subdivise pas sa base de règles en nombre de bases de sous-règles équivalentes au nombre de classes. Ainsi, il est seulement nécessaire de maintenir une moyenne globale (et non la moyenne par classe) de tous les échantillons de données sur la base desquels l'augmentation de densité peut être calculée. Pour simpl_eClass1, condition A, condition B, et (34) ne peut être donnée sous la forme :

Condition A: Si $(\delta_k = N)$, Alors z_k devient un nouveau point focal.

Condition B: $\exists i, i = [1, N], \mu_j^i(x_k) > e^{-1}, \forall j, j = [1, N]$

$$\delta_k = \left| \sum_{i=1}^N \text{sign}(\gamma_k^{i*}) \right|, \quad \gamma_k^{i*} = (z_k^{i*})^2 - (z_k)^2 + 2(z_k - z_k^{i*})\bar{z}_k^{ij*} \quad (33a)$$

Les paramètres de règles conséquents sont estimés localement à l'aide de l'approche des moindres carrés récurrents pondérés (WLRS) [Angelov et Filev, 2004] en utilisant (40) et (41).

$$\theta_k^i = \theta_{k-1}^i + C_k^i x_{ek-1}^T \lambda^i(x_{k-1}) (y_k - x_{ek-1} \theta_{k-1}^i) \quad (40)$$

$$C_k^i = C_{k-1}^i - \frac{\lambda^i(x_{k-1}) C_{k-1}^i x_{ek-1}^T x_{ek-1} C_{k-1}^i}{1 + \lambda^i(x_{k-1}) x_{ek-1} C_{k-1}^i x_{ek-1}^T} \quad (41)$$

Où C^i est la matrice de covariance de l' $i^{\text{ème}}$ règle avec une dimension de :

$(n+1) \times (n+1)$ et $\theta_0^i = 0, C_1^i = \Omega 1, i = [1, N]$ et Ω est fixé sur une valeur élevée.

Comme décrit dans [Angelov et Filev, 2004], quand une nouvelle règle est ajoutée, ses paramètres sont réglés à la moyenne pondérée des paramètres des N règles déjà existantes (42), et la matrice de covariance est donnée par (43). Quand une nouvelle règle remplace une

règle existante, alors le premier hérite simplement les paramètres de ce dernier. Dans ce cas également, la matrice de covariance est initialisée comme donné par (43).

$$\theta_k^{N+1} = \sum_{i=1}^N \lambda^i \theta_{k-1}^i \quad (42)$$

$$C_k^{N+1} = \Omega 1 \quad (43)$$

Pour déterminer l'étiquette de classe d'un échantillon donnée, la sortie global(\hat{y}) est estimée par la somme pondérée des sorties de toutes les règles (44), et l'index correspondant à l'élément avec la plus grande valeur de (\hat{y}) est désigné comme l'étiquette de classe de l'échantillon (45).

Par exemple, supposons que pour un échantillon donné, si $\hat{y} = [0.38 \ 0.48 \ 0.29]$ dans un problème à 3 classes alors l'échantillon assigné à l'étiquette de classe est le deuxième.

$$\hat{y} = \sum_{i=1}^N \lambda^i y^i; \quad \lambda^i = \frac{\tau^i}{\sum_{j=1}^N \tau^j} \quad (44)$$

Où \hat{y} est la sortie globale estimée, τ^i est la force d'activation de la $i^{\text{ème}}$ règle et calculée par (30) et (31).

$$classe = classe^{i^*}, \quad i^* = \arg_{i=1}^m \max(\hat{y}) \quad (45)$$

La classification et les phases de mise à jour du modèle de `simpl_eClass1` sont résumées dans l'algorithme suivant :

Algorithme

1. Lire l'échantillon (suivant), $x_k (k = 1, 2, \dots)$.

2. Classification :

Si $k = 1$ (x_k est le premier échantillon) **Alors**

Si $N = 0$ (pas de règles encore) **Alors**

la classification ne peut pas être effectuée.

Sinon (base de règles existante)

calculer le degré d'appartenance en utilisant (31),

la force d'activation de la règle en utilisant (30), et le poids de chaque règle (λ) en utilisant (44),

Calculer la sortie de chaque règle (partie des conséquents de (38)),

Calculer la sortie globale en utilisant (44) et attribuer une étiquette de classe à l'échantillon en utilisant (45).

3. Mise à jour du modèle:

Obtenir l'étiquette de classe actuelle (si disponible) $y_k = \text{classe}_k^i$
et le convertir en format numérique (vectoriel).

Si $N = 0$ **ou** $k = 1$ **ou** classe $_k^i$ n'a pas été vue jusqu'à présent

Attribuer z_k comme un nouveau point focal.

Fixer \bar{z}_k (moyenne) = z_k , $\delta_k = 1$; $N = 1$.

Initialiser les paramètres de la partie conséquences et la propagation de la fonction d'appartenance.

Sinon

calculer l'augmentation de densité en utilisant (33a).

Si les deux conditions (A) et condition(B) sont vraies **Alors**

Remplacer le point focal respectif avec z_k .

Mettre à jour la propagation de la fonction d'appartenance en utilisant (35).

Mise à jour les paramètres conséquents en utilisant (40).

Sinon si seulement la condition (A) est satisfaite **Alors**

Ajouter z_k comme un nouveau point focal.

$N = N + 1$

Initialiser la propagation de la fonction d'appartenance pour le nouveau point focal.

Définissez les paramètres de la moyenne pondérée des paramètres de règles existantes en utilisant (42) - (43).

Sinon

z_k n'influence pas la structure du 7^{ème} cluster.

Mettre à jour la propagation de la fonction d'appartenance du point focal le plus proche de z_k .

Mettre à jour les paramètres conséquents en utilisant (40).

4. Si il n'y a aucun échantillon de plus **Alors** s'arrêter

Sinon passez à l'étape 1.

4.3 Une méthode de pondération de règles en ligne pour classer les flux de données

Les structures à base de règles floues évolutives représentent des méthodes très puissantes pour la classification de flux de données en ligne. Les règles floues sont générées, modifiées et supprimées automatiquement dans ces systèmes. L'un des algorithmes les plus simples mais efficace de ce type est le classifieur évolutif (eClass) qui construit les règles sans aucune connaissance préalable, à partir de zéro "from scratch". Cependant, cet algorithme ne peut pas faire face correctement aux concepts *drift* et *shift* dans le concept de données. En 2012, Shahparast et al., ont proposé une nouvelle méthode efficace en ligne pour augmenter les performances de cet algorithme en fixant un poids approprié pour chaque règle et gérer le *Drift* et *Shift* des données en ajustant les poids appropriés; la zone d'influence de chaque règle peut être facilement contrôlée et changée dans le cas de modifications de l'environnement [Shahparast et al., 2012].

4.4.1 Méthode de réglage de poids :

L'utilisation du poids a une influence directe sur l'efficacité des classifieurs, dans les recherches antérieures, différentes approches heuristiques ont été proposées pour apprendre les poids des règles flous si-Alors [Ishibuchi et Yamamoto, 2005] ; [Mansoori et al., 2007]. Dans leurs travaux, [Shahparast et al., 2012] ont proposé une méthode de réglage de poids en ligne pour définir un poids approprié à chaque règle en ce qui concerne la modification de l'environnement. Cette méthode est une version étendue d'un paramètre de réglage en mode batch qui a été introduit avant par les auteurs [Zolghadri Jahromi et Taheri, 2008]. Cette approche est appliquée dans plusieurs ouvrages tels que: pondération d'échantillon [Zolghadri Jahromi et al., 2009], pondération de paramètres [Ghodratnama et al., 2010], la classification avec le plus proche voisin et hyperplan linéaire séparables [Taheri et al., 2009] et le réglage de poids de réseaux de neurones [Hamzeloo et al., 2011] ; d'ailleurs, en raison de sa grande vitesse et de ses bonnes performances, il a également été développé dans un autre travail [Yang et al., 2010].

L'algorithme est en fait une méthode de recherche hill-climbing. Il commence par une solution initiale (un poids initial pour les règles), et tente d'améliorer cette solution en trouvant le meilleur voisin. Dans ce cas, la solution est que le voisin qui ne diffère que par un seul paramètre est le poids de chaque règle. Considérons que nous voulons trouver un poids optimal pour la règle suivante:

$$R_i = \mathbf{SI}(x_1 \text{ est au tour de } x_1^{i*}) \mathbf{et} \dots \mathbf{et}(x_n \text{ est au tour de } x_n^{i*})$$

$$\mathbf{Alors } T^i \mathbf{ Avec } CF_i \quad (46)$$

Où CF_i est le grade de certitude (c'est à dire le poids de la règle) de la règle R_i . Pour trouver CF_i à chaque étape, tous les poids à l'exception de ce poids sont supposés fixes, et la juste valeur de ce paramètre est calculée de telle sorte que la précision de la classification des données d'entraînement sera maximale. La méthode de raisonnement pour classer un échantillon de données est acquise à partir de la multiplication du poids de la règle par la compatibilité des échantillons de donnée avec la règle appropriée:

$$R^* = \arg \max_{i=1}^n (\mu_i(x_k) \times CF_i) \quad (47)$$

La procédure de recherche de CF_i est brièvement décrite comme suit:

Dans un premier temps, l'effet de la CF_i sur la classification de chaque échantillon de données x_k est déterminé. A cet effet, CF_i est réglé sur deux valeurs différentes $+\infty$ (un grand nombre) et 0, et x_k est classé en prenant compte ces deux différents poids de règles. Dans le cas où CF_i est fixé à un grand nombre, R_i est absolument la règle gagnante pour x_k . En revanche, lorsque CF_i est mise à zéro, R_i n'a aucune chance d'être la règle gagnante pour x_k . Par conséquent, une autre règle R_i^{\sim} (la règle adversaire de R_i) est la règle gagnante pour x_k . $R_i^{\sim}(x_k)$ (La règle adversaire de R_i par rapport à x_k qui est brièvement montrée par R_i^{\sim}) est définie comme suit:

$$R_i^{\sim} = \arg \max_{j \neq i} (\mu_j(x_k) \times CF_j) \quad (48)$$

Si x_k est correctement classé par l'une des règles R^i ou R_i^{\sim} , alors x_k devrait être considérée dans la pondération de R^i . En fait, x_k est classée selon les valeurs de la CF_i . Par conséquent, il existe un seuil positif pour CF_i ou si une valeur inférieure à ce seuil est affectée à la CF_i , alors x_k est classé par la règle adverse R_i^{\sim} , sinon il est classé par R^i . Si CF_i est exactement réglée à ce seuil, alors les règles R^i et R_i^{\sim} ont la même chance de gagner dans la classification de x_k . Par conséquent, ce seuil est calculé comme suit:

$$th^t = \frac{CF^{\sim} \times \mu_i^{\sim}(x_k)}{\mu_i(x_k)} \quad (49)$$

Où CF^{\sim} est le poids de R_i^{\sim} , et $\mu_i(x_k)$ est la valeur d'appartenance de x_k dans R_i^{\sim} . Cela signifie que x_k est classé par R^i et R_i^{\sim} pour certaines CF_i inférieures ou supérieures à ce seuil, respectivement. Si x_k est correctement classé ou mal classé par ces deux règles, il se peut que le poids de R^i n'a aucune incidence sur le classement de x_k . Nous appelons ces échantillons, des échantillons de données don't-care samples (ils ne sont pas important pour la pondération de la règle R^i); par conséquent, nous ne considérons pas les échantillons don't-care pour ajuster ce poids.

Après l'obtention des valeurs seuils pour tous les échantillons de données (qui ne sont pas don't care) par (49), ils sont classés dans l'ordre croissant en tant que $[th_1, th_2, \dots, th_k, th_{k+1}, \dots, th_m]$. Sur la base de ces seuils, il ya m-1 intervalles internes (th_k, th_{k+1}) et 2 intervalles externes $(0, th_1)$ et $(th_m, +\infty)$. Dans chaque intervalle, certains échantillons de données sont classés correctement, et certains sont mal classés. L'intervalle avec le maximum de précision (nombre maximal de données est classé correctement) est choisi comme l'intervalle optimal et la valeur pour CF_i sera sélectionnée à partir de cet intervalle. Si l'intervalle optimal est interne, la valeur moyenne de cet intervalle est attribuée à la CF_i , sinon, si le dernier intervalle est optimale, $th_m + T$ est considérée comme une valeur appropriée pour la CF_i , où T est un grand nombre, et enfin si le premier intervalle est optimal, le poids est mis à zéro. Les règles de poids zéro peuvent être ignorées, ce qui conduit à avoir une base de règles plus compacte. Nous appliquons cette méthode pour toutes les règles, jusqu'à ce que tous les poids soient fixes et inchangés.

4.4.2 Le mécanisme de réglage de poids:

Dans cet algorithme proposé par [Shahparast et al., 2012], il existe des situations dans lesquelles ils se sont fixés un poids pour une nouvelle règle générée, où ils mettent à jour le poids des règles précédentes construites. Ils décrivent ces situations; chaque fois qu'ils veulent définir ou mettre à jour un poids de règle, ils utilisent le mécanisme de réglage de poids qui est expliqué ici. Cependant, ils appliquent plusieurs changements dans l'algorithme pour qu'il soit utile pour l'utilisation dans les systèmes en ligne. Dans l'algorithme de base, tous les échantillons de données sont utilisés pour définir un poids de règle, tandis qu'ils utilisent simplement des échantillons de données dans la fenêtre et calculent seulement les seuils de ces données.

Après l'obtention de ces seuils, ils trient dans une liste croissante de $[th_1, th_2, \dots, th_m]$. Pour obtenir l'intervalle optimal, il n'est pas nécessaire de calculer la précision exacte de chaque intervalle, il peut être fait avec un seul passage sur les seuils triés. Au lieu de calculer la précision exacte des intervalles, nous pouvons calculer la différence de la précision dans chaque intervalle avec le premier intervalle $[0, th_1)$, qui est appelé dans le présent document, le biais de la précision de l'intervalle et est calculé comme ci-dessous:

$$\begin{cases} Acc_k = Acc_0 + Biais_k \\ Acc_0 = 0 \end{cases} \quad (50)$$

Où Acc_k représente la précision du k-ième intervalle $((th_k), th_{k+1})$. Et $Biais_k$ est calculée comme ci-dessous:

$$Biais_k = \sum_{r=0}^{k-1} Sc_k(th_r) \quad (51)$$

Où $Sc_k(th_r)$ est le score du seuil associé de x_k et est calculé comme suit:

$$Sc_k(th_r) = \begin{cases} +1 \text{ if } T_i = Y_t \\ -1 \text{ if } T_i \neq Y_t \end{cases} \quad (52)$$

Où Y_t est l'étiquette de classe de x_k , et T_i est la classe conséquence de R^i . Sur la base de (51), $Biais_k$ est égale à $Biais_{k-1} + Sc_{k-1}(th_r)$. Par conséquent, l'intervalle optimal avec la polarisation maximale qui est en fait l'intervalle optimal avec le maximum de précision peut être obtenu en un seul passage sur les seuils triés de th_0 à th_m en $O(m)$, qui est dans le pire des cas $O(n)$ (m est le nombre d'échantillons de données don't-care de la fenêtre et n est la taille de la fenêtre). Après l'obtention de l'intervalle optimal, une valeur de CF_i est sélectionnée dans cet intervalle.

4.4.3 L'ajout d'un nouvel échantillon de donnée:

Insérer un nouvel échantillon de données peut modifier le poids de certaines règles. Si le nouvel échantillon de données est classé sans erreur, nous nous rendons compte que les règles et leurs poids sont déterminés correctement et donc il n'est pas nécessaire de changer le système. Si les poids des règles, avant d'envisager le nouvel échantillon de données, sont optimaux, ils sont encore optimaux si le nouvel échantillon de données est correctement classé. Toutefois, quand il est mal classé par le système, le classifieur est modifié afin d'augmenter l'efficacité du système avec ce nouvel échantillon de données. Dans ce cas, la mise à jour de toutes les règles n'est pas nécessaire. Nous venons de mettre à jour les règles qui ont la même classe conséquence que l'étiquette de classe du nouvel échantillon de données. A cet effet, nous cherchons à trouver dans ces règles, celle dont nous pouvons modifier son poids, de sorte qu'après la mise à jour de ce poids, le système classifie correctement X_{new} . Ce processus s'arrête lorsque toutes les règles mentionnées sont pondérées ou après que la pondération de l'une d'elles, X_{new} est classée correctement.

4.4.4 Suppression de l'échantillon de données le plus ancien :

Après avoir inséré X_{new} , lorsque la mémoire (fenêtre) est pleine, l'un des échantillons de données existantes est supprimé pour localiser ce nouvel échantillon de données. Le retrait de l'échantillon de données le plus ancien, X_{old} , (l'échantillon de données qui est arrivé au système en premier) est la meilleure façon de préserver le système dans l'état de mise à jour. Après l'élimination de X_{old} , le seuil associé de cet échantillon de données th_{old} , est retiré de la liste de seuils. Par conséquent, certains poids des règles ne deviennent pas optimaux après cette suppression et ils doivent être mis à jour. Cependant, il n'est pas nécessaire de mettre à jour le classifieur si X_{old} est mal classé par le système. De plus, nous n'avons pas à mettre à jour toutes les règles dans le cas où X_{old} est classé correctement. Nous mettons à jour que la règle gagnante et toutes les autres règles des classes adverses en considérant X_{old} . Ce processus se poursuit jusqu'à ce que X_{old} soit mal classé par le système.

Dans cette méthode, le mécanisme de pondération de la règle est fait pour les règles qui doivent être mises à jour, considérant les nouvelles circonstances. Il peut changer le poids d'une règle peu à peu par le changement progressif de la densité des données ou encore fixer le poids d'une règle à zéro lorsque la densité d'une classe est complètement enlevée à partir d'une partie de l'espace. Par conséquent, les problèmes de dérive (*Drift*) et de décalage (*Shift*) peuvent être traités par cette méthode. Le mécanisme complet de pondération des règles est illustré dans L'algorithme qui suit.

4.4.5 L'algorithme proposé par Shahparast et al. [2012] :

Initialiser le classifieur par le premier échantillon de données x_k (générer une règle avec x_k)

Régler le poids de la règle généré

Tant que le flux de données n'est pas fini **faire** pour chaque échantillon de données x_k ,
Ajouter x_k à la fenêtre

Si la fenêtre est pleine

ALORS

Retirer l'échantillon de données le plus ancien X_{old} , à partir de la fenêtre

Classifier l'échantillon de données enlevé

Si l'échantillon de données supprimé est classé correctement

ALORS

Régler le poids de la règle gagnante et les autres règles des

Classes adverses par la nouvelle fenêtre jusqu'à ce que X_{old} soit mal classées par le système.

Fin (si)

Fin (si)

Si l'étiquette de classe de x_k n'est pas connue

ALORS

Classifier x_k par le classificateur courant, en utilisant (47), L_i^{\wedge}

Fin (si)

Chaque fois que le véritable label de x_k est spécifié, obtenir ce véritable label l_i

Comparer L_i avec L_i^{\wedge} et calculer la précision du système (phase de test)

Si L_i n'existe pas parmi les labels existants

ALORS

Générer une nouvelle règle avec x_k et régler le poids pour cette règle

Sinon

Classifier x_k par le classifieur courant, en utilisant (47), L_i^{\wedge}

SI x_k est mal classé par le système ($L_i^{\wedge} \neq L_i$)

ALORS

Réajuster le poids de toutes les règles avec la Classe conséquent

De L_i jusqu'à ce que x_k soit classé correctement

Fin (si)

Fin (Si ALORS Sinon)

Fin (Tant que ... Faire)

5. Les domaines d'application des eclass :

eClass fonctionne en auto-développement de sa base de règles floues sur une base par échantillon. Ainsi, sa base de règles floues n'est pas fixe comme avec les classifieurs hors-ligne mais il peut s'adapter à l'information apportée par des échantillons de données qui arrivent du flux de données massives séquentielles. Le processus d'adaptation ne nécessite pas de réapprendre, car il est basé sur des calculs récursifs. Ceci s'oppose à la grande nature itérative des systèmes flous hors-ligne, qui impliquent de nouvelles charges de calcul et qui ne peuvent pas faire face à un grand nombre d'échantillons en entrée. eclass préserve également l'intelligibilité linguistique des règles générées. Toutes ces caractéristiques ont fait qu'eclass a pu être utilisé dans plusieurs domaines différents.

5.1 Diagnostic médical :

Durant les dernières années, il y a eu une explosion dans l'utilisation des systèmes de classification flous dans le domaine du diagnostic médical, cela est dû au fait que la théorie des ensembles flous répond efficacement à l'incertitude non statistique, qui est circonscrite dans les problèmes du domaine médical, et que l'intelligence artificielle offre une très grande panoplie d'outils pour modéliser des solutions aux différents problèmes. Parmi les méthodes floues, il y a l'algorithme eclass qui a été utilisé avec succès dans différents diagnostics médicaux.

Tout d'abord eclass a été introduit par [Xydeas et al., 2005] pour la classification des signaux cérébraux d'électroencéphalogrammes (EEG). Les auteurs avaient testé deux différentes méthodes. La première méthode est basée sur une architecture de réseau modulaire probabiliste qui emploie de multiples Modèles de Markov cachés dépendant (HMM-DM-D) sur les caractéristiques d'entrée (canaux). La deuxième méthode, eClass, est basée sur une base évolutive de règles floues de prototypes de signaux EEG en ligne qui représentent chacune des classes, qui prend en considération la proximité spatiale entre les signaux d'entrée.

Les deux approches utilisent l'apprentissage supervisé, mais diffèrent dans leur mode de fonctionnement. eClass est conçu de manière récursive, en ligne, et possède une structure évolutive, tandis que DM-HMM-D est formé hors ligne, dans un mode basé sur des blocs, et se caractérise par une architecture fixe. Les deux méthodes ont été largement testées sur des données EEG réelles qui sont enregistrées au cours de plusieurs sessions expérimentales, impliquant un seul sujet féminin qui est exposé à une légère douleur induite par un faisceau laser. Les résultats expérimentaux montrent la viabilité des approches proposées et leur potentiel pour résoudre les problèmes de classification similaires.



Figure II- 5 : Bonnet EEG dans sa forme physique monté sur le cuir chevelu d'un sujet

Dans cet article les auteurs ont pu démontrer que le HMM-DM-D est plus performant que le multi-HMM-D conventionnel. Avec une précision de classification douleur / pas de douleur de respectivement 94% et 97%. Comme démontré dans le tableau II-1.

Tableau II-1: Les résultats de classification pour les systèmes IM-HMM-D et DM-HMM-D.

Les performances de classification sont exprimées en pourcentage de Segments EEG correctement classés

Model \ Scale	100		50		20		10	
	Pain	No Pain	Pain	No Pain	Pain	No Pain	Pain	No Pain
IM-HMM-D	41.1765%	97.0588%	40.5882%	97.0588%	47.0588%	91.1765%	52.9412%	91.1765%
DM-HMM-D	91.1765%	97.0588%	97.0588%	94.1176%	97.0588%	94.1176%	55.8824%	92.9412%

Par ailleurs, ils ont montré qu' eClass est beaucoup plus flexible et définit la structure de sa base des règles floues en ligne, en réponse aux signaux EEG d'entrée présentés au système au cours de son processus de formation. Cette caractéristique de base des règles évolutive f est un atout majeur de eClass et un élément important de différenciation par rapport à d'autres systèmes de classification à structures off-line, fixes, car il permet de montrer de façon significative, la manière dont le classifieur peut être utilisé dans des applications en temps réel. En outre, les résultats expérimentaux ont clairement démontré le potentiel du système de eClass dont la méthodologie porte la promesse de nouveaux développements significatifs de recherche sur les systèmes de classification superviséon-ligne.

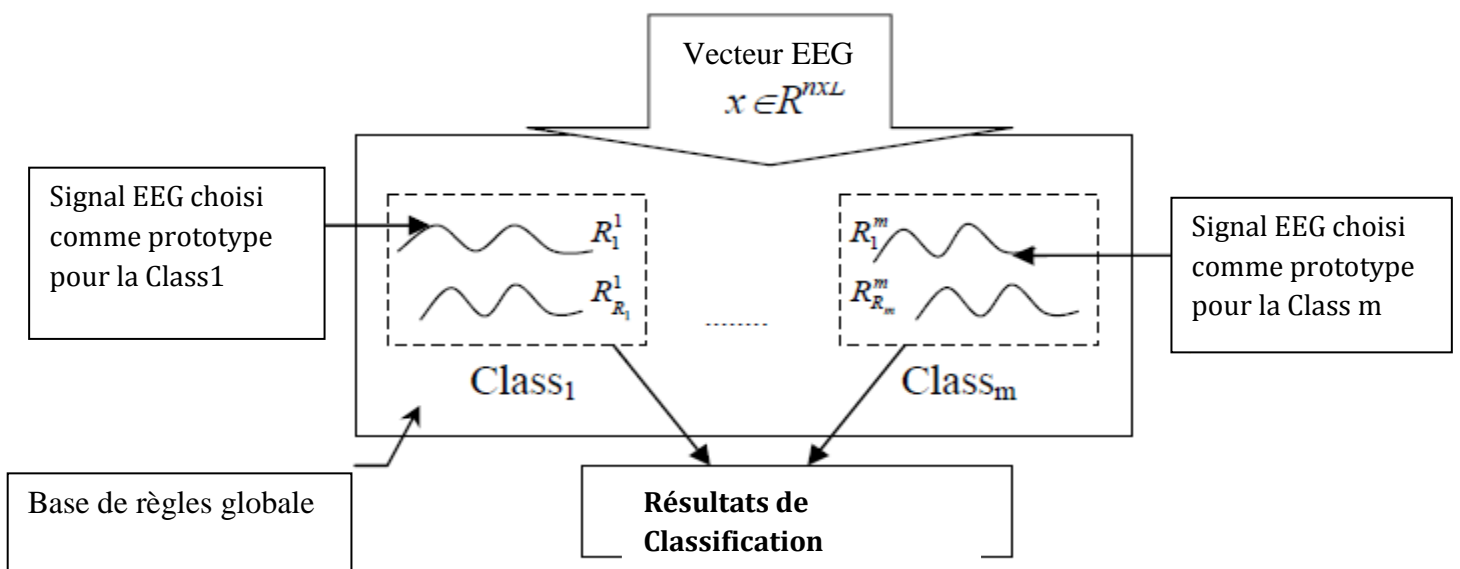


Figure II-6 : Modèle général du système eClass

Ensuite en 2006, toujours dans le domaine médical, eclass a été appliqué dans le diagnostic du cancer du sein par Lekkas et Mikhailov [Lekkas et Mikhailov, 2008]. Pour cela ils ont utilisé le multiple-input-multiple output (MIMO) Takag iSugeno Kang (TSK) fondé sur une base des règles de premier ordre. En outre, ils ont introduit une nouvelle fonction de notation de caractéristiques F-scores, qui identifie les caractéristiques les plus précieuses des données en temps réel. Leurs expériences montrent que l'algorithme retourne un taux de classification élevé et les résultats sont comparables à d'autres approches qui font l'apprentissage à partir d'observations numériques de nature médicale.

Ils ont fait leurs expériences sur la base de données WBCD qui contient des données réelles, provenant de ponction de l'aiguille du tissu mammaire humain. Cette base se compose de 699 cas, dont ils avaient supprimé seize entrées, qui contiennent des valeurs manquantes. Il existe neuf fonctions par exemple (F1,..., F9), qui correspondent à l'épaisseur du massif, l'uniformité de la taille des cellules, l'uniformité de la forme des cellules, l'adhésion marginal, la taille des cellules épithéliales unique, noyaux nus, fade chromatine, les noyaux normaux et mitoses respectivement. Ceux-ci sont représentés sous forme de nombres entiers dans l'intervalle [1,10].

Tableau II-2 : Résultats de l'utilisation d'eClass sur WBCD par rapport aux neuf caractéristiques.

Modèle	Train-teste	Nombre de paramètres	Nombre de règles	précision	Temps d'exécution
#1	70% - 30%	9	7	99.37%	516 ms
#2	80% - 20%	9	9	99.45%	687 ms

Le tableau II-2 montre la précision de classification pour les deux modèles d'eClass construites séparément. En ce qui concerne la première configuration, ils ont utilisé 70% des données pour l'apprentissage du classificateur et le reste 30% pour tester son exactitude. Pour le second, 80% ont été utilisés dans le cadre de l'apprentissage et le reste de 20% pour les essais.

L'étape suivante a consisté à obtenir les F-scores (les notations). Cette procédure a eu lieu avec l'ordre des données et la taille utilisée pour construire les modèles # 1 et # 2. Les résultats sont affichés dans le tableau II-3.

Tableau II-3 : Notes de fonction et de classement pour les deux modèles

Modèles	F1	F2	F3	F4	F5	F6	F7	F8	F9
#1	0.2978	0.3030	0.3285	0.1871	0.1767	0.3015	0.2575	0.2465	0.0739
#2	0.2946	0.3194	0.3450	0.1899	0.1771	0.3033	0.2826	0.2389	0.0635
classement	4	2	1	8	7	3	5	6	9

Après qu'ils aient attribué une note à chaque dimension en utilisant F-score, ils ont classé les fonctionnalités en fonction de leur importance et ont créé six configurations supplémentaires par modèle, chacun utilisant une fonctionnalité de différente taille de l'ensemble. Les précisions de ces modèles sont affichées dans le tableau 4. En comparant les précisions des

tableaux 2 et 4, on peut observer que les deux modèles d'eClass produisent la plus grande précision lorsque l'ensemble de données contient les neuf caractéristiques. Notamment, le second modèle indique la même précision lorsque la base de données contient les huit caractéristiques les plus importantes des neuf ; ou tous les neuf.

Tableau II-4 : Précision des deux modèles utilisant différents sous-ensembles de caractéristiques

Modèles	taille de l'ensemble des Caractéristiques					
	3	4	5	6	7	8
#1	96.86%	98.32%	97.28%	98.53%	98.95%	99.16%
#2	97.07%	98.53%	97.43%	98.71%	99.26%	99.45%

Les résultats obtenus indiquent des résultats très précis et une performance supérieure en comparaison avec d'autres méthodes efficaces dans la littérature. Les mêmes auteurs ont publié en 2010 un autre travail [Lekkas et Mikhailov, 2010], toujours dans le domaine médical. Ils ont utilisé cette fois-ci l'algorithme eclass pour le diagnostic de deux maladies très fréquentes à savoir le diabète et les maladies dermatologiques.

Les ensembles de données PID et DERM, qui sont examinés dans cette étude, sont disponibles auprès de l'UCI Machine Learning Repository à [Pima Indians Diabetes dataset. 2008], [Dermatology dataset. 2008] respectivement. Le jeu de données PID contient 768 échantillons de données et 8 fonctions numériques par échantillon (échantillon dimensionnalité). Chacun des échantillons comporte une étiquette qui indique la classe de l'échantillon. Il y a deux classes au total. La première classe est étiquetée comme " négative " au diabète et implique 500 échantillons (65,10% de l'ensemble de données), tandis que la seconde est étiquetée comme " positive " au diabète et implique 268 échantillons (34,89% de l'ensemble de données). Le fait qu'il s'agit d'un problème de classification binaire permet l'utilisation de la sensibilité et de mesures de la qualité de spécificité, en plus de la précision.

L'ensemble de données DERM contient 366 échantillons de données, dont 8 ont été supprimés car ils contiennent des valeurs manquantes. Chaque échantillon se compose de 34 éléments numériques, dont 12 sont de nature clinique et 22 de nature histopathologique. La dimension de cet ensemble de données est suffisamment grande pour souligner l'algorithme d'eClass et pour déterminer son efficacité dans le traitement en temps réel. Les échantillons appartiennent à l'une des six classes du problème.

A partir d'expériences menées sur l'algorithme d'eClass, il a été observé que la moyenne de précision du modèle augmente pour les modèles d'ordre de données spécifiques. Plus précisément, si les échantillons sont réorganisés en morceaux qui peuvent être indiqués par leur étiquette de classe unique, alors la précision du modèle va progressivement augmenter. Partant de ce constat, une technique de mise en mémoire tampon a été proposée dans cet article, ce qui suppose que les échantillons sont étiquetés. Il est semblable à celui décrit par [Aggarwal et al., 2004], dans le sens où les échantillons sont traités dans des fenêtres de temps et non en groupe.

En ce qui concerne l'ensemble de données du diabète des Indiens Pima, une précision de 79,37% a été obtenue, qui est de 0,84% de moins que le plus haut dans la littérature. La stratégie d'optimisation proposée a augmenté la précision et la spécificité du modèle de 4,05% et 7,63% respectivement. Pour l'ensemble de données dermatologiques, une précision de 97,55% a été obtenue, qui est de 1,65% de moins que le plus haut dans la littérature. Dans ce cas, la stratégie d'optimisation proposée a amélioré la précision du modèle de 4,82%. L'algorithme a été amélioré par rapport à d'autres algorithmes existants et semble surpasser la majorité.

L'évolution des structures à base de règles floues représente des méthodes extrêmement puissantes pour le classement en ligne de flux de données. Les règles floues sont générées, modifiées et supprimées automatiquement dans ces systèmes. Un des algorithmes simples mais efficaces de ce type est le classifieur eClass qui construit les règles sans aucune connaissance préalable, à partir de zéro. Cependant, cet algorithme ne peut pas faire face correctement aux concepts *Drift* et *Shift* de données. En 2012 Shahparast, et ses collaborateurs [Shahparast et al., 2012] ont proposé une nouvelle méthode en ligne efficace, pour augmenter les performances de cet algorithme, en fixant un poids approprié pour chaque règle et gérer les problèmes du *Drift* et *Shift* des données. En ajustant les poids appropriés, la zone d'influence de chaque règle peut être facilement contrôlée et changée en prenant compte le changement de l'environnement. Cet algorithme d'ajustement de poids est basé sur une méthode de réglage de poids en mode batch efficace, elle est conçue pour être compatible avec les caractéristiques du flux de données. L'algorithme proposé est évalué sur certains ensembles de données standard de l'UCI Repository et des flux de données du monde réel, comparé avec l'algorithme eClass. Les résultats montrent que l'algorithme proposé surpasse l'approche eClass, et a une amélioration significative dans la plupart des cas.

5.2 ROBOTIQUE :

L'une des tâches les plus difficiles de la robotique est le problème de *navigation autonome*, où un robot (ou un ensemble de robots) doit pouvoir accomplir certaines tâches sans aucune indication extérieure ; par exemple l'auto-localisation. Pour un robot, la capacité de s'auto-localiser et de reconnaître les sites est essentielle pour survivre et fonctionner efficacement. La disponibilité des coordonnées globales, un lien GPS et la communication sans restriction est souvent compromise par un certain nombre de facteurs, tels que les interférences, la météo et les objectifs de mission.

En 2006 puis en 2007 Zhou et Angelov [Zhou et Anvelov, 2006] ; [Zhou et Anvelov, 2007] ont publié des travaux qui visent à utiliser l'algorithme eclass dans le domaine de la robotique mobile.

Ces travaux consistaient à développer un classifieur évolutif flou basé sur l'auto-organisation, par étiquetage automatique des points de repère qui sont détectés en temps réel. L'approche proposée permet la détection et la reconnaissance de repères en étant entièrement autonome et en mode non-supervisé et cela sans l'utilisation des coordonnées absolues, tout lien de communication ou tout prétraitement. L'algorithme proposé est récursif, non itératif, en un seul passage et donc de calculs peu coûteux et adaptés aux applications en temps réel. Des

simulations approfondies ainsi que des tests de la vie réelle ont été réalisées dans un environnement intérieur (un bureau situé à InfoLab21, Université Lancaster) en utilisant Pioneer3 plate-forme DX, robotique mobile équipé de sonars et de détecteurs de mouvement et PC embarqué. Les résultats indiquent des taux supérieurs de reconnaissance, de flexibilité, et en termes de calcul de l'approche proposée en comparaison avec les méthodes similaires publiées antérieurement.



Figure II-7 : Pioneer 3DX. robot mobile autonome effectuant la détection des points de repère et leur reconnaissance

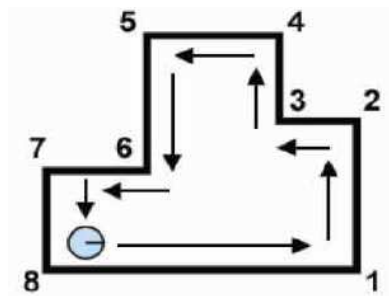


Figure II-8 : l'enceinte expérimentale (Bureau de test, B-69, InfoLab21, Lancaster) bureau avec 8 coins (6 concave et convexe 2)

- Le robot avait réussi à détecter et reconnaître 7 repères sur 8 possible.

5.3 Reconnaissance en temps réel des activités humaines :

La reconnaissance en temps réel des activités humaines (HAR) est également un domaine de recherche intensive [Robertson et Reid, 2006]; [Chen et Aghajan, 2006]. Très souvent, elle s'appuie sur les appareils photo et les signaux vidéo, mais ces méthodes ont souvent des problèmes liés à l'occlusion de la personne dans le champ de la caméra, choisir un bon point de vue pour suivre la personne, les connotations de la vie privée.. etc. Plus récemment, les capteurs portables ont offert une opportunité très prometteuse et attrayante pour recueillir des informations directement à partir du corps humain. Cette approche a l'avantage de contrôler l'humain / l'objet en permanence, sans empiéter sur sa vie privée.

En 2011 Andreu, Baruah et Angelov [Andreu et al., 2006] ont développé un système pour la reconnaissance de l'activité humaine en temps réel (HAR), en utilisant un classifieur évolutif à

base de règles floues (eClass) qui est en auto-apprentissage. Une version réursive de l'analyse en composantes principales (ACP) et l'analyse discriminante linéaire (LDA) où des méthodes de pré-traitement sont couplées avec le eClass, ce qui a conduit à une nouvelle approche pour HAR qui ne nécessite pas de calcul et de temps de pré-formation et des données provenant de nombreux sujets. La nouvelle méthode proposée pour le HAR évolutif (eHAR) prend en compte les spécificités de chaque utilisateur et de l'évolution possible dans le temps de ses habitudes. Les flux de données provenant de plusieurs dispositifs portables (figure II-9 et II-10) permettent de développer une intelligence omniprésente en leur permettant de personnaliser et de s'adapter à l'utilisateur spécifique .



Figure II-9 : Sunspot (sur une main et attaché au bras, la taille 63x38x25mm)

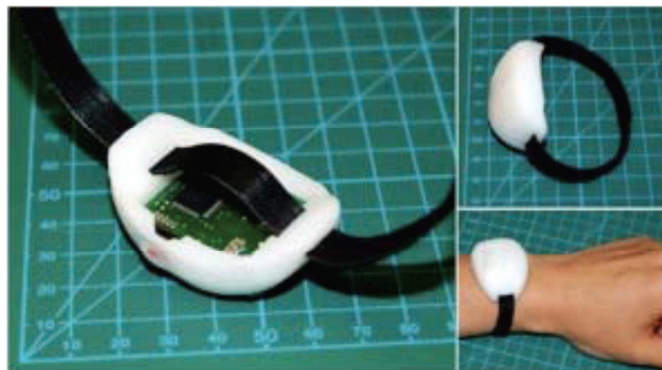


Figure II-10 : Capteur Porcupine v2.5 avec capteur modulaire et boîtiers en plastique (Source de l'image: Embedded Sensing System Group, TU Darmstadt, Germany)

L'évaluation et l'expérimentation ont montré que les résultats sont très prometteurs. Ce nouveau paradigme en évolution appliquée aux systèmes de reconnaissance d'activités humaines, offre de bonnes capacités d'auto-apprentissage et d'auto-adaptation. eHAR, est très compétitif en comparaison avec les approches existantes off-line et progressives. Cependant, les approches hors ligne nécessitent une énorme quantité de données d'entraînement, du temps et des ressources de calcul, ils sont donc moins pertinents pour les nouveaux utilisateurs. Les approches incrémentales sont dépassées par eHAR, mais plus important encore, elles n'ont pas la structure évolutive qu'offre l'approche proposée.

6. CONCLUSION :

Dans ce chapitre nous avons présenté l'approche des classifieurs à base de règles floues en général. Nous avons comparé cette approche et l'approche des algorithmes génétiques et évolutionnaires. Ensuite, nous avons comparé avec les algorithmes incrémentales. Nous avons vu également les différentes variantes de l'algorithme eclass (evolving classifier) ainsi que les améliorations proposées par différents auteurs dans différents domaines d'application. Finalement nous avons présenté quelques domaines où l'algorithme eclass a été appliqué avec succès.

Ce chapitre nous a permis également de voir que l'algorithme eclass est très efficace pour la classification online et qu'il est approprié pour les applications en temps réel. Les résultats obtenus dans presque tous les travaux cités dans ce chapitre sont satisfaisants, eclass semble surclasser la plupart des méthodes décrites dans la littérature.

Chapitre III :

Le cœur et l'électrocardiographie

L'activité cardiaque et les arythmies

1. Introduction

Ce chapitre présente le contexte d'étude : la reconnaissance automatique d'arythmies cardiaques. Il présente le mécanisme à l'origine de l'activité cardiaque. Le principe de l'électrocardiogramme et les problèmes accompagnants son enregistrement y sont également introduits.

Les battements cardiaques, dont la morphologie et la fréquence sont relatifs aux principales arythmies, sont passés en revue pour permettre de comprendre les problèmes de quantification de battements ainsi que leur reconnaissance automatique.

NB : Dans ce chapitre, l'introduction à la cardiologie est volontairement limitée aux notions utiles à la compréhension des chapitres suivants. Le lecteur intéressé pourra cependant trouver des informations complémentaires détaillées dans [Clifford et al., 2006] et [Acharya et al., 2007] et de manière plus intuitive dans [Blondeau et Hiltgen, 1980] et [Marieb, 1999].

2. Le système cardiovasculaire

Le cœur et les vaisseaux sanguins constituent le système cardiovasculaire (Figure III-1). Propulsé par le cœur, le sang circule dans tout l'organisme à travers un vaste réseau de vaisseaux sanguins, avec un flux adéquat continu et sous pression suffisante aux organes et aux tissus. Trois types de vaisseaux assurent le transport du sang : les artères, les capillaires et les veines. Les contractions rythmiques du cœur propulsent le liquide rouge dans les artères ; celles-ci acheminent le sang oxygéné du cœur et les substances nutritives à toutes les régions de l'organisme. Les capillaires, de minuscules vaisseaux, permettent ensuite les échanges entre le sang et les cellules grâce à leur paroi extrêmement fine. Le sang est ensuite réacheminé vers le cœur par les veines.

Les vaisseaux sanguins forment un réseau dont la longueur totale atteint 150 000 km ! Ils se répartissent en deux circuits distincts : la circulation pulmonaire et la circulation systémique. La circulation pulmonaire assure les échanges gazeux entre le sang et l'air contenu dans les poumons. La circulation systémique assure l'irrigation sanguine de tous les organes et tissus. Lorsque le cœur se contracte, ses deux ventricules éjectent simultanément le sang dans les deux circuits (figure III-2).

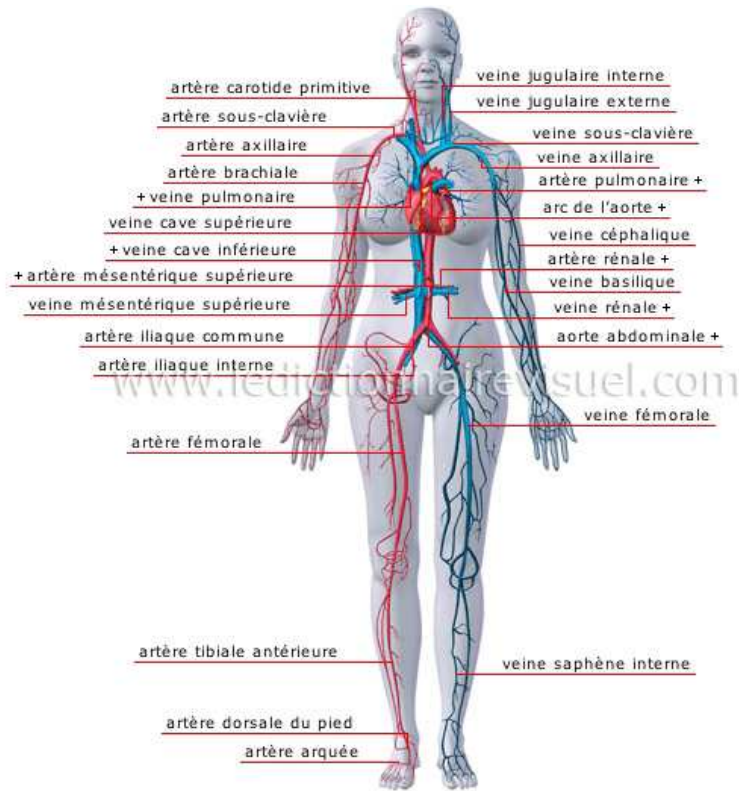


Figure III-1 : Principales veines et artères [www.ledictionnairevisuel.com. 2014]

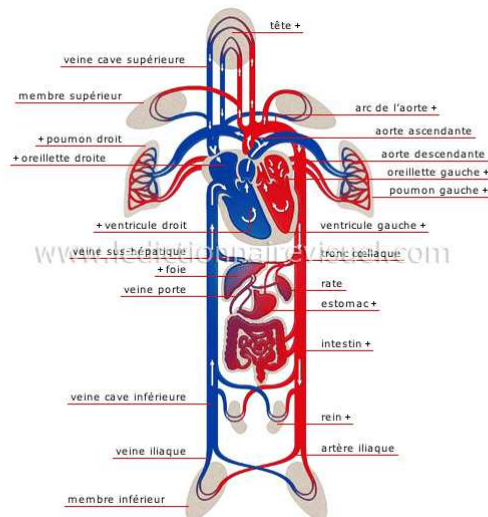


Figure III-2 : schéma de la circulation sanguine [www.ledictionnairevisuel.com. 2014]

3. Anatomie du cœur

Le cœur est un organe vital. Il propulse le sang et le fait ainsi circuler dans l'ensemble des vaisseaux sanguins du corps. Logé au centre gauche de la cage thoracique, entre les poumons dans un espace appelé médiastin. Il est placé à l'oblique, sa partie supérieure légèrement à

droite, de sorte que les deux tiers environ de sa masse se trouvent du côté gauche du corps. Le cœur se contracte en moyenne 70 fois par minute, soit plus de 100 000 battements par jour, propulsant chaque jour quelque 7000 litres de sang dans le système cardiovasculaire. Cet organe est essentiellement formé d'un muscle, le myocarde, qui délimite quatre cavités : deux oreillettes et deux ventricules. Les oreillettes reçoivent le sang tandis que les ventricules, plus gros, l'expulsent. Les ventricules sont fermés par des valves cardiaques, des structures élastiques fines qui s'ouvrent pour permettre le passage du sang, puis se ferment pour éviter qu'il ne reflue (figure III-3).

Les bruits du cœur que le médecin entend grâce à son stéthoscope, sont produits par la fermeture des valvules du cœur. Le premier « boum », fort et résonnant, correspond à la fermeture des valves auriculo-ventriculaires. Le second « boum », plus bref, annonce la fermeture des valves semi-lunaires. Tout bruit anormal peut signaler une défectuosité des valves et doit être pris au sérieux.

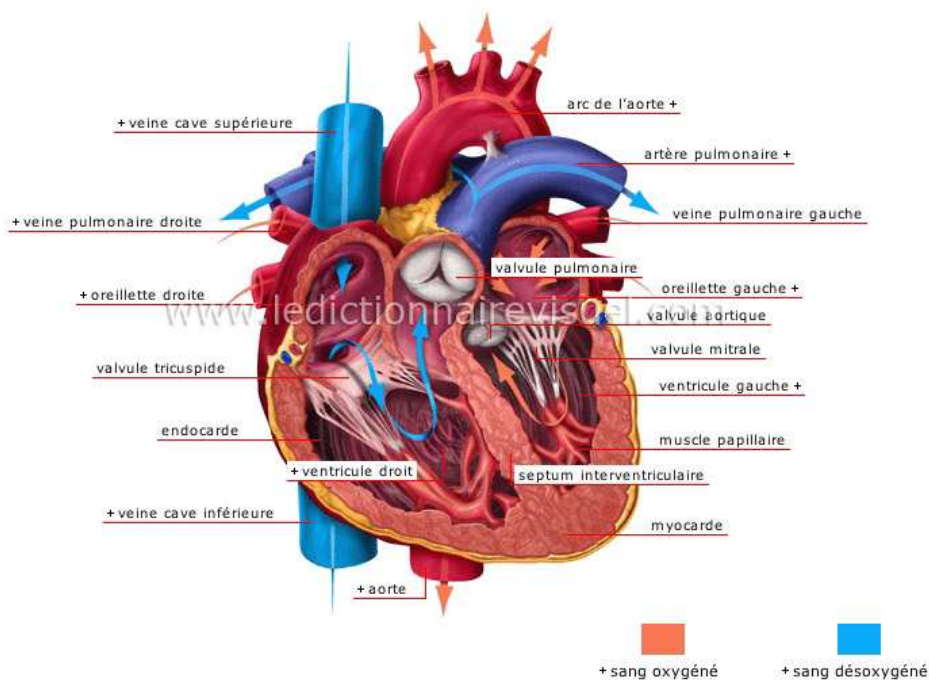


Figure III-3 : Anatomie du cœur [www.ledictionnairevisuel.com. 2014]

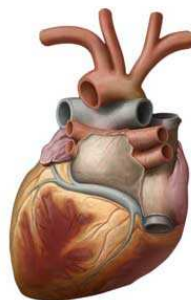


Figure III-4 : Vue postérieure du cœur [www.ledictionnairevisuel.com. 2014]

Le cœur ressemble à un poing fermé, il est parcouru par des sillons le long desquels s'étendent les vaisseaux coronaires (figure III-4). Le cœur d'une personne a à peu près la même grosseur que son poing. Chez un adulte, il mesure de 10 à 12 cm de hauteur, 8 à 9 cm de largeur et 6 cm d'épaisseur ; il pèse environ 300 grammes [Le dictionnaire visuel, 2013].

4. Activité mécanique cardiaque

Le cycle de la circulation sanguine se répète constamment et comprend deux périodes: la systole et la diastole. La systole est la période correspondant à l'éjection du sang dans la grande et petite circulation (Figure III-5). Elle se décompose en trois phases: la systole auriculaire, la contraction ventriculaire isovolumique et la systole ventriculaire [Zipes et Jalife, 2004].

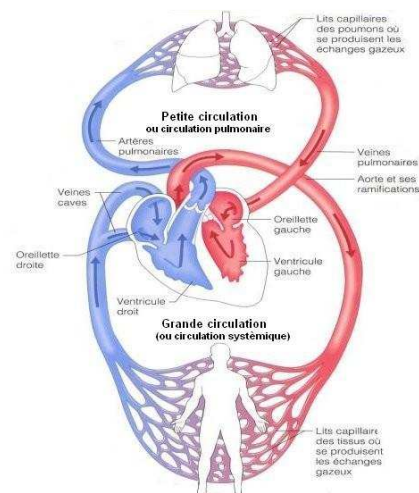


Figure III-5 : grande et petite circulation sanguine

- La systole auriculaire est la contraction des oreillettes lorsque celles-ci sont remplies de sang. La pression exercée par le muscle auriculaire ferme les orifices des veines caves et pulmonaires et provoque le versement du sang auriculaire dans les ventricules par les valves tricuspide et mitrale (figure III-6). Cependant, la majorité du remplissage des ventricules survient passivement pendant la diastole.

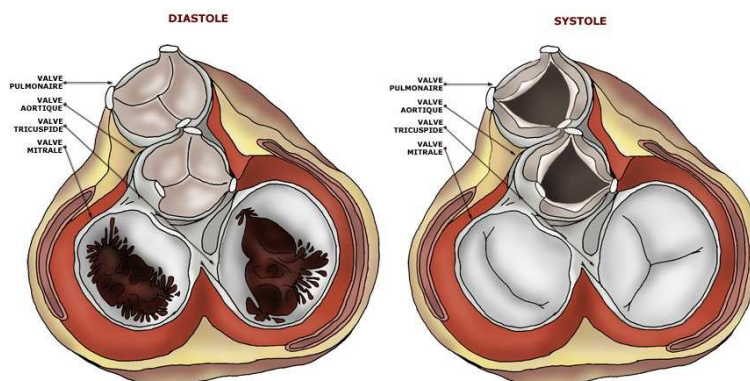


Figure III-6 : les différentes valves du cœur [http://www.stethographics.com, 2013]

- La contraction ventriculaire isovolumique commence lorsque les cellules musculaires du myocarde ventriculaire se contractent. La pression ferme les valves tricuspide et mitrale, les valvules sigmoïdes restent fermées pendant quelques dizaines de centièmes de secondes. Comme il n'y a pas d'éjection ventriculaire, la pression augmente fortement dans les ventricules.

- La systole ventriculaire commence lorsque les pressions dans les ventricules dépassent les pressions dans l'artère pulmonaire et l'aorte. Les valvules sigmoïdes s'ouvrent et l'éjection commence. Le sang oxygéné emprunte la crosse aortique et le sang désoxygéné le tronc pulmonaire. La pression aortique atteint un pic (pression artérielle systolique) puis redescend jusqu'à la fin de la systole.

La diastole est la phase de relaxation du cœur pendant laquelle il se remplit de sang. Cette période est composée de deux phases: la relaxation ventriculaire isométrique et la phase de repos.

- La relaxation ventriculaire fait suite à la systole. Les ventricules se relâchent, la pression chute jusqu'à être inférieure à celle exercée dans l'aorte et l'artère pulmonaire. En conséquence, les valvules sigmoïdes se ferment et, lorsque la pression devient inférieure à celle des oreillettes, les valves tricuspide et mitrale s'ouvrent.

- La phase de repos est celle pendant laquelle le sang des veines caves et pulmonaires s'écoule librement dans les ventricules via les oreillettes. La figure qui suit présente les différentes étapes du cycle cardiaque.

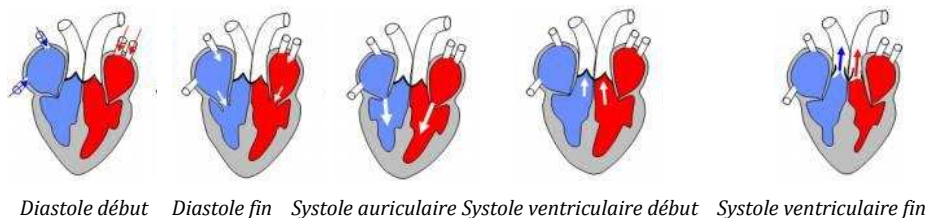


Figure III-7 : le cycle cardiaque. [zidelmal, 2012]

5. Activité électrique du cœur

Le rythme cardiaque possède deux composantes :

-une composante mécanique, cycle cardiaque, succession des phases de contraction (systole) et de relaxation (diastole) ;

-une composante électrique, directement responsable de la phase mécanique avec laquelle elle est parfaitement synchronisée.

La contraction des cellules musculaires cardiaques(ou myocarde) est provoquée par le passage d'une onde électrique parcourant le cœur à travers les oreillettes et les ventricules. Le courant électrique naît en un point précis du cœur (de l'ordre de quelques millimètres de diamètre), appelé nœud sinusal ou « pacemaker » (l'appareil du même nom joue le même rôle d'initiation du courant de contraction), situé au sommet de l'oreillette droite.

Cette source est constituée d'un amas de cellules capables de fabriquer un courant électrique de quelques millivolts. Partant du nœud sinusal, le courant se propage en tache d'huile dans le muscle cardiaque. Il circule dans les deux oreillettes jusqu'à leur base, provoquant leur contraction, d'où il converge vers la cloison séparant oreillettes et ventricules, au niveau d'un relais électrique appelé nœud auriculo-ventriculaire (NAV).

A partir du NAV, l'influx progresse dans les deux ventricules, droit et gauche simultanément, empruntant les voies conductrices très rapides que sont le faisceau de His et le réseau de Purkinje, jusqu'à la pointe du cœur, provoquant alors la contraction des ventricules (figure 8).

De manière plus claire et plus détaillée, L'activité électrique normale du cœur suit la séquence d'activation suivante :

- Le nœud sinusal (NS) :

L'activité électrique est générée spontanément dans le nœud sinusal qui est situé dans la partie haute de la paroi intérieure de l'oreillette droite, au niveau où débouche la veine cave supérieure. L'impulsion cardiaque initiée dans le nœud sinusal est transmise aux deux oreillettes. Cette activation est facilitée au moyen des voies spécialisées inter nodales qui relient le nœud sinusal au nœud auriculo-ventriculaire.

- Le nœud auriculo-ventriculaire (NAV) :

Il est situé en bas de l'oreillette droite et est constitué de cellules qui présentent une conduction électrique lente. L'activation électrique qui arrive au NAV est physiologiquement ralentie (approximativement 100 ms) avant d'arriver au faisceau de His. Cette propriété physiologique du NAV permet de protéger les ventricules d'un nombre excessif d'activations du NAV et d'activations auriculaires, elle concède aux oreillettes un temps de vidange plus grand, ce qui optimise la contraction ventriculaire.

- Le faisceau de His :

Il est situé dans la partie haute du septum inter ventriculaire, ses fibres traversent le tissu connectif (non excitable) qui sépare électriquement les oreillettes des ventricules. Dans les cas normaux, le NAV et le faisceau de His constituent la seule voie de propagation de l'activité électrique cardiaque entre les oreillettes et les ventricules. L'ensemble de ces deux structures est souvent appelé la jonction auriculo-ventriculaire. Le faisceau de His comprend un tronc initial qui se divise en deux branches, droite pour le ventricule droit et gauche pour le ventricule gauche.

- Les fibres de Purkinje :

Les branches du faisceau de His finissent dans un réseau de fibres qui arrivent dans les parois ventriculaires. Les fibres de Purkinje terminent en anastomoses avec les fibres myocardiques musculaires, facilitant leur excitation.

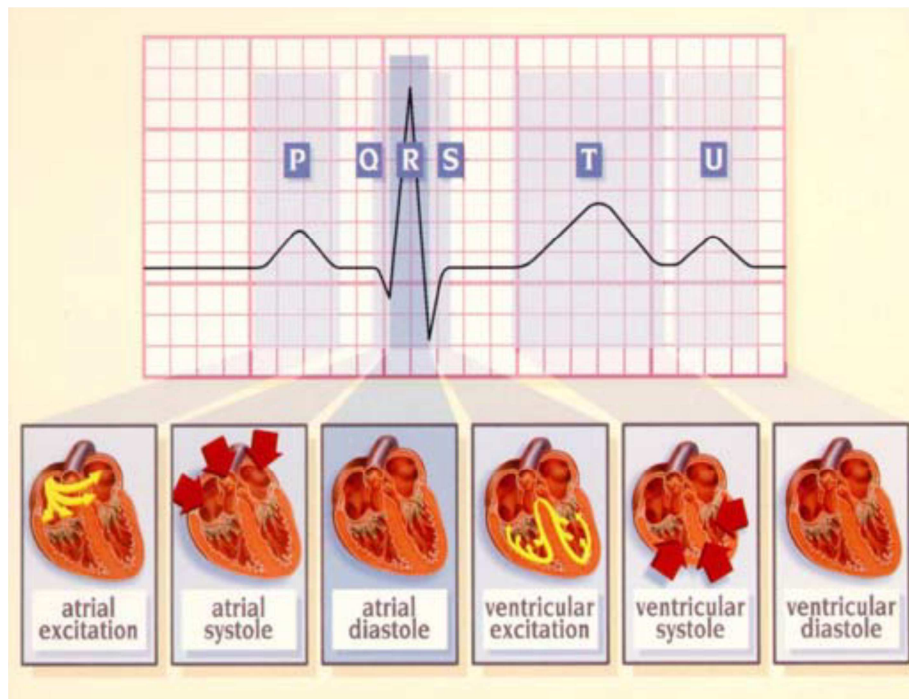


Figure III-8 : Parcours de l'onde électrique et phases de la contraction cardiaque

– effets sur l'ECG-[IRISA. 2005]

6. L'ELECTROCARDIOGRAMME (ECG)

Au cours des années 1880, E Marey [Marey, 1876] et Augustus Waller [Waller, 1893] avaient montré que l'activité électrique du cœur, découverte quelques années plus tôt, pouvait être suivie à partir de la peau; le corps humain étant électriquement conducteur, les potentiels d'action générés lors de l'activité électrique cardiaque peuvent être recueillis par des électrodes placés sur le corps. Puis, vers 1890, Willem Einthoven a réalisé le premier enregistrement cardiographique [Einthoven, 1941]. L'enregistrement de l'activité électrique du cœur se fait sur un plan frontal (par les dérivations périphériques) ou sur un plan horizontal (par les dérivations précordiales) selon l'emplacement des électrodes. La représentation graphique d'une telle activité est appelée : ElectroCardioGramme (ECG) ou (EKG pour les Anglo-saxons).

L'ECG est l'enregistrement des potentiels électriques parcourant le cœur. La stimulation cardiaque par le nœud sinusal entraîne la dépolarisation des cellules du cœur, de proche en

proche, depuis les oreillettes jusqu'aux ventricules. La propagation des potentiels d'action au sein du cœur correspond à la propagation d'une zone de dépolarisation (*de charge électrique positive*) dans un myocarde repolarisé (*négatif*). Puis suit la propagation d'un front de repolarisation (*électriquement négatif*) dans un myocarde complètement dépolarisé (*positif*). C'est donc l'importance, la vitesse, la direction moyenne du front de dépolarisation qui sont enregistrés par l'ECG. Cela revient donc à enregistrer le vecteur représentant la différence de potentiel entre les deux fronts positif et négatif.

Ainsi, au repos, un cœur complètement dépolarisé (*tout positif*) ou repolarisé (*tout négatif*) donnera un enregistrement « nul » puisqu'il n'y a pas de différence de potentiel, le tracé correspondant sera la ligne de base, la ligne isoélectrique. Une électrode voyant le front de dépolarisation se rapprocher enregistre un signal positif. Et, à l'inverse, si ce front s'éloigne, on a un signal négatif (Figure III-9) [Sende, 2009].

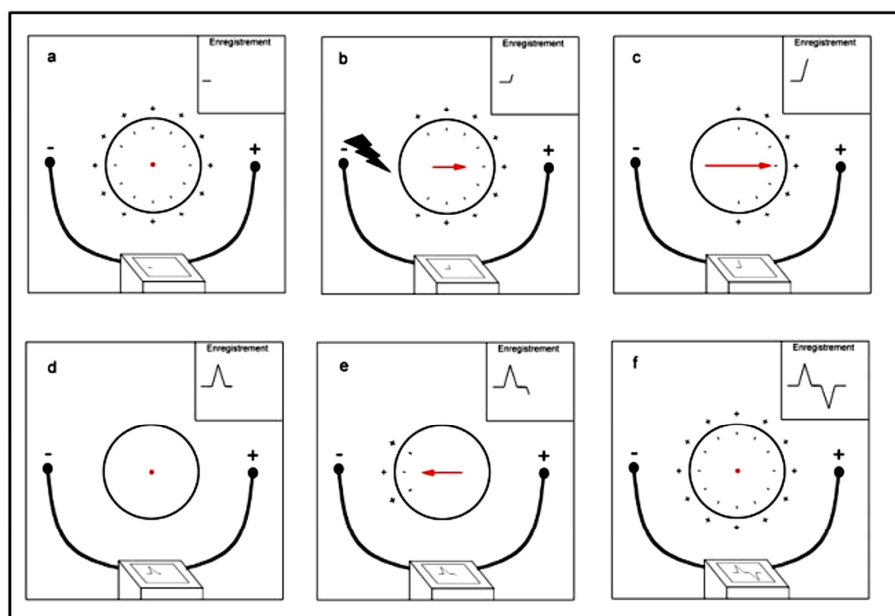


Figure III-9 : Electrophysiologie cellulaire [Dubois, 2004]

Considérons une cellule unique ; au repos, celle-ci est chargée uniformément ; le vecteur de dépolarisation est nul (point rouge), le tracé est donc plat (a). Une stimulation extérieure du côté gauche induit une perte de charge de ce côté ; l'impulsion électrique se propage alors de gauche à droite. Le vecteur de dépolarisation associé (flèche rouge) est orienté de l'électrode négative vers l'électrode positive : l'enregistrement présente donc une déflexion positive (b) qui est maximale lorsque la dépolarisation a atteint le milieu de la cellule. La fin de la dépolarisation se traduit par une pente descendante (c), car le vecteur est toujours orienté dans le même sens, mais son amplitude diminue. Une fois la cellule dépolarisée, le tracé est plat (d). La repolarisation de la cellule se traduit par un vecteur électrique orienté dans le sens opposé au précédent ; le tracé présente donc, dans un premier temps, une déflexion négative (e) pour ensuite redevenir plat (f) [Dubois, 2004].

6.1 Les dérivations

Les modalités d'enregistrements sont variées. Elles se distinguent selon l'emplacement des électrodes sur la surface du corps. L'enregistrement de plusieurs tracés (projection du signal sur diverses lignes est du corps est appelé : système de dérivations ECG).

Généralement les appareils électrocardiographiques peuvent enregistrer plusieurs différences de potentiel en même temps, selon l'emplacement et le nombre d'électrodes réparties sur le thorax et les membres. Chaque mesure de ces potentiels correspond à une dérivation de l'ECG. Un système de dérivations consiste en un ensemble cohérent de dérivations, chacune étant définie par la disposition des électrodes sur le corps du patient. L'emplacement des électrodes est choisi de façon à explorer la quasi-totalité du champ électrique cardiaque.

Si on mesure le vecteur cardiaque dans une seule direction, on ne sera pas en mesure de le caractériser entièrement. Il est donc important d'avoir un standard de positionnement des électrodes (dérivations) pour l'évaluation clinique du signal ECG. En pratique, douze dérivations sont utilisées dans les plans frontal et transversal pour explorer l'activité électrique du cœur. On distingue :

✚ Trois dérivations bipolaires (ou dérivations standard) :

Elles ont été déterminées par Einthoven et sont appelées bipolaires car le potentiel est mesuré entre deux électrodes. Elles sont effectuées à partir des potentiels du DI, DII, DIII, obtenues par permutation des électrodes placées sur le bras droit, le bras gauche et la jambe gauche de la manière suivante :

$$DI = VL - VR$$

$$DII = VF - VR$$

$$DIII = VF - VL$$

Où VR correspond au potentiel au bras droit, VL correspond au potentiel au bras gauche et VF correspond au potentiel dans la jambe gauche. La jambe droite est reliée à la masse. Les vecteurs obtenus forment alors un triangle équilatéral appelé triangle d'Einthoven comme donné sur la figure (III-10).

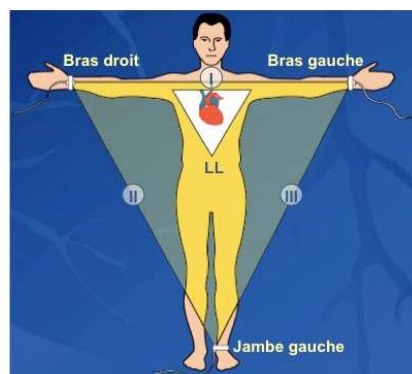


Figure III-10 : Triangle D'Einthoven.[Talbi, 2011]

Triangle équilatéral formé par les électrodes situées sur les deux avant-bras et la jambe gauche, et au centre duquel se situe approximativement le cœur. Ces électrodes permettent l'enregistrement de dérivations frontales DI, DII et DIII avec $DI + DIII = DII$. Ce triangle est le standard conventionnel pour étudier l'activité électrique du cœur sous des angles différents et calculer son axe électrique dans le plan frontal.

✚ Trois dérivations unipolaires aVR, aVL, aVF :

Les dérivations unipolaires des membres permettent d'étudier l'activité électrique du cœur sur le plan frontal. Elles ont été déterminées par Wilson. [Wilson et al., 1934]. Ces dérivations permettent de mesurer la tension entre un point de référence et le bras droit, le bras gauche et la jambe gauche respectivement. Le point de référence est réalisé par la moyenne des signaux qui apparaissent sur les deux autres membres qui ne sont pas en observation. A cet effet, on utilise des résistances de valeur élevée, supérieure à $5M\Omega$. La figure (III-11) montre les dérivations unipolaires.

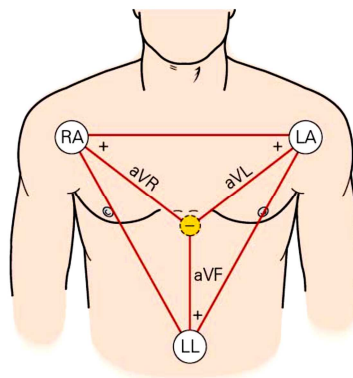


Figure III-11 : Dérivations unipolaires. [Talbi, 2011]

✚ Six dérivations précordiales :

Ce sont des dérivations unipolaires mises au point par Wilson [Wilson et al., 1934]. Elles sont posées sur le thorax et sont désignées par la lettre V suivie du numéro de leur emplacement. Le potentiel de l'électrode exploratrice est pris par rapport à la moyenne des potentiels VL, VR et VF. Six points, définis par Wilson, permettent d'obtenir les dérivations V1 à V6. Leur emplacement est représenté sur la figure (III-12).

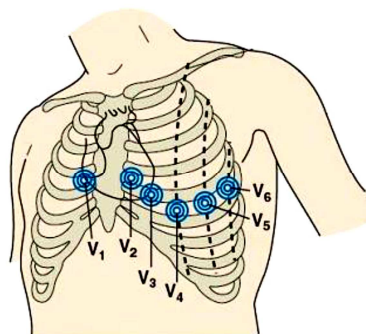


Figure III-12 : Dérivations thoraciques. [Talbi, 2011]

6.2 CONDITIONS D'ENREGISTREMENT

Pour un enregistrement de bonne qualité, le respect de certaines conditions est nécessaire. Le patient doit être couché sur le dos, en résolution musculaire complète, dans une position confortable et protégé du froid afin d'éliminer au maximum les ondulations de la ligne de base et les parasites dus aux tremblements musculaires ou au mauvais contact électrode-peau.

6.3 LE HOLTER

C'est une technique d'enregistrement de l'activité cardiaque d'un sujet pendant 24 ou 48 heures, lui permettant de continuer ses activités normalement, sans alitement ni hospitalisation. L'enregistrement se fait sur une bande magnétique ou un support numérique. Cet examen permet l'analyse du rythme cardiaque d'un individu de façon à y déceler d'éventuelles pathologies cardiovasculaires (Figure III-13).

Le Holter est particulièrement indiqué dans le cas de certaines pathologies cardiaques mais aussi en prévention, chez des individus porteurs d'un stimulateur cardiaque qu'il faut surveiller. Son utilisation est également recommandée pour la surveillance d'arythmies cardiaques (extrasystoles entre autres) ainsi que dans les cas où le diagnostic ne peut être posé avec certitude ; c'est le cas de certaines angines de poitrine par exemple, ou lorsque l'épreuve d'effort est impossible ou non significative, pour dépister une ischémie myocardique silencieuse (c'est-à-dire sans douleur thoracique) chez des sujets à facteurs de risque élevés (tabac, alcool, sucre, athérome, hypertension artérielle, etc)[Zidelmal, 2012].



Figure III-13 : le Holter

6.4 Ondes et intervalles de L'ECG

L'activité du cœur est enregistrée par des signaux électriques à la surface du corps à des points spécifiques, qui constitue l'électrocardiogramme (ECG). L'ECG est le signal biomédical le plus étudié pour caractériser les anomalies cardiaques. L'ECG peut être divisé approximativement en trois composantes principales **P**, **QRS** et **T** (figure III-14). Le rythme

cardiaque normal est composé par des ondes liées en général à des actions mécaniques du cœur.

Les différentes ondes et les intervalles les séparant ont permis de dresser le standard ECG. Toute interprétation de l'électrocardiogramme se réfère aux caractéristiques de forme et de largeur d'onde du signal ECG normal présentées sur la figure 14.

- **L'onde P** représente la dépolarisation auriculaire depuis le nœud sinusal vers le nœud atrio-ventriculaire. C'est l'onde qui précède le complexe **QRS**.

- **Le complexe QRS** correspond à la dépolarisation ventriculaire précédant l'effet mécanique de contraction, l'onde **R** est de grande amplitude car la masse des ventricules est supérieure à celle des oreillettes. Par définition, l'onde **Q** est la première onde négative, l'onde **R** la première onde positive du complexe, et l'onde **S** la première onde négative après l'onde **R**. Toute onde supplémentaire, positive ou négative, sera appelée **R'**, **S'**, **R''**, etc. La figure 8 donne les différents aspects que peut prendre le **QRS**.

- **L'onde T** représente la repolarisation des ventricules. Cette onde succède au complexe **QRS** après retour à la ligne isoélectrique, (sauf pathologie particulière).

- **Intervalle RR** : fréquence des battements cardiaques ; il correspond au délai entre deux dépolarisations des ventricules. C'est cet intervalle qui permet de calculer la fréquence cardiaque.

- **Intervalle PP** : période de polarisation des oreillettes et c'est le délai entre deux dépolarisations des oreillettes.

- **Segment PR** : pause du nœud AV (auriculo-ventriculaire). Le segment **PR** correspond au délai entre la fin de la dépolarisation des oreillettes et le début de celle des ventricules. C'est le temps pendant lequel l'onde de dépolarisation est bloquée au niveau du nœud AV.

- **Intervalle PR** : durée de conduction auriculo-ventriculaire ; cet intervalle correspond au délai entre le début de la dépolarisation des oreillettes et celle des ventricules. C'est le temps de propagation de l'onde de dépolarisation jusqu'aux cellules myocardiques ventriculaires.

- **Intervalle QT** : durée de systole ventriculaire, il représente le temps de systole ventriculaire, qui va du début de l'excitation des ventricules jusqu'à la fin de leur relaxation.

- **Segment ST** : durée de stimulation complète des ventricules, ce segment correspond à la phase pendant laquelle les cellules ventriculaires sont toutes dépolarisées, le segment est alors isoélectrique.

- Enfin, **l'onde U** non constante, traduit la repolarisation du réseau (des fibres) de Purkinje. C'est une onde positive qui suit l'onde T, visible essentiellement en précordiale, dont la signification est discutée (repolarisation prolongée des cellules du réseau de Purkinje ou à un facteur mécanique correspondant à la relaxation du myocarde).

En général, les anomalies cardiaques laissent leurs traces sur un ou plusieurs paramètres liés aux ondes de l'ECG.

L'enregistrement de l'évolution temporelle du champ électrique résultant de l'activité électrique du cœur est appelé l'électrocardiogramme de surface (ECG). L'ECG est effectué au moyen d'électrodes cutanées, sur un support papier standard. Les plans frontal et horizontal représentent respectivement les dérivations des membres et les dérivations précordiales.

En largeur 1 mm équivalent, à une durée de 40 ms et en hauteur 1 mm correspond à 0,1 mV mais d'habitude on exprime les voltages en mm [Marouf, 2010].

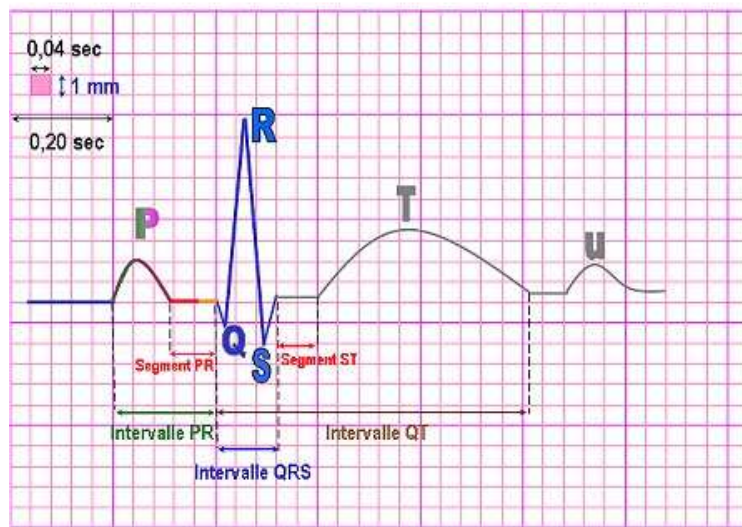


Figure III-14 : ECG normal.

7. Le rythme cardiaque sinusal

En absence de toute pathologie, le rythme est sinusal. Le rythme sinusal est le rythme cardiaque normal. Il correspond à une activation physiologique des oreillettes, puis des ventricules à partir du nœud sinusal. Son rythme est compris entre 60 et 80 battements par minute avec un intervalle régulier entre des battements normaux (voir figure III-I5). Le cœur s'accélère normalement lors d'une activité physique, dans des circonstances physiologiques qui exigent un surcroît de demande métabolique ou sous l'effet des émotions ou d'excitants tels que le café et le tabac.

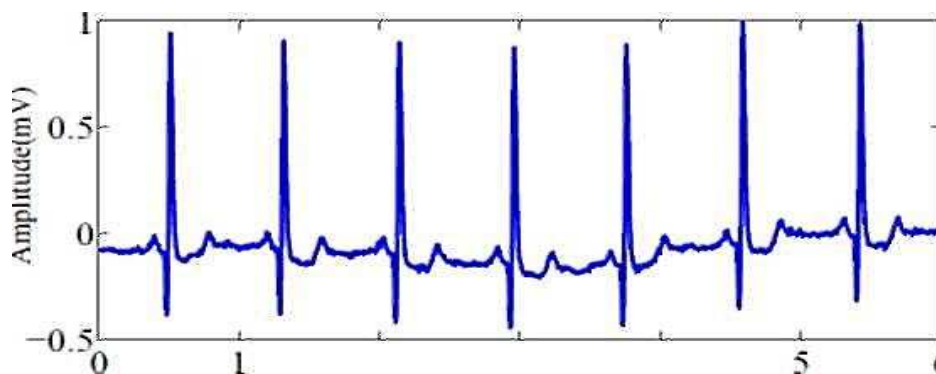


Figure III-15 : Rythme sinusal, segment de l'enregistrement AHA : 0201.

8. Troubles du rythme et de la conduction cardiaque

L'arythmie, qui peut être bénigne ou maligne, est un trouble du rythme cardiaque, c'est-à-dire une irrégularité des contractions du cœur qui affecte sa fréquence, l'intensité de ses contractions et sa régularité. On parle également d'arythmie quand le cœur bat à moins de 60 pulsations ou à plus de 100 pulsations par minute sans cause apparente. Une accélération (tachycardie), un ralentissement (bradycardie) ou un changement du rythme cardiaque sont les formes d'arythmie.

Les pathologies ou anomalies sont détectées et classées en fonction de leur déviation par rapport au rythme idéal qu'est le rythme sinusal. On peut les classer aussi selon leur fréquence et leur localisation. Une lecture complète sur les arythmies peut être faite dans [Beasley, 2003] et [Dominguez et Dubuc, 2005].

8.1 Extrasystole ventriculaire

L'extrasystole ventriculaire (ESV) est un battement anormal suffisamment répandu pour être présenté en introduction. Les ESV s'observent sur quasiment tous les enregistrements, principalement en période de récupération après un effort. Bien que leur présence n'indique aucune pathologie particulière, si de façon récurrente, leur nombre par minute est supérieur à 6, elles peuvent être un signe précurseur d'une tachycardie ventriculaire, qui peut constituer une pathologie majeure.

Contrairement aux battements normaux qui ont pour origine la dépolarisation des cellules sinusales, l'ESV naît de la dépolarisation spontanée d'un petit groupe de cellules ventriculaires, appelé alors foyer ectopique ventriculaire. L'impulsion électrique créée n'emprunte pas la voie normale de conduction (faisceau de His), et se propage donc plus lentement dans les ventricules. La contraction ventriculaire ainsi étalée dans le temps perd de son efficacité.

Le tracé d'un battement ESV est caractérisé par deux propriétés : l'onde R n'est pas précédée d'une onde P, puisqu'il n'y a pas eu d'activité auriculaire préalable, et la durée du complexe est supérieure à la durée d'un complexe QRS normal (figure III-16).

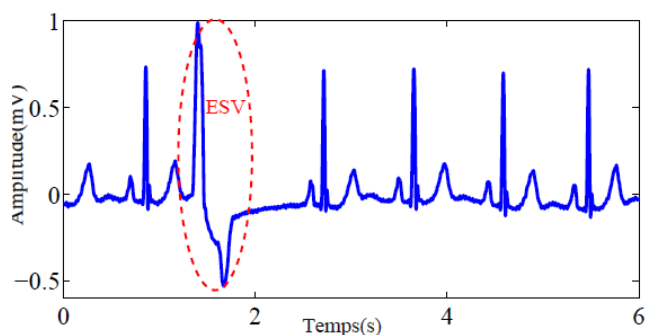


Figure III-16: Exemple d'extrasystole ventriculaire (MITDB : 119).

8.2 Arythmies à base de fréquence

8.2.1 Bradycardie

La bradycardie est caractérisée par une fréquence cardiaque inférieure à 60 bpm ; elle peut être d'origine sinusale, jonctionnelle, ou ventriculaire, selon le site d'initiation de l'impulsion électrique à l'origine des battements considérés.

a. Bradycardie sinusale

En plus de sa basse fréquence, la bradycardie sinusale est caractérisée par la présence systématique d'une onde **P** avant les complexes **QRS** (figure III-17) car l'origine de la contraction ventriculaire reste la dépolarisation du sinus et des oreillettes, comme lors de battements normaux. Les causes d'une telle arythmie sont multiples et souvent extérieures au système cardiovasculaire : traitement médicamenteux (bêtabloquant, digoxyne, ...), hypothermie, urémie, ...

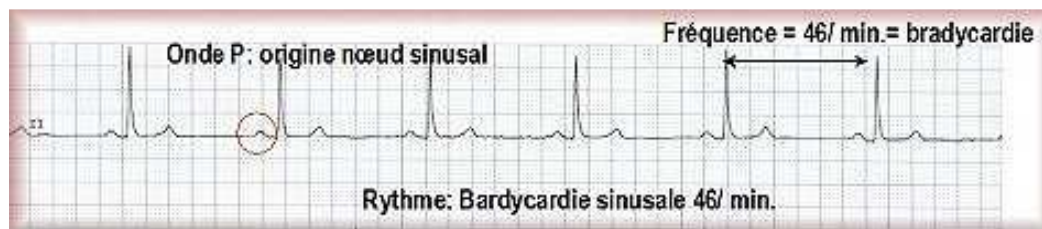


Figure III-17: Exemple de Bradycardie sinusale.

b. Bradycardie jonctionnelle

En cas de dysfonctionnement sinusal avec ou sans dépolarisation auriculaire, le nœud AV peut assurer le rôle de pacemaker de secours à une fréquence de l'ordre de 30 à 60 bpm.

L'impulsion électrique prend naissance dans le nœud auriculo-ventriculaire (AV) et suit le chemin de conduction habituel : la morphologie des ondes QRS est identique à celle du battement normal ; en revanche, l'onde P peut être absente ou, si elle est présente, elle peut être désynchronisée de la systole ventriculaire.

Ce rythme, appelé rythme d'échappement jonctionnel, dure jusqu'à ce qu'il soit inhibé par un rythme plus rapide (figure III-18).

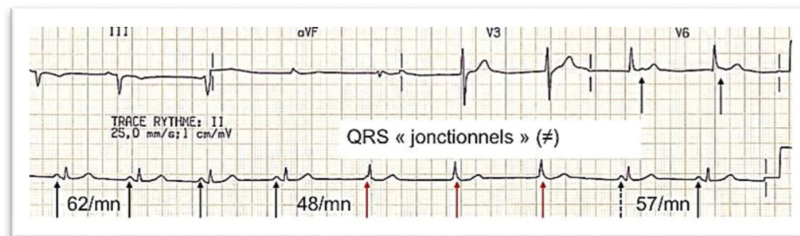


Figure III-18: Exemple de Bradycardie jonctionnelle.

Lorsque le problème de conduction intervient en aval du nœud AV, c'est un groupe de cellules du muscle ventriculaire qui est susceptible de devenir pacemaker ; le rythme devient alors une succession d'ESV à une fréquence très lente, entre 15 et 40 bpm, appelé rythme d'échappement ventriculaire. Ce type de pathologie constitue une indication typique de la pose d'un stimulateur cardiaque pour éviter une syncope possible.

8.2.2 Tachycardie

À l'inverse de la bradycardie, la tachycardie est caractérisée par le fait que la fréquence est supérieure à 100 bpm ; elle peut être d'origine sinusale, auriculaire ou ventriculaire.

a. Tachycardie sinusale

La tachycardie sinusale correspond à un rythme sinusal dont la fréquence est comprise entre 100 et 180 bpm ; une onde **P** est observée avant chaque complexe **QRS**, de forme identique à celle observée lors du rythme normal. Une tachycardie sinusale est considérée comme pathologique si elle est de longue durée et indépendante du contexte (figure III-19).

Les causes de cette pathologie sont habituellement extracardiaques, et incluent tous les facteurs de stimulations du système nerveux (surmenage,...) et l'influence de différentes substances comme l'adrénaline ou la caféine.

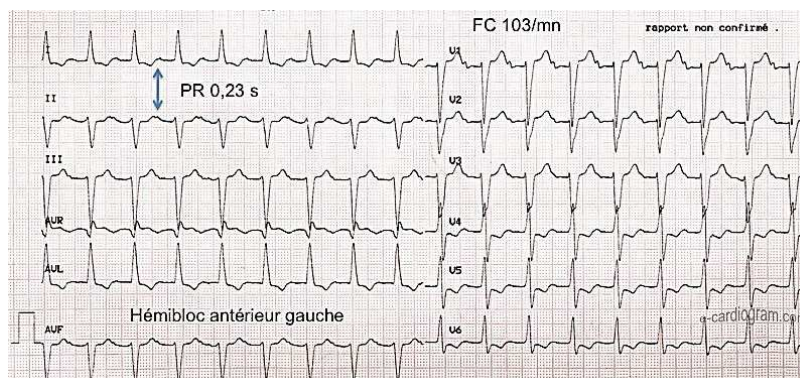


Figure III-19: Tachycardie sinusale.

b. Tachycardie auriculaire et nodale AV

La tachycardie auriculaire peut avoir pour origine un foyer ectopique, une boucle de stimulation (flutter) ou une voie qui court-circuite la voie AV, dite voie accessoire, avec réentrée par le nœud AV. Dans le cas d'un foyer ectopique, il s'agit d'un groupe de cellules situées dans les oreillettes, qui se dépolarisent spontanément et plus rapidement que le sinus, prenant ainsi sa place. La dépolarisation des oreillettes n'étant pas d'origine sinusale, la propagation de l'influx nerveux diffère de celle qui a pour origine le sinus, et l'on observe une onde P de forme inhabituelle.

La fréquence typique de décharge de ce type de foyer est comprise entre 120 et 200 bpm. La décharge régulière d'un foyer ectopique localisé dans le nœud AV peut aussi être la cause d'une tachycardie, dite tachycardie nodale AV ; dans ce cas, la fréquence des battements peut atteindre 250 bpm. Contrairement à la tachycardie auriculaire, aucune onde P ne précède les complexes QRS, car il n'y a pas d'activité auriculaire avant le battement (il peut y avoir une activité auriculaire rétrograde dont la trace sur l'ECG est noyée dans le complexes QRS).

Le principal risque de ce type de pathologie est le manque d'efficacité des ventricules qui, contraints de se contracter très fréquemment, n'ont pas le temps de se remplir correctement de sang : l'alimentation du corps en oxygène peut en être altérée.

c. Tachycardie ventriculaire (TV)

La tachycardie ventriculaire a pour origine un ou plusieurs foyer(s) ectopique(s) ventriculaire(s) (qui se dépolarisent à tour de rôle). Les battements ont donc la forme d'extrasystoles ventriculaires très rapprochées (figure III-20). Ce type de rythme est dangereux à cause de son évolution en fibrillation ventriculaire qui conduit au décès du patient si elle n'est pas traitée à l'aide d'un défibrillateur dans les quelques minutes qui suivent son apparition.

Les bradycardies et tachycardies engendrent un rythme régulier pendant leur manifestation. Cependant, certains de ces événements peuvent, d'une part, se manifester de manière sporadique par phases, et produire ainsi des rythmes différents d'assez longues durées, ou d'autre part, créer de véritables troubles de la régularité.

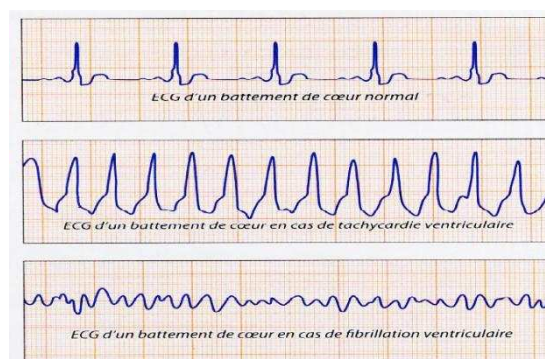


Figure III-20: Comparaison entre un ECG normal, ECG d'une Tachycardie Ventriculaire et ECG d'une fibrillation ventriculaire.

8.3 Les arythmies à base de régularité

L'absence de régularité des battements cardiaques est une caractéristique du rythme, importante pour le diagnostic ; elle est souvent associée à un trouble de la production ou de la conduction de l'impulsion électrique (foyers ectopiques, blocs, boucles, ...). Les arythmies permanentes ou sporadiques nécessitent un suivi médical et sont des indications typiques à la pose régulière d'un enregistreur ECG (ou Holter).

8.3.1 Foyer ectopique auriculaire ou nodal AV

Les irrégularités de rythme peuvent traduire la présence d'un ou plusieurs foyers ectopiques auriculaires (figure III-21). La fréquence d'expression de ces foyers, leur alternance entre eux et avec le sinus, et la transmission aux ventricules, peuvent entraîner diverses conséquences rythmiques : de l'extrasystole auriculaire isolée (ESSV) avec un repos compensatoire, qui introduit une distorsion rythmique locale, jusqu'à la fibrillation auriculaire évoquée plus loin, en passant par une tachycardie régulière ou une tachyarythmie (bigéminisme).

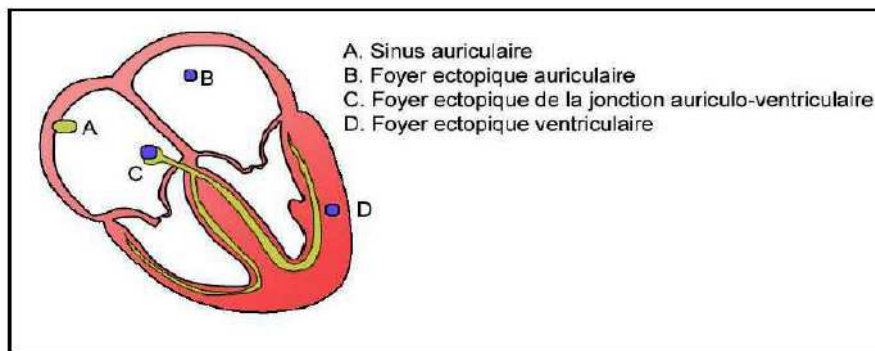


Figure III-21 : emplacement des foyers ectopiques.[zidelmal, 2012]

8.3.2 Foyer ectopique ventriculaire

Lorsque le foyer ectopique se trouve dans les ventricules, on obtient un battement ESV. Les ESV étant fréquemment suivies d'un repos compensatoire, sorte de temps de récupération, elles introduisent en général une distorsion locale du rythme, qui peut réapparaître plus ou moins fréquemment ou régulièrement selon la fréquence d'expression du ou des foyers ectopiques, et leur caractère aléatoire ou non (figure III-22).

Bigéminismes et Trigéminismes

Les bigéminismes et trigéminismes sont des rythmes à deux commandes. La commande de base (généralement sinusale) est interrompue par des battements d'origine ectopique. Lorsque l'on se trouve en présence d'un bigéminisme, les QRS qui appartiennent au rythme de base sont suivis d'un QRS d'origine ectopique avec une succession de 1/1. On parle de trigéminisme lorsqu'on est en présence d'une succession 2/1. Les figures (III-22 et III-23)

présentent respectivement un exemple de rythme de bigéminisme et de trigéminisme dans le cas d'un rythme sinusal avec un foyer ectopique ventriculaire.

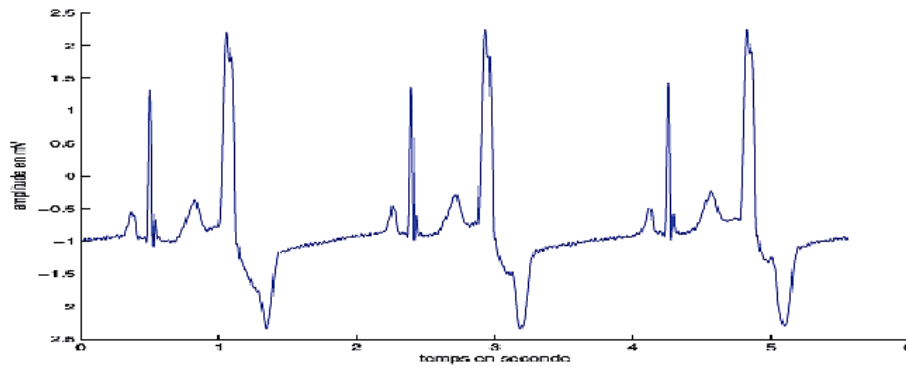


Figure III-22 : Exemple de rythme Bigéminisme. [zidelmal, 2012]

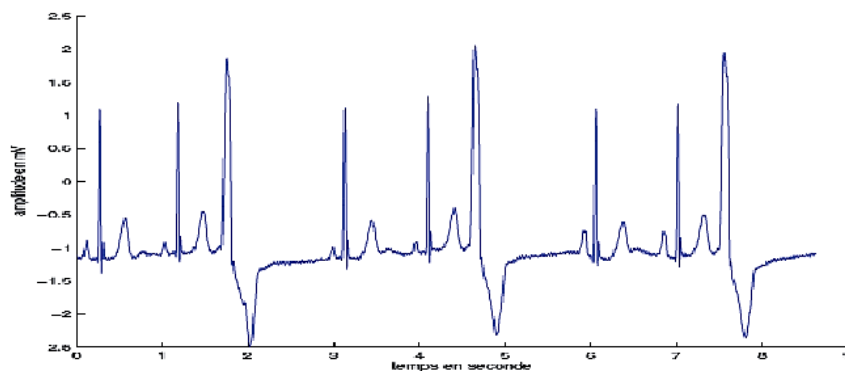


Figure III-23 : Exemple de rythme Trigéminisme. [zidelmal, 2012]

8.3.3 Fibrillations et Flutter

a. Flutter auriculaire

Le flutter est parmi les causes de troubles du rythme, dans ce cas, la fréquence de l'onde P peut atteindre 300 bpm, voire davantage. À cette fréquence, le nœud auriculo-ventriculaire ne parvient pas à conduire toutes les impulsions électriques vers les ventricules, et l'on observe généralement des blocs 2:1, au cours desquels seule une contraction auriculaire sur deux est suivie d'une systole ventriculaire ; la fréquence ventriculaire est donc d'environ 150 bpm dans un tel cas.

Plus rarement, des flutters 3:1, 4:1 sont observés. Dans ce type de pathologie, il existe un risque majeur pour la santé du patient, en raison de la possibilité de génération d'embolies, dus aux turbulences dans le flux sanguin au niveau des oreillettes.

b. Fibrillation auriculaire (FA)

La fibrillation auriculaire est plus fréquente que le flutter, elle touche 5 à 10% des personnes de plus de 65 ans. Dans cette pathologie, l'activité auriculaire n'est plus composée de

battements ; soumis à plusieurs foyers ectopiques et à des boucles de conduction locales, le fonctionnement du myocarde auriculaire est totalement désorganisé. Ces mouvements anarchiques peuvent néanmoins transmettre quelques impulsions au nœud auriculo-ventriculaire, qui les transmet à son tour aux ventricules et entraîne leurs contractions.

Comme cette transmission revêt un caractère aléatoire, le rythme des complexes QRS est complètement irrégulier. L'absence des systoles auriculaires n'est pas grave en soi, mais elle est cependant responsable d'une baisse significative de l'efficacité cardiaque à deux niveaux :

- D'une part, parce que le cœur ne bénéficie pas de la systole auriculaire qui assure une partie du remplissage sanguin des ventricules,
- Et d'autre part, parce que le rythme moyen est généralement supérieur à 100 bpm et peut atteindre 200 bpm.

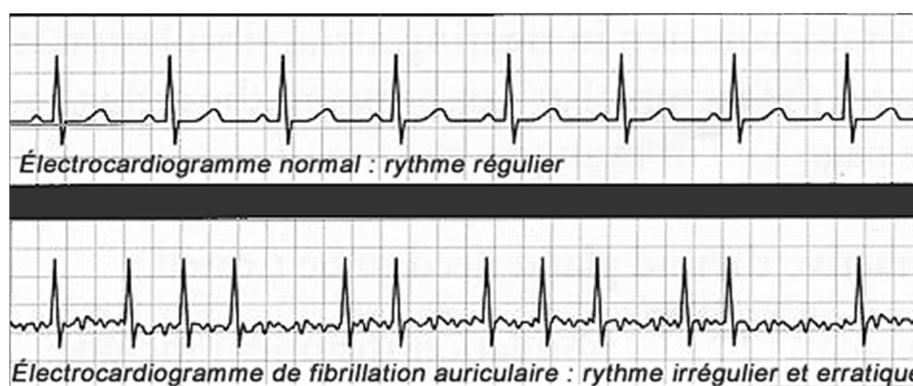


Figure III-24 : Comparaison entre un rythme régulier et une fibrillation auriculaire.

Le risque majeur lié à cette pathologie est celui du flutter, c'est-à-dire de l'envoi possible, dans la circulation, d'embolies formées au niveau des oreillettes. Ce risque est relativement faible lorsqu'une FA est permanente, tandis qu'il est augmenté lors d'épisodes de FA sur fond sinusal (FA paroxystique), surtout lors du passage d'un rythme à l'autre.

c. Fibrillation ventriculaire (FV)

Elle représente l'équivalent physiologique de la fibrillation auriculaire, mais transposée aux ventricules : les ventricules se déchargent alors de manière totalement désynchronisée, il n'y a plus de systole cardiaque. La fibrillation ventriculaire constitue une arythmie particulièrement grave, car elle est une menace de mort imminente : en effet le cœur n'assure plus du tout son travail de pompe, le sang ne circule plus, ce qui conduit à une asphyxie de tous les tissus du corps, dont le myocarde lui-même. Sans une intervention (défibrillation) immédiate, susceptible de resynchroniser la dépolarisation des cellules du myocarde et faire ainsi « repartir » le mouvement cardiaque, la mort s'ensuit. Les personnes qui présentent de tels risques peuvent aujourd'hui bénéficier de l'implantation d'un défibrillateur : placé au niveau du thorax (pacemaker), il est muni d'une sonde qui peut détecter l'anomalie rythmique et conduire l'appareil à délivrer une forte décharge électrique.

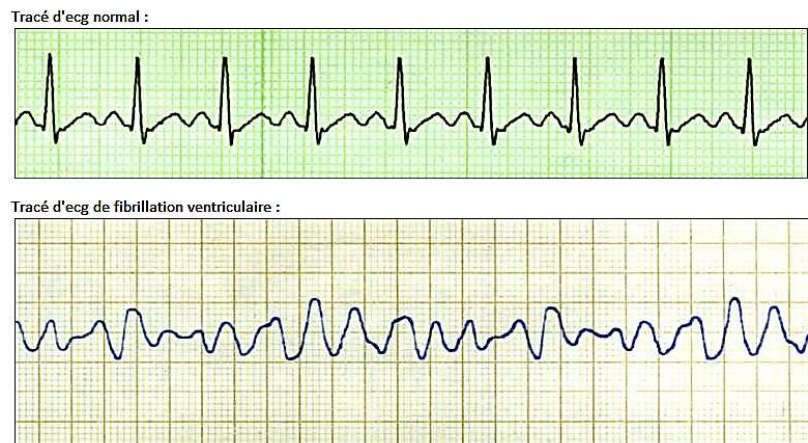


Figure III-24 : Comparaison entre un rythme régulier et une fibrillation ventriculaire.

8.4 Les blocs cardiaques

Les blocs cardiaques sont dus à une rupture de conduction dans le myocarde, ce qui altère sa dépolarisation. Ces ruptures peuvent être plus ou moins sévères : freinantes (allongement du temps de parcours), intermittentes (un stimulus sur 2 ou 3 est conduit), ou complètes (aucune conduction). On distingue quatre types de blocs cardiaques :

a. Bloc Sino-Auriculaire (BSA) : Le nœud sinusal peut ne pas transmettre de stimulus aux cellules des oreillettes. La conséquence est qu'au moins un cycle complet n'est pas effectué. Après la pause, due au bloc, le cycle reprend normalement si aucun autre foyer ectopique n'a déclenché de contraction.

b. Bloc Auriculo-Ventriculaire (BAV) : On appelle BAV l'altération de la conduction du stimulus de dépolarisation entre les oreillettes et les ventricules. On distingue trois degrés de sévérité.

- Les BAV de premier degré provoquent l'allongement du segment PR de chaque cycle ;
- Les BAV de deuxième degré traduisent l'absence momentanée d'onde QRS après une onde P normale. Lorsque les segments PR précédents sont normaux, on parle de Mobitz de type II. Lorsque les segments précédents sont rallonges, on parle de Mobitz de type I ;
- Les BAV de troisième degré sont dit complets, c'est-à-dire qu'aucune dépolarisation auriculaire ne parvient aux ventricules. Un foyer ectopique ventriculaire ou jonctionnel joue alors le rôle de pacemaker. Le foyer est identifiable par la forme et la fréquence des battements. Les activités auriculaire et ventriculaire sont complètement dissociées.

c. Bloc de branches : Un bloc de branche est dû au blocage de la dépolarisation dans l'une des branches du faisceau de His. Un bloc dans l'une ou l'autre branche provoque un retard dans la dépolarisation des ventricules, il est désynchronisé et le complexe QRS se voit élargi. Les

figures (III-25 et III-26) indiquent respectivement un bloc de branche gauche ou *Left Bundle Branch Block* (LBBB) et un bloc de branche droite ou *Right Bundle Branch Block* (RBBB).

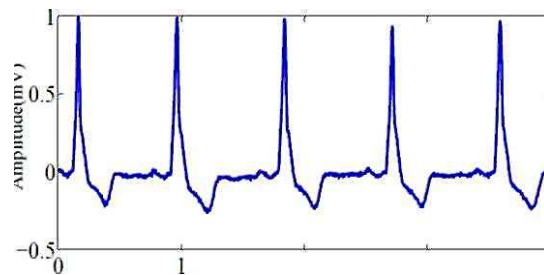


Figure III-25 : Exemple de Bloc de Branche Gauche (MITDB : 111).

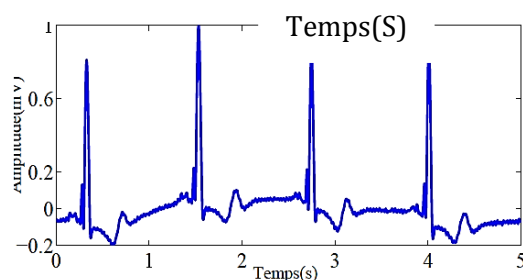


Figure III-26: Exemple de Bloc de Branche Droite (MITDB : 124).

9. Parasites et Artefacts visibles sur l'électrocardiogramme

La valeur diagnostique de l'électrocardiogramme dépend en grande partie de la qualité de l'interprétation et celle-ci est influencée par la qualité technique de l'enregistrement. Pour une interprétation correcte, visant à fournir un maximum d'informations, l'essentiel est d'obtenir un *ECG* dénué de parasites, d'artefacts et d'erreurs de manipulation. D'ailleurs ces perturbations ont fait objet de plusieurs études [Moody et al, 1984], [Mark, 1988], [Borries et al., 2005], [Chen et al., 2006].

La plupart des problèmes d'enregistrement qui étaient fréquents avec les anciens appareils ont aujourd'hui pratiquement disparu avec les appareils modernes, surtout depuis l'introduction des systèmes totalement informatisés. Cependant, certaines perturbations restent encore difficiles à traiter de manière automatique.

Ces bruits peuvent avoir plusieurs origines : technique, physique, pathologique, ou pharmacologique. Nous allons développer uniquement les aspects techniques et physiques.

9.1 BRUITS TECHNIQUES

Le matériel utilisé lors de l'enregistrement doit être manipulé avec précaution car il peut être source de bruits dont les plus courants sont présentés ci-dessous :

a. Bruits dus au secteur : Le réseau de distribution électrique peut parfois brouiller le signal électrocardiographique avec une onde dont l'harmonique fondamentale est à 50 Hz (voir Figure III-27). Ce bruit apparaît sur tous les enregistrements. Il peut être assez fort mais s'élimine facilement avec un filtre sélectif car c'est un bruit de haute fréquence à bande étroite.

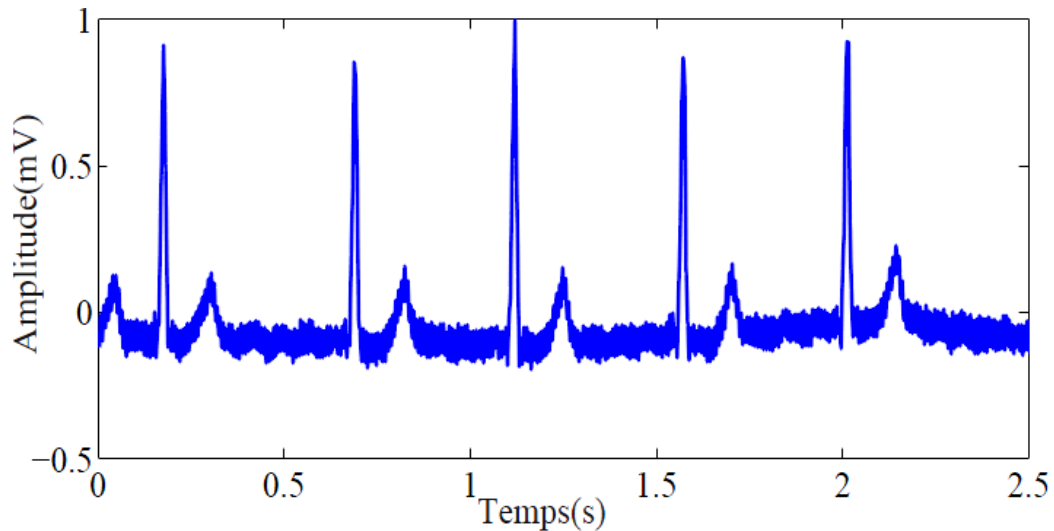


Figure III-27 : Signal électrocardiographique perturbé par le secteur. (Segment de l'enregistrement AHA : 0201).

b. Bruits dus au mauvais contact électrode-peau : Lorsque les électrodes sont incorrectement connectées, des sauts brusques de la ligne de base apparaissent (voir Figure III-28). L'effet sur le tracé peut aller de la simple diminution d'amplitude à l'apparition de pics lorsque les électrodes sont en contact intermittent avec la peau. Ces pics peuvent parfois être confondus avec les ondes du tracé normal. Ce type de bruit s'élimine difficilement car sa puissance se trouve dans la même gamme de fréquence que celle des complexes QRS.

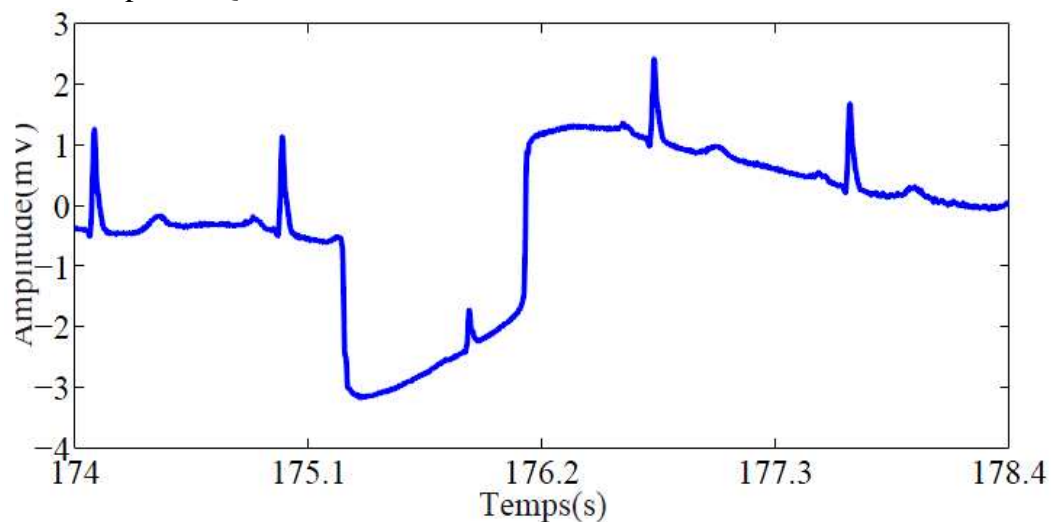


Figure III-28 : Bruit dû au mauvais contact électrode-peau (MITDB : 101).

C. AUTRES BRUITS COURANTS :

Parmi les bruits courants, on peut citer les artefacts dus aux mouvements des câbles électriques, à la saturation des instruments de mesure, aux mauvais câblages, au port de vêtements synthétiques, etc.

9.2 Bruits physiques

Les artefacts physiques sont dus aux activités électriques du corps humain telles que les contractions des muscles ou encore la respiration.

a. **Dérives de la ligne de base** : Lors de l'enregistrement de l'ECG, l'activité respiratoire peut faire osciller la ligne de base du signal à un rythme régulier. D'autres perturbations peuvent avoir pour effet de déplacer temporairement la ligne de base comme par exemple, les mouvements du patient. Un tel bruit est visible sur la figure (III-29). Ces perturbations sont généralement peu gênantes pour l'analyse de l'ECG et peuvent être en grande partie filtrées car leur puissance se situe dans une bande de fréquence basse et empiète peu sur celle de l'ECG.

b. **Bruits myoélectriques ou tremblements somatiques** : La contraction d'un muscle est commandée par une dépolarisation des cellules musculaires. Bien que les électrocardiographes soient construits pour être surtout sensibles aux fréquences du myocarde, l'ECG enregistre aussi les contractions des muscles squelettiques. L'aspect le plus courant de ce bruit est une oscillation de haute fréquence et non stationnaire. Elle est liée à un tremblement musculaire d'un sujet qui n'est pas convenablement détendu. Un exemple de tel bruit est représenté sur la figure (III-30).

Ces perturbations sont assez gênantes lorsque le patient bouge beaucoup ou lorsqu'il frissonne, elles peuvent noyer les ondes P et T et empêcher parfois la détection des pics R. L'apparition de ces perturbations dépend de l'état du patient, s'il est très tendu ou atteint de maladies (Parkinson par exemple), l'enregistrement peut être de mauvaise qualité sur toutes les voies de l'ECG.

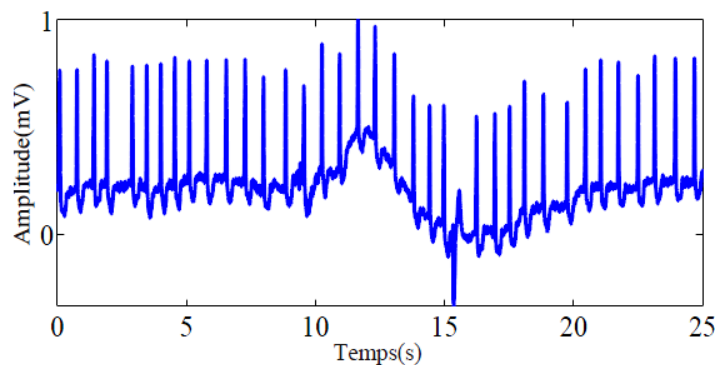


Figure III-29 : Ondulation de la ligne de base (MITDB : 119).

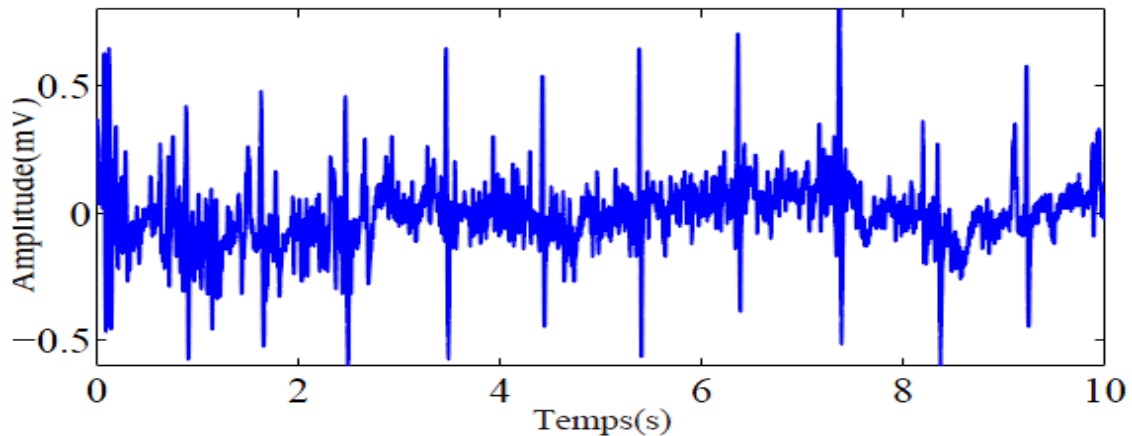


Figure III-30 : Exemple de bruits musculaires sur un segment de l'enregistrement MITDB : 119.

c. Autres artefacts altérant l'ECG : Certaines maladies généralisées peuvent affecter le tracé électrocardiographique. L'hyperthyroïdie, l'ischémie, l'hypokaliémie (prolongement de l'intervalle QT, onde T aplatie), modifient l'électrocardiogramme.

L'usage de médicaments, notamment la digoxine qui bloque la conduction AV et ralentit la fréquence cardiaque. La digitaline provoque un abaissement du segment ST avec inversion des ondes T et tend à raccourcir l'intervalle QT.

10. Reconnaissance automatique des arythmies : Etat de l'art

Depuis quelques décennies, la reconnaissance automatique d'arythmies cardiaques est devenue un domaine de recherche très actif. Il représente un carrefour entre plusieurs disciplines comme la médecine et la reconnaissance de formes. La classification automatique d'arythmies cardiaques passe par deux étapes :

Description et quantification des motifs à reconnaître.

Apprentissage et décision.

10.1 Paramètres descriptifs d'un battement cardiaque

Le calcul de ces paramètres se fait sur le signal ECG dans le domaine temporel, dans le domaine fréquentiel ou dans le domaine temps-fréquence selon l'information recherchée. Les performances d'un descripteur de battements cardiaques dépendent de plusieurs facteurs comme la qualité de l'enregistrement, les prétraitements, la base d'étude ainsi que le nombre de dérivations étudiées.

Avant de procéder à la quantification des battements cardiaques, leur localisation est nécessaire. La détection des complexes QRS constitue le préalable de toute analyse automatique de l'ECG. Durant ces 30 dernières années, plusieurs chercheurs se sont penchés sur ce problème en posant divers algorithmes basés pour la plupart sur la dérivée de l'onde R ou sur le contenu fréquentiel des complexes QRS. Les différentes méthodes ont été récapitulées et regroupées par type dans [Kohler et al., 2002].

La qualité d'un classifieur automatique de battements cardiaques est étroitement liée au choix des paramètres descriptifs. Ces derniers doivent être les plus discriminants possibles sans s'éloigner de ceux habituellement utilisés par les médecins. Chaque battement cardiaque est principalement décrit par sa morphologie. Certains auteurs l'ont simplement représentée en sous-échantillonnant le segment QRS [De-Chazal et al., 2004 ; De-Chazal et Reilly, 2006], d'autres ont utilisé des outils de compression en appliquant les fonctions d'Hermite [Lagerholm et al., 2000], ou la Transformée de Karhunen Loève. Cette idée fût proposée par [Moody et Mark, 1989] puis reprise par [Gomez, 2006]. Ces méthodes de compression permettent de réduire relativement le nombre d'échantillons en sauvegardant les caractéristiques temporelles et fréquentielles du battement ainsi que sa morphologie, pour distinguer un battement normal d'un battement anormal, V. Krasteva et al Ont procédé au calcul des coefficients d'inter-corrélation entre les battements de la base d'étude et un battement de référence convenablement choisi dans chaque classe considérée [Krasteva et Jecova, 2007].

Comme il a été exposé précédemment, les battements anormaux proviennent d'un foyer ectopique ou d'un pacemaker. Ils peuvent aussi être dus à la rencontre d'un bloc de branche droit (RBBB) ou gauche (LBBB). Ces battements sont connus pour leur largeur et leur lenteur. A partir de là. Certains auteurs ont pensé à analyser le contenu fréquentiels des battements par la Transformée de Fourier [Minami et al., 1999] ou par bancs de filtres [Afonso et Tompkins, 1999]. La transformée en ondelettes, offrant les possibilités d'analyser séparément les composantes fréquentielles, s'avère de plus en plus appropriée pour le traitement des signaux biologiques et notamment pour cette application où les descripteurs temps-fréquences s'avèrent efficaces [Christov et al., 2006].

L'étude de la régularité des battements cardiaques est incontournable lors de rétablissement d'un diagnostic. L'intervalle RR est un paramètre classiquement utilisé par les cardiologues et par tous les chercheurs qui travaillent sur la reconnaissance automatique d'arythmies [De-Chazal et al., 2004 ; De-Chazal et Reilly, 2006], [Krasteva et Jecova, 2007].

10.2 Algorithmes d'apprentissage et de décision

Le score d'un classifieur dépend étroitement du choix et du calcul des paramètres caractéristiques de la forme à reconnaître. Cependant, le choix de la méthode de classification n'influence pas moins le résultat et le taux de bonne classification. La littérature propose une variété de règle d'apprentissage : Les réseaux de neurones ont été appliqués avec succès pour la discrimination de battements cardiaques [Yeap et al., 1990],[Hu et al., 1997],[De-Chazal et al., 2004], [De-Chazal et Reilly, 2006],[Marouf. 2010], [Patra 2010], [Amar. 2011]. D'autres travaux ont utilisé les K plus proches voisins [Christov et al. 2005]. A côté de ces méthodes d'apprentissage, les Supports Vector Machines (SVMs), connus pour leur excellent pouvoir de généralisation en classification et en régression, ont conduit à des résultats prometteurs dans l'aide au diagnostic médical [Osowski et al., 1997], [Zidelmal et al., 2009].

10.3 Problèmes liés à la reconnaissance automatique des arythmies

D'après la littérature, plusieurs problèmes sont souvent rencontrés lors de la classification des désordres cardiaques. Parmi ces problèmes, on cite :

- Le signal ECG présente une variabilité élevée inter et intra-patient, dans la morphologie et la synchronisation. Par conséquent, la réalisation d'un classificateur globale est un problème très difficile.
- Les paramètres extraits à partir du signal d'ECG sont très susceptibles aux variations de la morphologie de l'ECG, qui est non stationnaire, ainsi qu'aux artefacts qui l'affectent.
- Les processus d'extraction de paramètres sont confrontés à la question du choix des paramètres les plus appropriés (les plus discriminants) tout en réduisant les vecteurs de caractéristiques de façon optimale.

11. Conclusion

A travers ce chapitre, nous avons introduit les éléments de base de l'électrocardiographie du cœur, de son fonctionnement et des différents aspects de l'ECG liés à ses enregistrements ou à la présence d'arythmies. Ces éléments sont nécessaires pour la compréhension des prochains chapitres.

Après une introduction sur l'anatomie du cœur humain, nous avons décrit brièvement ses deux activités électrique et mécanique ainsi que le lien entre elles.

Dans la deuxième partie de ce chapitre, nous avons exposé les notions d'électrocardiographie. Quelques pathologies cardiaques typiques ont été présentées et illustrées, chacune par un enregistrement ECG correspondant. Les différents types d'artefacts qui peuvent affecter le signal ECG enregistré et contraindre son traitement sont également exposés. Nous avons terminé ce chapitre en exposant quelques algorithmes de reconnaissance automatique d'arythmies cardiaques ainsi que les contraintes souvent rencontrées.

Chapitre IV :

Résultats et discussion

Expérimentation, Résultats et discussion

1. Introduction

Le but principal de notre étude est de mettre en œuvre un classifieur évolutif flou qui a été proposé récemment par **Xydeas et al. (2005)** dans le cadre de la classification des signaux électroencéphalogrammes en deux classes « douleur et non douleur » et de l'appliquer, afin de classer d'une manière automatique et en temps réel (en ligne) les signaux ECG, sachant que c'est la première fois que ce classifieur est utilisé pour la reconnaissance d'arythmies cardiaques. Les résultats obtenus sont analysés afin de tirer des conclusions qui nous seront utiles pour proposer une amélioration pour ce classifieur dans de futurs travaux.

L'algorithme d'apprentissage appliqué se caractérise par :

- Une base de règles initialement vide : en effet ce classifieur a le pouvoir de commencer l'apprentissage à partir de zéro et sans connaissances préalables. La première donnée arrivée va constituer la première règle (la première donnée sera considérée comme le prototype autour duquel sera formée la première classe).
- Une structure ouverte et flexible : au fur et à mesure que les nouvelles données arrivent ; le classifieur va apprendre et faire évoluer sa base de règles floues. Par le calcul du potentiel des données, on aura des prototypes avec haut pouvoir descriptif autour desquels les clusters se forment. L'algorithme permet aussi la modification et la suppression d'anciennes règles devenus ambigus.

Dans le cadre de notre travail on cherche à reconnaître les pathologies cardiaques disponibles sur la base de données de l'Université de Californie (UC Irvine machine Learning repository) <http://archive.ics.uci.edu/ml/datasets/Arrhythmia>. Dans un premier temps nous allons considérer deux classes uniquement (présence et non présence d'arythmies).

2. Description de la base de données Arrythmia

Nous avons utilisé la base de données d'arythmies cardiaques de l'UCI Machine Learning [UCI Repository, 1998]. Cet ensemble de données contient 452 échantillons provenant de cas de 16 classes. La première classe est "normale", et les 15 autres classes sont de 15 types d'arythmies. Ces 15 classes sont fusionnées en une seule classe appelée classe "anormale". La classe16 se réfère à l'ensemble de celles non classées. Pour chaque échantillon, il y a 279 attributs, où les quatre premiers : âge, sexe, taille et poids, sont la description générale du patient, et les 276 autres attributs sont extraits des 12 enregistrements ECG standards. Pour les détails de l'ensemble de données, se référer à [Guvenir et al., 1997].

H. Altay Guvenir a mené cette étude afin de faire la distinction entre la présence et l'absence d'arythmies cardiaques et de les classer dans l'un des 16 groupes. Pour le moment, il existe des programmes qui font une telle classification. Cependant, il existe des différences entre la classification des cardiologues et celle des programmes. Nous nous efforçons de minimiser cette différence à l'aide d'outils d'apprentissage de la machine.

NB : Les noms et les numéros d'identification des patients ont été récemment retirés de la base de données.

Table IV-1 : Caractéristiques de la base de données UCI Arrythmia

Caractéristiques de la BDD:	Multi-variée	Nombre d'Instances:	452	région:	vie
Caractéristiques des Attributs:	Catégorique, entier, réel	Nombre d'attributs:	279	Date de don :	1998-01-01
Tâches associées:	Classification	Valeurs manquantes :	oui	Nombre d'occurrences (citations) sur le WEB:	68993

3. Traitements effectués sur la base de données :

Une des caractéristiques de cette base de données est son nombre élevé d'attributs qui pourrait nécessiter une stratégie de réduction. Plusieurs méthodes de fonctions réductrices ont été proposées pour cet ensemble de données. [Anugerah et al., 2013] présentent une étude comparative des différentes sélections d'attributs comme l'une des méthodes de prétraitement utilisés dans les applications d'apprentissage machine. Basé sur les méthodes de recherches et d'évaluateurs d'attributs, neuf combinaisons de sélection d'attribut sont testées et comparées. Ces données de résultats de réduction d'attributs sont ensuite classées selon huit classificateurs. Les meilleurs résultats globaux sont atteints par la combinaison de l'évaluateur SCF (Correlation-based Feature Selector) avec la méthode de recherche « Best- first » [Hall, 1999].

3.1 L'évaluateur d'attributs CFS (Correlation-based Feature Selector):

SCF est utilisé dans diverses applications du diagnostic médical à des fins de sélection de caractéristiques. C'est un algorithme de filtrage simple qui classe les sous-ensembles d'attributs selon une fonction d'évaluation heuristique basée sur la corrélation. Le biais de la fonction d'évaluation tend vers des sous-ensembles qui contiennent des attributs qui sont fortement corrélés avec la classe et non corrélés entre eux. Les scores pour chaque sous-ensemble d'attributs sont calculés selon l'équation (53).

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k+(k(k-1))\overline{r_{ff}}}} \quad (53)$$

Où M_S est le «mérite» d'une fonction heuristique du sous-ensemble S contenant les caractéristiques de k , $\overline{r_{cf}}$ est la moyenne de corrélation attribut-classe ($f \in S$) et $\overline{r_{ff}}$ est la moyenne attribut-attribut inter-corrélation. L'équation 7 forme le noyau de la SFC et impose un classement sur les sous-ensembles d'attributs dans l'espace de recherche de tous les sous-ensembles d'attributs possibles.

3.2 La recherche Heuristique « Best-First » :

La recherche « Best- First » est une stratégie de recherche d'intelligence artificielle qui permet de revenir en arrière sur le chemin de recherche. « Best- First » se déplace à travers

l'espace de recherche en apportant des modifications locales au sous-ensemble d'attributs courant. Si le chemin à l'étude paraît moins prometteur, la recherche « Best- First » peut revenir en arrière à un sous-ensemble précédent plus prometteur et continue la recherche à partir de celui-ci. La recherche « Best- First » explorera l'ensemble de l'espace de recherche, il est courant d'utiliser un critère d'arrêt. Normalement il s'agit de limiter le nombre de sous-ensembles complètement développées qui ne se traduisent par aucune amélioration.

Dans notre étude, après avoir utilisé l'algorithme de SCF avec la stratégie de recherche « Best- First », pour réduire la dimension du jeu de données d'arythmies, nous avons obtenu 40 attributs. Parmi ces attributs, nous remarquons que les caractéristiques les plus importantes retenues ont une signification médicale.

L'application de l'évaluateur d'attributs SCF combiné à la méthode de recherche « Best-First » est faite en utilisant « **WEKA 3.6** », voir (figure IV-2).

3.2.1 WEKA 3.6 : le logiciel d'extraction de données

Weka est une collection d'algorithmes d'apprentissage machine pour les tâches d'exploration de données. Les algorithmes peuvent être, soit appliqués directement à un ensemble de données, ou appelés à partir d'un code Java. Weka contient des outils pour les prétraitements de données, la classification, la régression, le regroupement, les règles d'association, et la visualisation. Il est également bien adapté pour le développement de nouveaux programmes d'apprentissage machine.

Weka est un logiciel open source publié sous la licence GNU General Public License <http://www.cs.waikato.ac.nz/ml/weka/>. (Figure IV-1).



Figure IV-1 : interface du logiciel Weka 3.6.11.

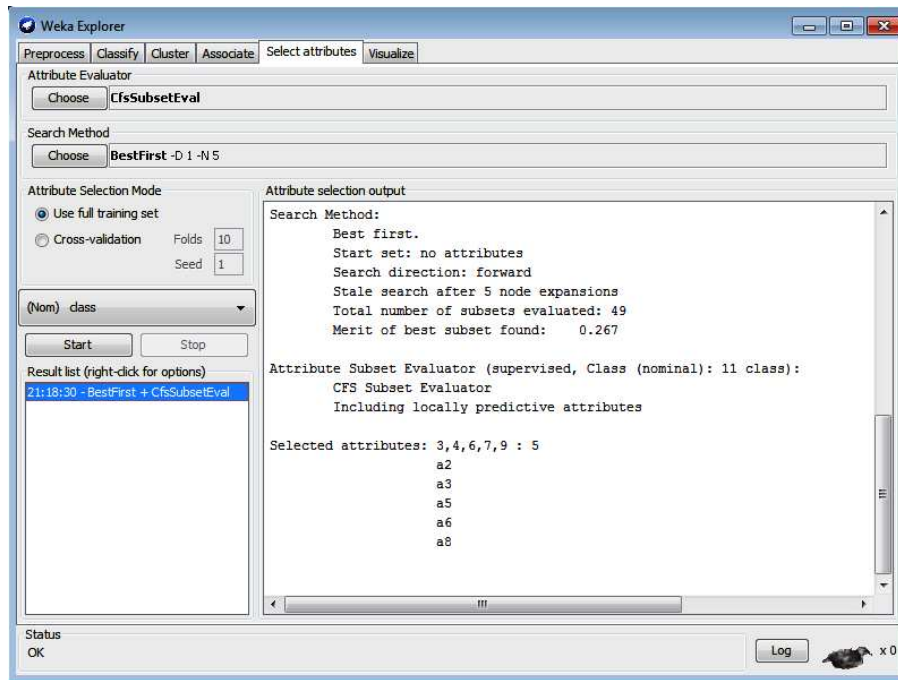


Figure IV-2 : interface de Weka 3.6.11 pour la sélection d'attributs.

4. Choix de L'algorithme :

Au cours de notre étude de l'art, nous avons trouvé plusieurs algorithmes différents que nous pouvons scinder en deux conceptions différentes.

En effet, il y a des algorithmes qui préconisent le calcul du potentiel d'une donnée afin de savoir si c'est une donnée que l'on peut utiliser comme prototype et construire ainsi une règle autour de cette dernière ou modifier une règle déjà existante ; sinon (si le potentiel est faible) l'assigner à l'une des classes existantes. La deuxième philosophie consiste à effectuer d'abord un pré classement ; ensuite en calculant le potentiel (en prenant compte des échantillons déjà lus de la classe concernée), soit on confirme le classement, soit on met à jour la structure du classifieur, en ajoutant ou en modifiant une règle. C'est cette deuxième philosophie qui nous a semblé être la mieux appropriée à notre contexte applicatif et semble donner de meilleurs résultats.

L'algorithme eclass comprend deux types différents selon leur sorties. Eclass0 avec étiquette de classe en sortie et eclass1 qui est basé sur la régression sur les paramètres. Dans notre étude, nous nous intéressons à eclass0.

4.1 Algorithme eclass0:

L'algorithme eclass0 proposé par Angelov et Zhou est présenté ci-dessous [Angelov et Zhou, 2008].

Begin eclass0

Initialisez eclass0 par le premier échantillon de données

$z_1 = [x_1, l_1]$; $(p_1)_1 \leftarrow 1$ (ou par une classe initiale si elle existe).

Tant que le flux de données n'est pas fini, **faire** pour l'échantillon de données.

Lire le vecteur de paramètres de l'échantillon de données x_k ;

Calculer l'appartenance à chacun des ensembles flous par (4) ;

Calculer la force d'activation de la règle par (3) ;

Classer l'échantillon en utilisant l'étiquette de classe gagnante \hat{L}_k par (2) ;

Au prochain pas ($K \leftarrow K + 1$) ou toutes les fois après cela ;

Obtenir le vrai label L_k de l'échantillon ;

Calculer son potentiel $p_k(z_k)$ en utilisant (11) ;

Mettre à jour les potentiels des prototypes existants par (12) ;

Si (13) influant **alors**

Ajouter un nouveau cluster basé centre au nouveau point de donnée(14) ;

Initialiser son potentiel à 1 comme indiqué par (15) ;

Mettre à jour les étendues de fonctions d'appartenance de la règle dans la base de sous-règles de la classe correspondante par (17) ou (18) ;

Si (16) influant **alors**

Enlever les règles qu'il influe ;

Sinon

Ignorer (ne pas changer la structure du cluster) ;

Mettre à jour les étendues de fonctions d'appartenance de la règle dans la base de sous-règles de la classe correspondante par (17) ou (18) ;

Fin si

Fin (tan que...faire)

Fin (eclass0).

4.2 Description de l'algorithme :

Cet algorithme comprend deux phases : une phase de classification et une phase de mise à jour.

4.2.1 Phase de classification :

Dans cette première phase de classification, on cherche à prédire la classe de la donnée à classer. Pour cela, on commence par initialiser le classifieur par le premier échantillon de donnée lu, en acceptant ce dernier comme premier prototype autour duquel sera formé le premier cluster/règle.

Son potentiel est mis initialement à 1 ainsi que son support et sa dispersion. On calcule l'appartenance à chacun des ensembles flous par (4) ; pour cela il faut d'abord :

- Calculer le support à l'instant K pour pouvoir calculer la dispersion (le support est initialement mis à 1 quand un prototype est produit, ensuite il est augmenté de façon incrémental pour chaque échantillon, qui est plus proche de ce prototype qu'à tout autre).
- On calcule ensuite la dispersion des données par règle et par paramètre (5) pour pouvoir calculer la fonction d'appartenance.

Après avoir calculé l'appartenance, on calcule le degré d'activation de la règle par (3) qui représente une T-norme de ces dernières, c'est-à-dire qu'on prend la plus petite valeur.

Finalement, pour classer la donnée courante, on prend l'étiquette gagnante par (2) (en utilisant le principe du « winner takes all », en d'autres termes on prend l'étiquette de la classe qui a le plus grand degré d'activation.

4.2.2 Phase de mise à jour :

Pour cette phase de mise à jour on va chercher la vraie étiquette de classe pour l'échantillon en utilisant l'étiquette issue de la première phase. Pour cela :

-On calcule le potentiel de la donnée courante afin d'identifier les prototypes qui représentent le mieux la classe à laquelle ils appartiennent.

-Pour le calcul du potentiel, on peut soit le calculer en utilisant la distance euclidienne, soit par la distance de cosinus. Vu que la base que nous avons retenue contient beaucoup de valeurs nulles, nous avons choisi la deuxième option car le cosinus se comporte mieux avec

ces dernières (la donnée avec le plus haut potentiel est choisie pour être le point focal de l'antécédent d'une règle floue).

-Ensuite on calcule récursivement le potentiel des prototypes existants pour les comparer au potentiel de la donnée courante. Là nous avons trois cas possibles :

- Si (13) influant, c'est-à-dire que le potentiel de la donnée courante est supérieur au potentiel d'un prototype existant, alors on ajoute un cluster au nouveau point de donnée par (14), son potentiel est initialisé par (15). On met à jour les étendues des fonctions d'appartenance de la règle dans la base de sous règles de la classe correspondante par (17) ou (18) avec ρ le pas d'apprentissage, c'est une valeur comprise entre [0,1] **Angelov et Zhou** ont proposé la valeur 0.5 pour simplifier l'expression).
- Si (16) influant, c'est-à-dire que la donnée courante a un meilleur potentiel que le prototype et qu'elle est décrite par ce dernier, alors il faut remplacer ce prototype par cette même donnée courante.
- Le troisième cas, si la donnée courante a un potentiel inférieur à celui de tous les prototypes existants, alors ne pas changer la structure du classifieur. Mettre à jour les étendues des fonctions d'appartenance de la règle dans la base de sous règles de la classe correspondante par (17) ou (18) avec ρ le pas d'apprentissage.

Afin d'appliquer l'algorithme `eclass0` sur notre base de données, nous avons utilisé l'algorithme disponible dans la boîte à outils `IRootLab` développée par le laboratoire Bioinformatics dirigé par Plamen Angelov [**Trevisan et al., 2013**].

4.3 La boîte à outils `IRootLab` :

`IRootLab` est une boîte à outils `MATLAB` libre et open-source pour analyse des données bio-spectroscopiques vibratoires (VBS). Elle offre une bibliothèque de classes pour programmation orientée objet, interfaces graphiques (GUI) et la génération automatique de code `MATLAB`. La bibliothèque de classes contient un grand nombre de méthodes, concepts (différents classifieurs, sélecteurs d'attributs...) et visualisations pour l'analyse des données VBS (figure IV-3), dont certaines sont introduites dans la boîte à outils. Les interfaces graphiques fournissent une interface pour la bibliothèque de classe, y compris un module pour fusionner plusieurs fichiers en un ensemble de données spectrales. La génération automatique de code permet aux développeurs d'écrire rapidement des scripts d'analyse de données VBS, constituant une ressource unique parmi les outils pour VBS. La documentation comprend un

manuel, des tutoriels, référence Doxygen générée et une vitrine de démonstrations (Figure IV-4). IRootLab peut gérer certains des formats de fichiers les plus populaires utilisés dans la bio-spectroscopie vibratoire. Cet outil est sous Licence: GNU-LGPL. Site officiel : <http://irootlab.googlecode.com/>.

Le lien pour la documentation : <http://bioph.lancs.ac.uk/irootlabdoc>.

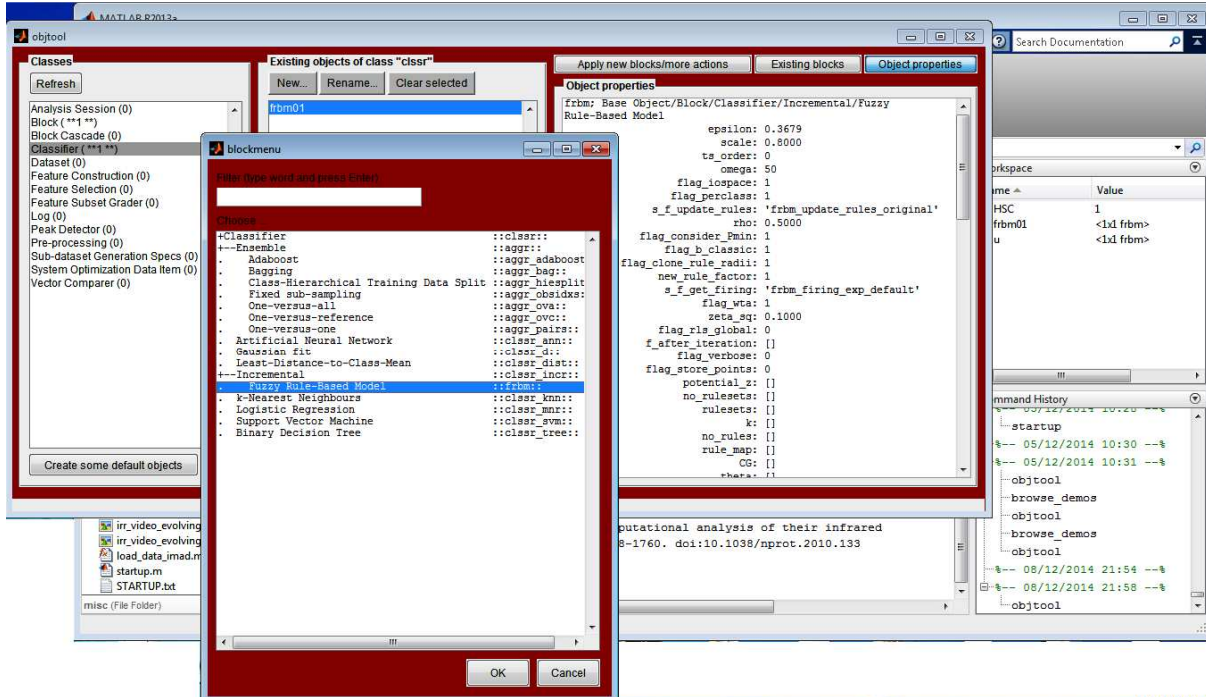


Figure IV-3 : les différents concepts et classifieurs proposés dans la boîte à outils IRootLab.

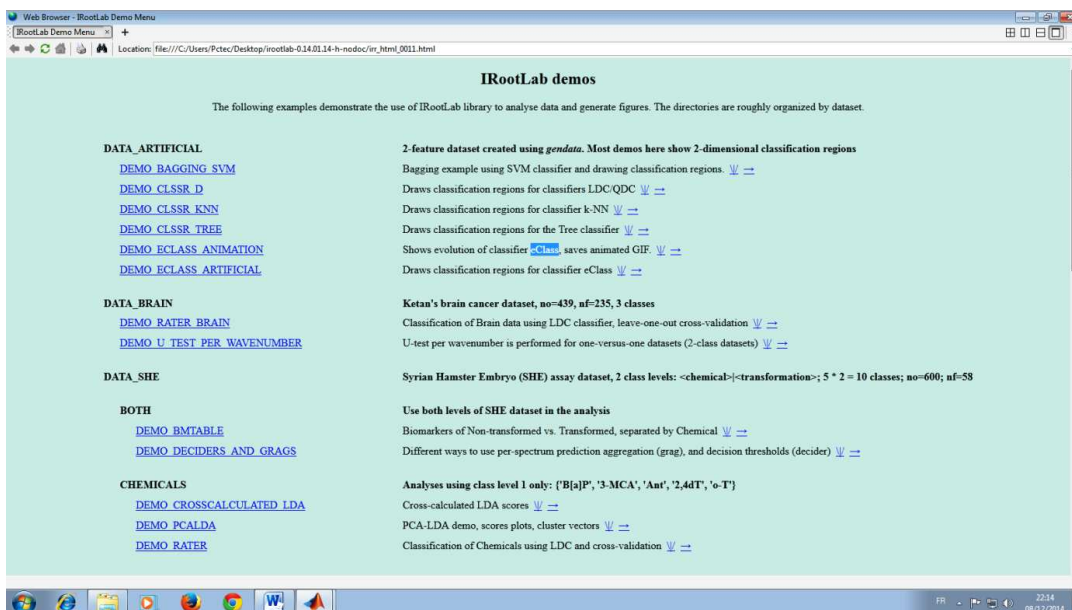


Figure IV-4 : vitrine de démonstrations disponibles sur IRootLab.

5. Résultats Obtenus :

Dans ce travail nous utilisons l'algorithme eclass disponible dans la boîte à outils IRootLab afin de différencier entre les signaux ECG normaux et les signaux correspondant à des arythmies cardiaques. Nous avons appliqué cet algorithme sur la base de données UCI Arrhythmia Repository après l'avoir soumis à un prétraitement pour réduire et sélectionner les attributs pertinents (Figure IV-5).

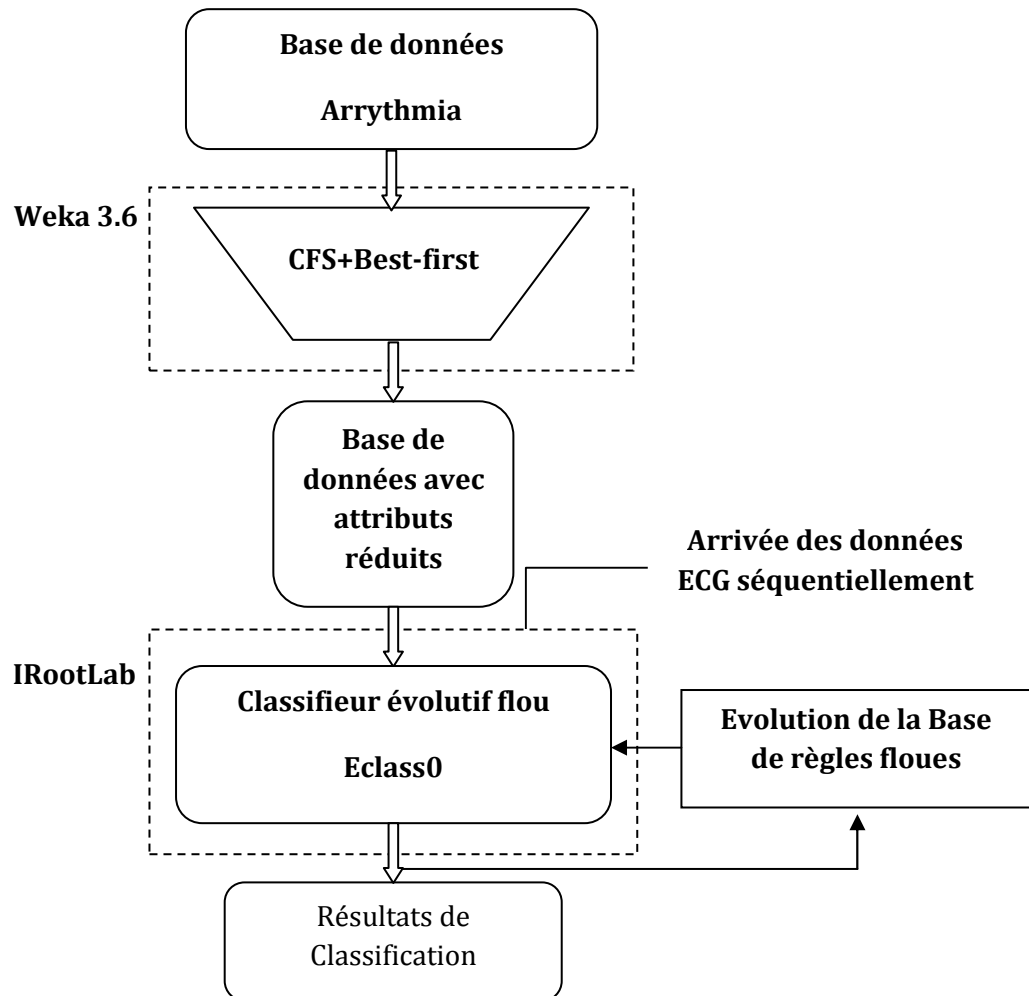


Figure IV-5 : Schéma de la classification en-ligne.

5.1 mesures de performances :

Ce classifieur sera évalué en fonction de sa sensibilité, spécificité, taux de classification de données.

a. Sensibilité (Se) : représente la probabilité que le test soit positif si le signal présente une arythmie.

$$\text{Sensibilité (\%)} = \text{VP} / (\text{VP} + \text{FN}) * 100$$

b. Spécificité (Sp) : représente la probabilité que le test soit négatif si le signal ne présente aucune anomalie.

$$\text{Spécificité (\%)} = \text{VN} / (\text{VN} + \text{FP}) * 100$$

c. Taux de classification :

$$\text{TC} = (\text{VN} + \text{VP}) / (\text{VN} + \text{FN} + \text{VP} + \text{FP}) * 100$$

Avec VP, VN, FP et FN représente respectivement:

- **Vrai positif :** un signal anormal classé anormal;
- **Vrai négatif :** un signal normal classé normal;
- **Faux positif :** un signal normal classé anormal;
- **Faux négatif :** un signal anormal classé normal.

5.2 Expérimentation:

Nous avons appliqué plusieurs classifieurs disponibles dans la même boîte à outils sur notre base de données préalablement traitée, afin de comparer notre classifieur aux autres algorithmes incrémentaux qui sont : K-plus proches voisins, Arbres de décision binaire, et le support vecteur machine.

A noter que l'algorithme eclass0 est la seule méthode en-ligne, les autres méthodes testées sont des méthodes incrémentales en off-ligne.

Le tableau suivant présente les résultats de l'expérimentation menée :

Table IV-2 : performances des différents classifieurs appliqués.

Méthodes	Taux de classification (%)	Sensibilité (%)	Spécificité (%)
Eclass0	86.26	76.78	91.89
KPPV	89.10	85.18	91.50
AD Binaire	84.51	74.57	89.90
SVM	82.69	73.47	87.55

On remarque que le taux de classification obtenu par `eclass0` surclasse la majorité des classifieurs que nous avons testés avec un taux de classification de 86.26 %, ce qui est inférieur de 2.84% par rapport à la meilleure performance qui est obtenue par la méthode du K-plus proche voisin. Par contre nous avons obtenu la meilleure performance en termes de spécificité. Le classifieur le moins performant dans cette expérience est le support vecteur machine avec un taux de bonne classification inférieur de 6.41% par rapport à la meilleure performance.

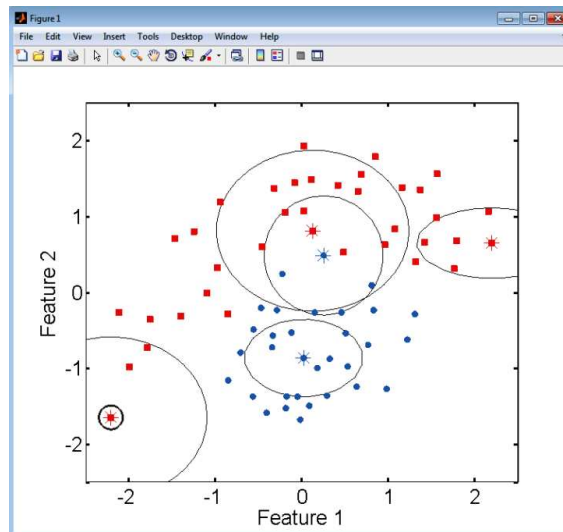


Figure IV-6 : représentation de deux paramètres à l'instant $K=38$.

On peut voir dans la figure ci-dessus la représentation de deux paramètres à l'instant où 38 échantillons de données sont lus. On peut distinguer les points focaux représentés par un soleil ainsi que leurs zones d'influences représentés par des ellipses. Les règles générées sont formées autour de ses points focaux, ensuite elles sont mises à jours au fur et à mesure que d'autres échantillons de données sont lus.

5.3 Comparaison des résultats avec l'état de l'art :

Dans cette section nous comparons les résultats obtenus pour la reconnaissance d'arythmies cardiaques avec ceux obtenus par d'autres auteurs utilisant d'autres méthodes, toujours dans le même but. Le tableau ci-dessous présente les différences dans les taux de classifications.

Table IV-3 : performances de différentes méthodes utilisées dans la reconnaissance d'arythmies cardiaques.

auteurs	méthode	Taux de classification
Ammar(2012)	SMA Neuronal	98.73%
Talbi (2011)	RN MLP	92.72%
Marouf (2010)	Modèle neuronal	98.32%
Zidelmal (2012)	SVM	99.64%
Hendel et al. (2010)	Cartes topologiques de kohonen	95.18%
Chalabi et al (2009)	LVQ	95.90%
Notre travail	Eclass0	86.26%

On remarque depuis le tableau ci-dessus que les résultats obtenus par notre classifieur sont inférieurs aux autres méthodes utilisées par d'autres auteurs. Cela peut s'expliquer par notre manque d'expérience et de connaissance de cette méthode, qui est relativement nouvelle. Mais nous espérons obtenir de meilleurs résultats dans de futurs travaux.

5.4 Résultats obtenus par l'algorithme eclass dans d'autres études dans le domaine médical :

L'algorithme eclass a déjà été utilisé avec succès à des fins de diagnostic médical, le tableau suivant présente les performances obtenues lors de ces études.

Table IV-4 : performances de la méthode eclass dans le domaine du diagnostic médical.

Auteurs	Domaine d'étude	Taux de classification
Lekkas et Mikhailov (2009)	Cancer du Sein	99.45%
Lekkas et Mikhailov (2010)	Diabète des PIMA	97.55%
Notre travail	Arythmies cardiaques	86.26%

A travers le tableau ci-dessus on peut voir que la méthode eclass est parfaitement adaptée pour être utilisée dans le domaine du diagnostic médical et peut même surclasser la majorité des autres méthodes en termes de performances.

6. Conclusion :

Dans cette étude, nous avons testé l'algorithme eclass pour la reconnaissance d'arythmies cardiaques. Pour cela, nous avons d'abord traité la base de données utilisées afin de réduire le nombre élevé d'attributs qui caractérise cette base. Ensuite, nous avons appliqué un classifieur évolutif flou.

Les résultats obtenus sont encourageants ; sachant que ce travail constitue le premier essai dans le domaine du diagnostic médical en ligne au sein de notre institut, il constitue une base. Nous espérons pouvoir améliorer nos résultats dans des travaux futurs et appliquer le deuxième type de cet algorithme qui est eclass1 afin de proposer une amélioration.

Conclusion générale

Conclusion générale

Conclusion générale

Dans cette étude, l'utilisation de classifieur évolutif à base de règles floues est proposé et appliqué pour construire un système semi-supervisé d'aide au diagnostic et de reconnaissance d'arythmies cardiaques. Dans cette optique, nous voulions distinguer les signaux normaux et les signaux anormaux.

Nous avons appliqué cet algorithme sur la base de données d'arythmies cardiaques de L'université de Californie (UCI Machine Learning Repository: Arrhythmia Data Set) après avoir fait quelques prétraitements. En effet, nous avons appliqué le filtre CFS (Correlation-based Feature Selector) qui est un évaluateur de sous-ensembles d'attributs en le combinant avec la stratégie de recherche heuristique « Best-First » afin de réduire le nombre élevé d'attributs.

Les résultats obtenus indiquent des performances assez encourageantes (86.26% de bonne classification) et honorables en comparaison avec d'autres méthodes efficaces décrites dans la littérature.

Ce classifieur évolutif présente des atouts majeurs par rapport aux autres classifieurs hors-ligne à structures fixes, car sa flexibilité et le fait qu'il peut apprendre à partir d'une base de règles vide, montrent de manière significative, la façon dont le classificateur peut être utilisé dans des applications en temps réel. En outre, les résultats expérimentaux démontrent clairement le potentiel du système eClass dont la méthodologie porte la promesse de nouveaux développements significatifs dans le domaine de l'apprentissage et sur les systèmes de classification en ligne.

Notre travail futur se concentrera sur l'amélioration des résultats ainsi que l'application du deuxième type d'algorithme eclass à savoir eclass1. Nous examinerons encore plus cette méthode de classification en temps réel afin de proposer notre propre contribution pour l'optimiser.

Références Bibliographique

References Bibliographiques:

[**Acharya et al., 2007**] U.R. Acharya, S.J. Suri, A.E.J. Spaan, S.M. Krishnan, *ADVANCES IN CARDIAC SIGNAL PROCESSING*, Springer-Verlag Berlin Heidelberg, 2007.

[**Afonso et Tompkins. 1999**] O. Afonso et W.J. Tompkins. Classification of premature ventricular complexes using filter bank features, induction of decision trees and a fuzzy rule based system. *Trans. On biomedical engineering*, vol 46, no 2, pages 192-202, 1999.

[**Aggarwal et al., 2004**] Aggarwal CC, Han J, Wan J, Yu PS. On demand classification of data streams. In: *Proceedings of the international conference in knowledge discovery data mining*; 2004. p. 503–8.

[**Afonso et Tompkins, 1999**]

[**Almaksour et Anquetil, 2009**] Almaksour, A. and Anquetil, E. (2009). Fast incremental learning strategy driven by confusion reject for online handwriting recognition. In *Tenth International Conference on Document Analysis and Recognition (ICDAR 2009)*, pages 81–85.

[**Almaksour et Anquetil, 2010**] Almaksour, A. and Anquetil, E. (2010). Improving premise structure in evolving takagi-sugeno neuro-fuzzy classifiers. In *Proceedings of the Ninth International Conference on Machine Learning and Applications (ICMLA 2010)*.

[**Almaksour, 2011**] Almaksour, A. (2011). *Incremental learning of evolving fuzzy inference systems : application to handwritten gesture recognition*. PhDthesis, Institut National des Sciences Appliquées (INSA) de Rennes.

[**Amar, 2012**] Mr. Belgacem Amar these de magister. Theme Classification des signaux ECG avec un system-multi-agent neuronal. 2012

[**Angelov, 2002**] P. Angelov, *Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems*. Berlin, Germany: Springer-Verlag, 2002.

[**Angelov, 2004**] P. Angelov, “An approach for fuzzy rule-base adaptation using on-line clustering,” *Int. J. Approx. Reason.*, vol. 35, no. 3, pp. 275–289, Mar. 2004.

[**Angelov et Buswell, 2001**] P. Angelov and R. Buswell, Evolving Rule-based Models: A Tool for Intelligent Adaptation, *Proc. of the 9th IFSA World Congress*, Vancouver, BC, Canada, 25-28 July 2001, pp.1062-1067.

[**Angelov et Buswell, 2002**] P. Angelov and R. Buswell, “Identification of evolving rule-based models,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 667–677, Oct. 2002.

[**Angelov et Filev, 2004**] Angelov, P. and Filev, D. (2004). An approach to online identification of takagi-sugeno fuzzy models. *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, 34(1) :484 – 498.

Références Bibliographiques

- [**Angelov et Filev, 2004b**] P. Angelov and D. Filev, “An approach to on-line identification of evolving Takagi–Sugeno models,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 484–498, Feb. 2004.
- [**Angelov et Filev, 2005**] P. Angelov and D. Filev, “Simpl_eTS: A simplified method for learning evolving Takagi–Sugeno fuzzy models,” in *Proc. 2005 IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Reno, NV, May 22–25, pp. 1068–1073.
- [**Angelov et Zhou, 2006**] P. Angelov and X. Zhou, “Evolving fuzzy systems from data streams in real-time,” in *Proc. 2006 Int. Symp. Evol. Fuzzy Syst.*, Sep. 7–9, pp. 29–35.
- [**Angelov et al., 2004**] P. Angelov, C. Xydeas, and D. Filev, “On-line identification of MIMO evolving Takagi–Sugeno fuzzy models,” in *Proc. Int. Joint Conf. Neural Netw. Int. Conf. Fuzzy Syst. (IJCNN-FUZZ-IEEE)*, Budapest, Hungary, Jul. 25–29, 2004, pp. 55–60.
- [**Angelov et al., 2006**] P. Angelov, V. Giglio, C. Guardiola, E. Lughofer, and J. M. Lujan, “An approach to model-based fault detection in industrial measurement systems with application to engine test benches,” *Meas. Sci. Technol.*, vol. 17, no. 7, pp. 1809–1818, 2006.
- [**Angelov et al. 2007**] Angelov P, Zhou X, Klawonn F. Evolving fuzzy rule-based classifiers. In: Proceeding of the IEEE symposium on computational intelligence applications in image and signal processing; 2007.p. 220–5.
- [**Angelov et al., 2007b**] Angelov, P., Zhou, X., Filev, D., Lughofer E.: Architectures for evolving fuzzy rule-based classifiers. IEEE International Conference on Systems, Man and Cybernetics, ISIC. doi:10.1109/ICSMC.2007.4413728 (2007).
- [**Angelov et Zhou, 2008**] Angelov P, Zhou X. Evolving fuzzy rule-based classifiers from data streams. IEEE Transactions of Fuzzy Systems Special Issue on Evolving Fuzzy Systems 2008;16(6):1462–75.
- [**Anugerah et al.; 2013**] Anugerah Galang Persada, Noor Akhmad Setiawan, and Hanung Adi Nugroho, “Comparative Study of Attribute Reduction on Arrhythmia Classification Dataset,” in *Information Technology and Electrical Engineering (ICITEE)*, IEEE, 2013, pp. 68 – 72.
- [**Astroem et Wittenmark, 1989**] K. J. Astroem and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.
- [**Asuncion et Newman, 2007**] A. Asuncion and D. J. Newman. (2007). UCI Machine Learning Repository [Online]. School of Information and Computer Science, University of California, Irvine. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [**Beasley, 2003**] B.M. Beasley. *Understanding EKGs, A practical approach*. Prentice Hall, 2003.
- [**Blondeau et Hiltgen, 1980**] M. Blondeau, M. Hiltgen, *ELECTROCARDIOGRAPHIE CLINIQUE*, Masson, 1980.

Références Bibliographiques

- [**Borries et al., 2005**] R.F. Borries , H.J Pierluissi et H.Nazeran. Wavelet transform-Based ECG baseline drift removal for Body surface potential mapping. In proc of the 27th Annual conference on engineering in medicine and biology, Shanghai.Pages 3891-3894, 2005.
- [**Bouillon, 2012**] Manuel Bouillon, INSA de Rennes Master 2 Recherche en Informatique Apprentissage incrémental et décrémental Classification avec un système d'inférence floue évolutif appliquée à la reconnaissance de gestes manuscrits. 5 juin 2012.
- [**Carpenter et al.,1992**] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 698–712, Sep. 1992.
- [**Casillaset al., 2002**] J. Casillas, O. Cordon, and F. Herrera, "COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 4, pp. 526–537, Aug. 2002.
- [**Chalabi et al., 2009**] Zid-el-khir CHALABI, Abderhim BOUDJEMAOUI, Larbi SAADIA, et Nasreddine BERRACHED. Détection et Classification Automatiques d'ArythmiesCardiaques. 2009
- [**Chai, Ng, etChieu, 2002**]K. M. A. Chai, H. T. Ng, and H. L. Chieu, "Bayesian online classifiers for text classification and filtering," in *Proc. SIGIR 2002*, Tampere, Finland, Aug. 11–15, pp. 97–104.
- [**Chen et Aghajan,2006**] W. Chen, H. Aghajan , "Real-Time human pose estimation: a case study in algorithm design for smart camera networks" in *Proceedings of the IEEE*, vol. 96, no. 10, 2006, pp.1715-1732.
- [**Chen et al., 2006**] S.W.Chen, C.H.Chen et H.L.Chan. Areal –time QRS detection method based on moving-averaging incorporating with wavelet denoising . *comput. Method.Program.Biomed.*Vol 82. No. 3, pages 187-195, 2006.
- [**Chiu, 1994**] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, pp. 267–278, 1994.
- [**Christov et al., 2005**] I.Christov, I.Jecova et G.Botolan. Premature ventricular contraction classification by the Kth nearest-neighbours rule. *Physiol. Meas*, vol 6. Pages 123-130, 2005.
- [**Christov et al., 2006**] I.Christov, G.G.Herrero et V.Krasteva. Comparative study of morphological and time-frequency ECG descriptors for heartbeat classification, 2006.
- [**Clifford et al., 2006**] G.D. Clifford, F. Azuaje, P.E. McSharry, *ADVANCED METHODS AND TOOLS FOR ECG DATA ANALYSIS*, Artechouse, 2006.

Références Bibliographiques

- [Cordon et al., 2004] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, “Ten years of genetic fuzzy systems: Current framework and new trends,” *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 5–31, 2004.
- [Cordon et al., 2004] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, “Ten years of genetic fuzzy systems: Current framework and new trends,” *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 5–31, 2004.
- [Cornuéjols et Miclet, 2002] Cornuéjols, A. and Miclet, L. (2002). *Apprentissage artificiel*.
- [De-Chazal et al., 2004] P. De-Chazal, M. Dwyer et R. Reilly. *Automatic classification of heartbeats using ECG morphology and heartbeat interval features*. IEEE Trans. on Biomedical Engineering, vol. 51, no. 7, pages 1196-1206, 2004.
- [De-Chazal et Reilly, 2006] P. De-Chazal, R. Reilly. A patient adapting heartbeat classifier using ECG Morphology and heartbeat interval features. IEEE Trans. on Biomedical Engineering, vol. 53, no. 12, pages 2535-2543, 2006.
- [Domingos et Hulten, 2001] P. Domingos and G. Hulten, “Catching up with the data: Research issues in mining data streams,” presented at the Workshop Res. Issues Data Mining Knowl. Discovery, Santa Barbara, CA, 2001.
- [Dominguez et Dubuc, 2005] Manuel Dominguez, Marc Dubuc « Arythmie cardiaque: guide d'apprentissage » PUM. 2005. 148 pages.
- [Dubois, 2004] R. Dubois « Application des nouvelles méthodes d'apprentissage à la détection précoce d'anomalies cardiaques en électrocardiographie » thèse doctorat 2004.
- [Duda et al., 2000] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Chichester, U.K.: Wiley-Interscience, 2000.
- [Einthoven, 1941] W. Einthoven, "The galvanometric registration of the human electrocardiogram, likewise a review of the use of capillary-electrometer in physiology", in *Cardiac classics: Willius FW-CV Mosby*, St Louis, 1941.
- [Fayyad et al., 1996] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. *From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: MIT Press, 1996.
- [Ferrer-Troyano et al., 2005] F. Ferrer-Troyano, J. S. Aguilar-Ruiz, and J. C. Riquelme, “Incremental rule learning based on example nearness from numerical data,” in *Proc. 2005 ACM Symp. Appl. Comput.*, 2008, pp. 568–572.
- [Fritzke, 1994] B. Fritzke, “Growing cell structures – a self-organizing network for unsupervised and supervised learning”, *Neural Networks*, vol. 7, No 9, pp. 1441-1460, 1994.

Références Bibliographiques

- [**Ghodratnama et al., 2010**] S. Ghodratnama and M. R. Moosavi and M. Taheri and M. Zolghadri Jahromi, “A cost sensitive learning algorithm for intrusion detection”, Proceedings of ICEE 2010, May 11-13, 2010.
- [**Goldberg, 1989**] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [**Gomez et al., 2002**] J. Gomez, F. Gonzalez, D. Dasgupta, and O. Nasaroui, “Complete expression tree for evolving fuzzy classifier systems with genetic algorithms,” in *Proc. North Amer. Fuzzy Inf. Process. Soc. Conf. Fuzzy Logic Internet (NAFIPS-FLINT)*, 2002, pp. 469–474.
- [**Gomez et al., 2006**] H.G.Gomez, I.Jecova, I.Christovet V.Krasteva. Relative estimation of the Karhunen-Loève Transform basis function for detection of ventricular Ectopic beats. *Computers in cardiology*, vol. 33. Pages 569-572, 2006.
- [**Guvenir et al., 1997**] Guvenir H. A., Acar B, Demiroz G, Cekin A., “A supervised machine learning algorithm for arrhythmia analysis”, *Computers in Cardiology*. 1997 Sep; 433 – 436.
- [**Hamzeloo et al., 2011**] S. Hamzeloo, H. Shahparast, M. Taheri and M. Zolghadri. “Weight adjusting method with non-differentiable activation functions for neural networks”, 2011. (In Persian).
- [**Hastie et al., 2001**] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Heidelberg, Germany: Springer-Verlag, 2001.
- [**Hall ; 1999**] Mark A Hall, *Correlation-based Feature Selection for Machine Learning*, Ph.D. thesis, University of Waikato, 1999.
- [**hendel et al., 2010**] Mouniahendel, Abdelkader Benyettou, Hendel Fatiha, Khelil Hiba. Classification des signaux ECG base sur la transformée en ondelettes et les réseaux de neurones à apprentissage supervisé et non supervisé. 2010
- [**Holland, 1975**] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.
- [**Homeira Shahparast et al., 2012**] Homeira Shahparast, Mohammad Taheri, Sam Hamzeloo, Mansoor Zolghadri Jahromi Dept. of Computer Science, Engineering & IT Shiraz University Shiraz, Iran « An Online Rule Weighting Method to Classify Data Streams » The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012) 978-1-4673-1479-4/12/\$31.00 ©2012 IEEE.

Références Bibliographiques

- [**Hu et al., 1997**] Y.H. Hu, S.H.Paleady et W.J.Tompkins. A patient adaptable ECG beat classifier using a mixture of experts approach. *Trans. On biomedical engineering*, vol 44, no 9, pages 891-900, 1997.
- [**Hunt et al., 1996**] Hunt, K., Haas, R., and Murray-Smith, R. (1996). Extending the functional equivalence of radial basis function networks and fuzzy inference systems. *Neural Networks, IEEE Transactions on*, 7(3) :776 –781.
- [**IRISA, 2005**] IRISA - Institut de Recherche en Informatique et Systèmes Aléatoires - www.irisa.fr. livret flyer porte ouverte 14, 15, 16 octobre 2005.
- [**Ishibuchi et al., 1995**] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, “Selecting fuzzy if–then rules for classification problems using genetic algorithms,” *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 260–270, Aug. 1995.
- [**Ishibuchi et al., 2004**] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling With Linguistic Granules: Advanced Information Processing*. Berlin, Germany: Springer-Verlag, 2004.
- [**Ishibuchi et T. Yamamoto, 2005**] H. Ishibuchi and T. Yamamoto, “Rule Weight Specification in Fuzzy Rule-Based Classification systems”. *IEEE T. Fuzzy Systems* 13(4): 428-435, 2005.
- [**Javier Andreu et al., 2006**] Javier Andreu, Rashmi Dutta Baruah, Plamen Angelov, "Real Time Recognition of Human Activities from Wearable Sensors by Evolving Classifiers", 2011 IEEE International Conference on Fuzzy Systems June 27-30, 2011, Taipei, Taiwan
- [**Jin et Agrawal, 2003**] R. Jin and G. Agrawal, “Efficient Decision Tree Construction on Streaming Data”, *Proc. of ACM SIGKDD*, 2003.
- [**Kasabov, 2001**] Kasabov, N. (2001). Evolving fuzzy neural networks for supervised/unsupervised on-line knowledge-based learning.
- [**Kasabov et Song, 2002**] Kasabov, N. and Song, Q. (2002). Denfis : Dynamic evolving neural fuzzy inference system and its application for time-series prediction.
- [**Klawonn et Klement, 1997**] F. Klawonn and P. E. Klement, “Mathematical analysis of fuzzy classifiers,” *Lect. Notes Comput.Sci.*, vol. 1280, pp. 359–370, 1997.
- [**Klinkenberg et Joachims, 2000**] R. Klinkenberg and T. Joachims, “Detection concept drift with support vector machines,” in *Proc. 7th Int. Conf. Mach. Learning (ICML)*, 2000, pp. 487–494.
- [**Klir et Folger, 1988**] G. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [**Kohler et al., 2002**] B. Kohler et C. Henning et Orgmeister. The principle of software QRS detection. *IEEE eng. Med. BIOL. MAG.* Vol. 3, no. 21. Page 42-57, 2002.

Références Bibliographiques

- [**Krasteva et Jecova, 2007**] V. Krasteva et I. Jecova. QRS template matching for recognition of ventricular ectopic beats. *Anal. of biomedical engineering*, vol 55, No. 12, pages 2065-2076, 2007.
- [**Kuncheva et Steimann, 1999**] Kuncheva LI, Steimann F. Fuzzy diagnosis (editorial). *Artificial Intelligence in Medicine* 1999;16(2):121–8.
- [**Kuncheva, 2000**] L. I. Kuncheva, “How good are fuzzy if–then classifiers?” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 4, pp. 501–509, Aug. 2000.
- [**Lagerholm et al., 2000**] M. Lagerholm, G. Peterson et L. Edenbrandt. Clustering ECG complex using Hermite Functions and self organizing maps. *IEEE trans. Biomed. Eng.*, vol 47, no. 7, pages 838-848, 2000.
- [**Lake, 1990**] C. Lake. *Clinical monitoring*. Saunders, Philadelphia. 1990.
- [**Mikhailov, 2008**] Lekkas S, Mikhailov L. Breast cancer diagnosis based on evolvable fuzzy classifiers and feature selection. In: Allen T, Ellis R, Petridis M, editors. *Proceedings of the 28th international conference on innovative techniques and applications of artificial intelligence*. Cambridge, UK: Springer; 2008. p.185–95.
- [**Lekkas et Mikhailov, 2010**] Stavros Lekkas, Ludmil Mikhailov «Evolving fuzzy medical diagnosis of Pima Indians diabetes and of dermatological diseases », *Artificial Intelligence in Medicine* 50 (2010) 117–126.
- [**Lughofer, 2008**] Lughofer, E. (2008b). Flexfis : A robust incremental learning approach for evolving takagi-sugeno fuzzy models. *Fuzzy Systems, IEEE Transactions on*, 16(6) :1393 – 1410.
- [**Lughofer et Angelov, 2009**] Edwin Lughofer. Plamen Angelov. « Detecting and Reacting on Drifts and Shifts in On-Line Data Streams with Evolving Fuzzy Systems ». *IFSA-EUSFLAT*. 2009.
- [**Lughofer et Angelov, 2011**] Lughofer, E. and Angelov, P. (2011). Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Appl. Soft Comput.*, 11(2) :2057–2068.
- [**Lühr et Lazarescu, 2009**] Lühr, S. and Lazarescu, M. (2009). Incremental clustering of dynamic data streams using connectivity based representative points. *Data & Knowledge Engineering*, 68(1) :1 – 27.
- [**Maloof et Michalski, 2004**] Maloof, M. A. and Michalski, R. S. (2004). Incremental learning with partial instance memory. *Artif. Intell.*, 154(1-2) :95–126.
- [**Mamdani, 1977**] Mamdani, E. (1977). Application of fuzzy logic to approximate reasoning using linguistic synthesis. *Computers, IEEE Transactions on*, C-26(12) :1182 –1191.
- [**Mansooriet al., 2007**] M. J. Zolghadri, S. D. Katebi, “A weighting function for improving fuzzy classification systems performance”, *Fuzzy Sets and Systems*, 158(5): 583-591, 2007.

Références Bibliographiques

- [**Marey, 1876**] E. Marey, "Des variations électriques des muscles et du coeur en particulier, étudiées au moyen de l'électromètre de M. Lippmann", CR AcadSci (Paris), vol. 82, 1876.
- [**Marieb, 1999**] E. Marieb, ANATOMIE ET PHYSIOLOGIE HUMAINES, DeBoeck Université, Paris, 4 édition, 1999.
- [**Mark et Moody, 1988**] R. Mark G. Moody. MIT-BIH Arrhythmia database directory .Massachusetts inst. Technology. (MIT), 1988.
- [**Marouf, 2010**] Radjà MAROUF thèse de magistère en informatique. Option : intelligence Artificielle et Aide à la Décision. Thème Amélioration de l'apprentissage d'un Modèle Neuronal pour la reconnaissance des anomalies cardiaques soutenue en Juillet 2010.
- [**Martin, 2005**] Martin, T. (2005) Fuzzy Sets in the fight against digital obesity. *Fuzzy Sets and Systems*, 156 (3), 411–417.
- [**Minami et al., 1999**] K. Minami, H. Nakajima et T. Toyoshima. Real-time discrimination of ventricular tachyarrhythmia with fourier-transform neural network. . IEEE trans. Biomed. Eng, vol 46, no. 2, pages 179-185, 1999.
- [**Minku et al., 2009**] Minku, F. L., Inoue, H., and Yao, X. (2009). Negative correlation in incremental learning. *Natural Computing : an international journal*, 8 :289–320.
- [**Moody et al., 1984**] G. Moody, W. Muldrow et R. Mark. A noise stress test for arrhythmia detectors. IEEE computers in cardiology, no 11, pages 381-384, 1984.
- [**Moody et Mark., 1989**] G. Moody et R. Mark. QRS morphology representation and noise estimation using the Karhunen-Loève transform. Computers in cardiology, vol 16, pages 269-272, 1989.
- [**Naisbitt, 1988**]. Naisbitt, J. (1988) Megatrends: Ten New Directions Transforming Our Lives, Grand Central Publ.
- [**Nauck et Kruse, 1997**] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy Sets Syst.*, vol. 89, pp. 277–288, 1997.
- [**Newman, 1987**] B Newman, JV Selby, MC King, C Slemenda, R Fabsitz, and GD Friedman. Concordance for type 2 (noninsulin-dependent) diabetes mellitus in males twins. Technical report, *Diabetologia*, 30 :pp. 763–768., 1987.
- [**Norman, 1986**] Norman, D.A. (1986). Reflections on cognition and parallel distributed processing. In: parallel distributed processing. McClelland, J.R. Rumelhart, D. and the PDP research group (ED). Cambridge, MA. MIT press.
- [**Osowski et al., 1997**] S. Osowski, L.T Hoai et T. Markiewicz. support vector machine based expert system for reliable heartbeat recognition. Trans. Biomed. Eng Vol. 51, no. 4. Page 582-589, 1997.

Références Bibliographiques

- [Pang et al., 2005] S. Pang, S. Ozawa, and N. Kasabov, “Incremental linear discriminant analysis for classification of data streams,” *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 35, no. 5, pp. 905–914, Oct. 2005.
- [Patra, 2010] P. D. Patra. Integration of FCM, PCA and Neural Networks for classification of ECG Arrhythmias. *IAENG International journal of computer science*, vol. 36, no.3, 2010.
- [Polikar et al., 2001] Polikar, R., Udpa, L., Udpa, S., Member, S., Member, S., and Honavar, V. (2001). Learn++ : An incremental learning algorithm for supervised neural networks. *IEEE Transactions on System, Man and Cybernetics (C), Special Issue on Knowledge Management*, 31 :497–508.
- [Ramezani et al., 2008] Ramezani, R., Angelov, P., and Zhou, X. (2008). A fast approach to novelty detection in video streams using recursive density estimation. In *Intelligent Systems, 2008. IS '08. 4th International IEEE Conference*, volume 2, pages 14–2 –14–7.
- [Robertson et Reid, 2006] N. Robertson and I. Reid, “A general method for human activity recognition in video” in *Comput. Vis. Image Underst.*, Vol. 104, 2006, pp. 232-248.
- [Roubos et al., 2003] J. Roubos, M. Setnes, and J. Abonyi, “Learning fuzzy classification rules from data,” *Inf. Sci.*, vol. 150, pp. 77–93, 2003.
- [Ruping, 2001] Ruping, S. (2001). Incremental learning with support vector machines. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 641 –642.
- [Schneider, 1987] Schneider, W. (1987), Connectionism: Is it a paradigm shift for psychology? *Behavior research methods, instruments, and computers*: 19: 73-83.
- [Sende, 2009] Jean Sende « guide pratique de l’ECG ». 2^{ème} édition– 2009.
- [Simon, 1983] Herbert A. Simon, Why should machines learn?, *Machine Learning : An artificial intelligence approach* (R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, eds.), vol. I, Morgan Kaufmann, 1983.
- [Steimann, 2001] Steimann F. On the use and usefulness of fuzzy sets in medical AI. *Artificial Intelligence in Medicine* 2001;21:131–7.
- [Syed et al., 1999] Syed, N. A., Liu, H., and Sung, K. K. (1999). Handling concept drifts in incremental learning with support vector machines. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '99*, pages 317–321, New York, NY, USA. ACM.
- [Taheri et al., 2009] M. Taheri and E. Chitsaz and S. D. Katebi and M. Zolghadri Jahromi, “A Novel Piecewise Linear Clustering Technique Based on Hyper Plane Adjustment”, *Communications in Computer and Information Science*, 2009, Volume 6, Part 1, 1-8.
- [Talbi, 2011] Mohamed lamine Talbi, these de doctorat en sciences. Theme analyse et traitement du signal électrocardiographique (ECG), 2011.

Références Bibliographiques

- [**Takagi and Sugeno, 1985**] Takagi, T. and Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1) :116–132.
- [**Turing, 1950**] Turing, A.M. (1950), Computing Machinery and intelligence. *Mind* 59:433-460.
- [**Trevisan et al., 2013**] Trevisan, J., Angelov, P.P., Scott, A.D., Carmichael, P.L. & Martin, F.L. (2013) "IRootLab: a free and open-source MATLAB toolbox for vibrational biospectroscopy data analysis". *Bioinformatics* 29(8), 1095-1097. doi: 10.1093/bioinformatics/btt084.
<http://bioinformatics.oxfordjournals.org/content/early/2013/03/12/bioinformatics.btt084.short>
- [**Vapnik, 1998**] V. N. Vapnik, *The Statistical Learning Theory*. New York: Springer-Verlag, 1998.
- [**Waller, 1893**] A. Waller, An introduction to human physiology. New-York: LongmansGreen, 1893.
- [**WidmeretKubat, 1996**] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learning*, vol. 23, no. 1, pp. 69–101, 1996.
- [**Wilson et al., 1934**] Wilson FN, Johnston FD, Macleod AG, Barker PS (1934): Electrocardiograms that represent the potential variations of a single electrode. *Am. Heart J.* 9: 447-71.
- [**Xydeas et al., 2005**] Xydeas C, Angelov P, Chiao S, Reoullas M. Advances in classification of EEG signals via evolving fuzzy classifiers and dependant multiple HMMs. *Computers in Biology and Medicine* 2005;36(10):1064–83.
- [**YageretFilev, 1993**] R. R. Yager and D. P. Filev, "Learning of fuzzy rules by mountain clustering," in *Proc. SPIE Conf. Appl. Fuzzy Logic Technol.*, Boston, MA, 1993, pp. 246–254.
- [**Yang et al., 2010**] T. Yang and L. Cao and C. Zhang, "A Novel Prototype Reduction Method for the K-Nearest Neighbor Algorithm with $K \geq 1$ ", *Lecture Notes in Computer Science*, 2010, vol. 6119/2010, 89-100.
- [**Yeap et al., 1990**] T.H. Yeap, F. Johson et M. Rachiniowski. ECG beat classification by a neural network. Annual international conf of IEEE Eng in Med and Biol. Vol 12, pages 1457-1458, 1990.
- [**Zadeh, 1969**] Zadeh LA Biological application of the theory of fuzzy sets and systems.. In: Proceedings of international symposium on biocybernetics of the central nervous system. Boston, USA: Little Brown and Co.; 1969. p. 199–212.
- [**Zadeh, 1978**] L. Zadeh, "Fuzzy sets as the basis for a theory of possibility," *Fuzzy Sets Syst.*, vol. 1, pp. 3–28, 1978.

Références Bibliographiques

- [Zhou etAngelov, 2006] Zhou X, Angelov P. Real-time joint landmark recognition and classifier generation by an evolving fuzzy system. In: Proceedings of IEEE world congress on computational intelligence; 2006. p. 6314–21.
- [Zhou etAngelov, 2007] Zhou X, Angelov P. Autonomous visual self-localization in completely unknown environment using evolving fuzzy rule-based classifier. In: Proceedings of IEEE symposium on computational intelligence in security and defense applications; 2007. p. 131–8.
- [Zidelmal et al., 2009] Z. Zidelmal, A. Amirouet, A. Belouchrani. using support vector machine SVMs with reject option for heartbeat classification. 2nd international conference on bio-inspired systems and signal processing, no. 2, pages 204-210. 2009.
- [Zidelmal, 2012] Zahia Zidelmal épouse Amirou. Thèse de doctorat université Tizi Ouzou. Thème Reconnaissance d'arythmies cardiaques par Support Vector Machines (SVMs). Soutenu en juin 2012.
- [Zipes et Jalife, 2004] D. P. Zipes, J. Jalife, CARDIAC ELECTROPHYSIOLOGY: FROM CELL TO BEDSIDE, Philadelphia, W.B. Saunders and Company, 2004.
- [Zolghadri Jahromi et Taheri, 2008] M. Zolghadri Jahromi and M. Taheri, “A proposed method for learning rule weights in fuzzy rule-based classification systems”, Fuzzy Sets and Systems, 159(4): 449-459, 2008.
- [Zolghadri Jahromi et al., 2009] M. Zolghadri Jahromi and E. Parvinnia and R John, “A method of learning weighted similarity function to improve the performance of nearest neighbor”, Information Sciences 179, 2964–2973, 2009.

References webographiques:

- [<http://www.stethographics.com>. 2013]
- <http://www.stethographics.com/ihs/french04/anatomy.htm>. octobre 2013.
- [Algerie1, 2012] <http://www.algerie1.com/actualite/445-des-deces-en-algerie-sont-dus-aux-maladies-cardiaques/>.
- [Dermatology dataset. 2008] Dermatology dataset. Available from: <http://archive.ics.uci.edu/ml/machinelearning-databases/dermatology/dermatology.data>. Accessed: 1st of May, 2008.
- [Le dictionnaire visuel, 2013] <http://www.ikonet.com/fr/ledictionnairevisuel/static/qc/coeur>. Octobre 2013.
- [UCI Repository, 1998] *Concepts et algorithmes*. Algorithmes (Paris). Eyrolles. Blake CL, Merz CJ. UCI Repository of Machine Learning Databases, 1998. Available from: <http://archive.ics.uci.edu/ml/datasets.html>.

ملخص:

هذه الدراسة تستكشف منهجية لتطوير مصنف غامض في الوقت الحقيقي. و تسمح بتصنيف البيانات التي سيتم تجهيزه افي الوقت الحقيقي عن طريق تعديل متكرر لقاعدة قوانين الحكم الغامضة. بالإضافة إلى ذلك، فإنه يظهر كيف يمكن تحسين هذه الطريقة و تطبيقها على مجال التشخيص الطبي، و تحديد امراض عدم انتظام ضربات القلب. إن الغالبية العظمى من الأساليب القائمة للتشخيص الطبي تتطلب معالجة البيانات في وقت متأخر و منفصل. للأسف، فإن ذلك يسمح فقط لمحة عن تحليل المجال الحقيقي. إذا أصبحت البيانات الجديدة متاحة فإنها تتطلب حسابات حساسة التكلفة و يرجع ذلك إلى أن إعادة التعلم هو عملية تكرارية. اكلاس هي بنية جديدة نسبيا لتطوير الأنظمة القائمة على قواعد غامضة، والتي تتغلب على هذه المشاكل.

في هذه اللحظة نقدر دقة المصنف بـ 86.26 ٪، ولكن نأمل أن تتحسن قريبا. أظهرت هذه الدراسة أن هذه الطريقة يمكن استخدامها بشكل فعال لتصنيف امراض عدم انتظام ضربات القلب من عينات رقمية منفصلة. وتشير النتائج إلى أن دقة هذا النموذج يمكن تحسينها. أخيرا، يمكن للنظام استخراج قواعد سهلة التفسير من قبل الخبراء الطبيين للحصول على فهم أفضل للعينة قيد التحليل في جميع مراحل عملية التشخيص التقليدية.

الكلمات الرئيسية: المصنف التطويري القائم على قواعد غامضة؛ التشخيص الطبي، تقدير أمراض عدم انتظام ضربات القلب في الوقت الحقيقي.

Résumé :

Ce travail consiste à étudier une méthodologie pour faire évoluer un classifieur flou en ligne, qui permet aux données d'être traitées en temps réel, en modifiant de manière récursive une base de règles floues sur une base par échantillon de flux de données. En outre, il montre comment cette méthode peut être améliorée et appliquée au domaine du diagnostic, pour déceler les arythmies cardiaques.

La grande majorité des méthodes existantes, pour les diagnostics médicaux flous, nécessitent le traitement de données en mode hors ligne, comme un lot. Malheureusement, cela ne permet qu'un aperçu du domaine réel à analyser. Si de nouveaux enregistrements de données deviennent disponibles, ils exigent des calculs de coûts sensibles, en raison du fait que ré-apprendre est un processus itératif. eClass est relativement une nouvelle architecture pour l'évolution des systèmes à base de règles floues, qui surmonte ces problèmes.

Nous estimons la précision du classifieur à 86.26 % mais nous espérons l'améliorer d'ici peu. Cette étude a montré qu'eClass peut être efficacement utilisée pour le classement des arythmies cardiaques à partir d'échantillons numériques discrets. Les résultats indiquent que la précision du modèles eClass peut être encore améliorée. Enfin, le système peut extraire des règles lisibles par l'homme, qui pourraient permettre à des experts médicaux d'obtenir une meilleure compréhension d'un échantillon en cours d'analyse tout au long du processus de diagnostic traditionnel.

Mots-clés: classificateur évolutif ; règles floues; diagnostic en ligne; arythmies cardiaques.

Abstract:

This paper explores a methodology to evolve a fuzzy classifier online that allows data to be processed in real time by modifying recursively a fuzzy rule base on a per- sample data stream. In addition, it shows how this method can be improved and applied to the field of medical diagnosis, to identify cardiac arrhythmias.

The vast majority of existing methods for fuzzy medical diagnostics require the processing of data in offline mode, as a batch. Unfortunately, it only allows a snapshot of the real domain analysis. If new data records become available they require cost sensitive calculations due to the fact that re-learning is an iterative process. eClass is a relatively new architecture for the development of systems based on fuzzy rules, which overcomes these problems.

We estimate the accuracy of the classifier as 86.26 %, but we hope to improve soon. This study showed that eClass can be effectively used for the classification of cardiac arrhythmias from discrete digital samples. The results indicate that the accuracy of eClass models can be further improved. Finally, the system can extract readable rules who could allow medical experts to get a better understanding of a sample under analysis throughout the traditional diagnostic process.

Keywords: evolving classifier; diagnosis online; cardiac arrhythmias; readable rules.