

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 Guelma



Faculté Des Mathématiques, Informatique Et Sciences de la Matière
Département D'informatique
Laboratoire de domiciliation Laboratoire des Sciences et Technologies de l'Information et de la Communication

THÈSE
EN VUE DE L'OBTENTION DU DIPLOME DE
DOCTORAT EN 3^{ème} CYCLE

Domaine : Informatique Filière : Informatique
Spécialité : Informatique

Présentée par

SALAH Halima

Intitulée

Un système bio-inspirés flou pour la reconnaissance des formes incomplètement définies.

Soutenue le : 11/04/2021.

Devant le Jury composé de :

Nom et Prénom	Grade		
Mr. HADDADI Salim	Professeur	Univ. de Guelma	Président
Mr. SERIDI Hamid	Professeur	Univ. de Guelma	Encadreur
Mr. NEMISSI Mohamed	Professeur	Univ. de Guelma	Co-encadreur
Mr. LAFIFI Yacine	Professeur	Univ. de Guelma	Examineur
Mr. FARAH Nadir	Professeur	Univ. de Annaba	Examineur
Mr. KHADIR Med Tarek	Professeur	Univ. de Annaba	Examineur

Année Universitaire : 2020/2021

Remerciements

Par ces quelques lignes, je tiens à remercier toutes les personnes qui ont participé de près ou de loin au bon achèvement de cette thèse.

Je voudrais d'abord exprimer ma grande gratitude et mes remerciements distingués à mes directeurs de thèse Prof.**SERIDI Hamid** et Prof.**NEMISSI Mohamed** pour le sujet passionnant qu'ils m'ont permis d'entreprendre. Je les remercie également pour le temps et la patience qu'ils m'ont accordés tout au long de ces années. Leurs conseils pertinents et leurs interventions toujours claires et avisées m'ont guidé en permanence tout au long de ce travail.

Je tiens également à remercier vivement Prof. **HADDADI Salim** qui m'a fait l'honneur de présider le jury de mon doctorat, ainsi que Prof. **FARAH Nadir**, Prof. **LAFIFI Yacine** et Prof. **KHADIR Med Tarek**, pour avoir accepté de rapporter ma thèse et de faire partie du jury.

Un très grand merci à tous mes collègues membres du laboratoire LabSTIC, pour leur soutien durant ce parcours doctoral.

Enfin, je tiens à exprimer ma gratitude à toute ma famille et en particulier mes parents, pour leur confiance, leur soutien et leur amour, à mon petit prince, mon mari, ma soeur et mes frères.

Résumé

Le travail dans cette thèse a pour objectif de proposer un système de reconnaissance des formes basé sur les règles floues. Afin d'élaborer un système à la fois efficace et compact, nous nous sommes intéressés aux approches de clustering pour la génération des règles floues et aux méta-heuristiques bio-inspirées pour l'optimisation des paramètres du système. Nous avons ainsi mené une étude analytique portant sur l'effet des poids des règles et des fonctions d'appartenance, puis proposé trois modèles de classification. Le premier modèle se base sur l'optimisation des fonctions d'appartenance par fusion. Le deuxième modèle se base sur le clustering pour générer les règles floues et les méta-heuristiques bio-inspirées pour l'optimisation des poids des règles. Dans ce modèle nous avons utilisée l'optimisation par essaim de particules (PSO) et l'algorithme de recherche coucou (cuckoo search). Finalement, le troisième modèle, qui constitue notre principale contribution, se base sur l'utilisation du PSO pour l'optimisation des paramètres de la méthode de clustering. L'idée de base repose sur le fait que la détermination d'un nombre optimal de règles revient à déterminer des valeurs adéquates des paramètres d'entrée de la méthode de clustering. Notre contribution consiste ainsi à considérer la définition de ces paramètres comme un problème d'optimisation et à utiliser le PSO pour cette tâche. Pour évaluer les modèles proposés, nous avons effectué des tests sur plusieurs bases de données et comparé nos résultats avec d'autres méthodes récentes.

Mots clés : reconnaissance des formes , classification , bio-inspirée , fonction d'appartenance , clustering soustractive, Optimisation par essaim de particules (PSO), classification basée sur des règles floues, poids des règles floues.

Abstract

The aim of this thesis is to propose a pattern recognition system based on fuzzy rules. In order to build a system that is both efficient and compact, we focused on clustering approaches for the generation of fuzzy rules and bio-inspired meta-heuristics for the optimization of system parameters. We therefore, conducted an analytical study on the effect of rule weights and membership functions, and then proposed three classification models. The first model is based on the optimization of membership functions by merging. The second model is based on clustering to generate fuzzy rules and bio-inspired metaheuristics for rule weight optimization. In this model we used particle swarm optimization (PSO) and the cuckoo search algorithm. Finally, the third model, which is our main contribution, is based on the use of the PSO for the optimization of the clustering method parameters. The basic idea is based on the fact that determining an optimal number of rules is equivalent to determining adequate values of the input parameters of the clustering method. Our contribution is to consider the definition of these parameters as an optimization problem and to use the PSO for this task. To evaluate the proposed models, we performed tests on several databases and compared our results with other recent methods.

Key words : pattern recognition, classification, bio-inspired, membership function, sub-

stractive clustering, particle swarm optimization, fuzzy rule based classification, rule weights.

المخلص

يهدف العمل في هذه الأطروحة إلى اقتراح نظام التعرف على الأنماط بناءً على قواعد غامضة (ضبابية). من أجل تطوير نظام فعال ومضغوط على حد سواء ، نحن مهتمون بطرائق التجميع لاستحداث قواعد ضبابية والاستدلال الاستعراضي المستوحى حيويًا لتحسين إعدادات النظام. لهذا أجرينا دراسة تحليلية حول تأثير أوزان القواعد ودالات الإنتماء ، ثم اقترحنا ثلاثة نماذج تصنيف. يعتمد النموذج الأول على تحسين دالات الإنتماء عن طريق الدمج. النموذج الثاني مبني على التجميع لاستحداث قواعد ضبابية و استعراضية مستوحاة حيويًا لتحسين أوزان القواعد. استخدمنا في هذا النموذج التحسين بجسيمات السرب (PSO) وخوارزمية بحث الوقواق (Cuckoo Search). أخيرًا ، النموذج الثالث الذي يشكل مساهمتنا الرئيسية يعتمد على استخدام طريقة التحسين بجسيمات السرب PSO لتحسين إعدادات طريقة التجميع. تستند الفكرة الأساسية إلى حقيقة أن تحديد العدد الأمثل للقواعد يصبح تحديد القيم المناسبة لإعدادات الإدخال لطريقة التجميع.

تتمثل مساهمتنا في اعتبار تعريف هذه الإعدادات كمشكلة تحسين، واستخدام التحسين بجسيمات السرب PSO لهذه المهمة. لتقييم النماذج المقترحة، أجرينا اختبارات على العديد من قواعد البيانات وقمنا بمقارنة نتائجنا مع نتائج طرق حديثة أخرى.

Table des matières

Introduction générale	1
1 Reconnaissance des formes et classification	5
1.1 Introduction	5
1.2 Processus de la reconnaissance des formes	6
1.2.1 Acquisition des données	7
1.2.2 Prétraitement	7
1.2.3 Extraction de caractéristiques	7
1.2.4 Classification	8
1.3 Techniques d'apprentissage	8
1.3.1 Apprentissage supervisé	9
1.3.2 Apprentissage non-supervisé	9
1.3.3 Apprentissage semi-supervisé	9
1.3.4 Apprentissage par renforcement	10
1.4 Classification supervisé	10
1.4.1 K-plus proche voisin (KNN)	11
1.4.2 Classificateur Naive Bayes	12
1.4.3 Machine à vecteurs de support	13

1.4.4	Classificateur à base de règles	14
1.5	La classification non-supervisé : Clustering	15
1.5.1	Clustering hiérarchique	15
1.5.2	Clustering par K-means	16
1.5.2.1	L'algorithme C-means flou	17
1.6	Conclusion	17
2	La classification à base des règles floues	19
2.1	Introduction	20
2.2	Concepts fondamentaux du système logique flou	20
2.2.1	Théorie des ensembles flous	20
2.2.2	Opérateurs flous	22
2.2.2.1	L'opérateur Négation (Complément)	22
2.2.2.2	L'opérateur ET (intersection)	23
2.2.2.3	L'opérateur OU (union)	23
2.2.2.4	Norme et conorme triangulaire	24
2.2.3	Fonctions d'appartenance	26
2.2.3.1	Notions de base sur les fonctions d'appartenance floue	27
2.2.3.2	Les formes des fonctions d'appartenance	28
2.3	Structure générale des systèmes flous	31
2.3.1	Fuzzification	32
2.3.2	Base de connaissances floues	33
2.3.3	Moteur d'inférence flou	34
2.3.4	Défuzzification	34
2.4	Le système d'inférence de Mamdani	36
2.5	Le système d'inférence de Sugeno	37
2.6	Classification à base des règles floues	39
2.6.1	Principe de classification floue	39

2.6.2	Méthodes de raisonnement (décision)	41
2.6.3	Domaine d'application de la classification floue	42
2.7	Conclusion	43
3	Optimisation des règles floue par les méthodes bio-inspirées.	45
3.1	Introduction	45
3.2	Des algorithmes bio-inspirés pour la classification basée sur des règles floues (FRBCSs)	46
3.2.1	Algorithme génétique pour les FRBCS	46
3.2.1.1	Brève description de l'algorithme génétique	46
3.2.1.2	Application de l'algorithme génétique aux FRBCS	47
3.2.2	Système immunitaire artificiel pour FRBCSs	49
3.2.2.1	Brève description de Système immunitaire artificiel	49
3.2.2.2	Application de Système immunitaire artificiel pour FRBCSs	50
3.2.3	L'optimisation par colonies de fourmis (ACO) pour FRBCSs	50
3.2.3.1	Brève description de L'optimisation par colonies de fourmis	50
3.2.3.2	Application de L'optimisation par colonies de fourmis pour FRBCS	51
3.2.4	L'optimisation par essaim de particules (PSO) pour FRBCS	52
3.2.4.1	Brève description de L'optimisation par essaim de par- ticules	52
3.2.4.2	Application de l'algorithme d'optimisation par essaim de particules pour FRBCSs	53
3.3	Conclusion	53
4	Contribution et résultats	54
4.1	Introduction	54
4.2	Méthodes et techniques utilisé	55

4.2.1	Clustering en utilisant soustractive clustering	55
4.2.1.1	Initialisation des fonctions d'appartenance par le clustering	55
4.2.1.2	Soustractive clustering (SC)	56
4.2.2	Optimisation par essais particulaires (OEP ou PSO)	58
4.2.3	Optimisation en utilisant l'Algorithme de la recherche Coucou	58
4.2.4	Génération des règles floue	59
4.2.4.1	Formes des règles utilisées	59
4.2.4.2	La procédure de génération des règles floues	60
4.3	Etude théorique sur l'effet des fonctions d'appartenance et des poids des règles sur la classification basée sur des règles floues	61
4.3.1	Effet des poids des règles (CF)	63
4.3.2	Effet des fonctions d'appartenance (MF)	66
4.4	Les modèles proposés	69
4.4.1	1 ^{er} modèle :	69
4.4.1.1	L'idée principale	69
4.4.1.2	Processus	69
4.4.1.3	Test et résultats	71
4.4.2	2 ^{ème} modèle :	76
4.4.2.1	Idée principale	76
4.4.2.2	Processus	77
4.4.2.3	Tests et résultats	79
4.4.3	3 ^{ème} Modèle [91] :	81
4.4.3.1	Idée principale	81
4.4.3.2	Processus	83
4.4.3.3	Test et résultat	85
4.5	Conclusion	92

Conclusion et perspective	94
Bibliographie	96

Table des figures

1.1	Organigramme d'un système de reconnaissance des formes	6
1.2	Classification par la méthode des k-plus proche voisin	12
1.3	Classification par la méthode des Machines à vecteurs de support	14
1.4	Clustering par K-means	17
2.1	Le complément d'une fonction gaussienne	23
2.2	Exemple des opérateur ET et OU de deux fonctions d'appartenance gaussienne	24
2.3	Exemple des fonctions d'appartenance gaussienne	29
2.4	Exemple de la fonction d'appartenance de la cloche	29
2.5	Exemple de la fonction d'appartenance triangulaire	30
2.6	Exemple de la fonction d'appartenance trapézoïdale	31
2.7	Structure générale d'un système basé sur la logique floue	32
2.8	Exemple d'un système d'inférence de Mamdani	37
2.9	Exemple d'un système d'inférence Sugeno	38
2.10	Le processus de classification à base des règles floues	40
4.1	Initialisation des fonctions d'appartenance	56

4.2	Zone de décision de chaque règle floue SI-ALORS avec : (a) $K=3$ et (b) $K=2$	62
4.3	Deux problèmes synthétiques de classification (a) 1 ^{er} problème : linéairement séparable (b) 2 ^{ème} problème : non linéairement séparable.	63
4.4	Frontières de décision sans CF, $K=3$	65
4.5	Frontières de décision avec les CF	65
4.6	Frontières de décision avec CF ajusté	66
4.7	Les frontières de décision avec CF, $K = 2$	67
4.8	Les frontières de décision avec fonction d'appartenance modifiée	67
4.9	Les frontières de décision :(a) avec CF,(b) fonction d'appartenance modifiée sans CF (c) fonctions d'appartenance modifiées avec CF	68
4.10	Shéma générale du premier modèle proposé	70
4.11	Exemple de fusion de deux MF	71
4.12	Fusion des fonctions d'appartenance générées (MF) (a). MFs générées pour la caractéristique 1 (b). MFs fusionnées pour la caractéristique 1 (c). MFs générées pour la caractéristique 2 (d). MFs fusionnées pour la caractéristique 2	72
4.13	Les frontières de décision de l'exemple synthétique par : (a) la méthode heuristique [53], (b) méthode proposée, (c) fusion des MF	73
4.14	Fonctions d'appartenance générées de : (a) la 3 ^{ème} caractéristique de l'Iris (b) la 4 ^{ème} caractéristique de l'Iris	74
4.15	Les frontière de décision des 3 ^{ème} et 4 ^{ème} paramètres de l'iris : (a) avec la méthode heuristique [53],(b) méthode proposée	74
4.16	Shéma générale du 2 ^{ème} modèle proposée (a) la première version (b) la deuxième version	77
4.17	Application de soustractive clustering sur : (a) tous les ensembles de données, (b) et sur chaque classe séparément	82
4.18	Clustering soustractive avec différentes valeurs de rayon	83

4.19 Le schéma général du 3 ^{ème} modèle	84
---	----

Liste des tableaux

2.1	Quelque domaines d'applications de la classification floue	43
4.1	Le taux de classification des deux problèmes de classification avec divers K	64
4.2	Taux de classification avec CF ajusté pour problème de classification 2	66
4.3	Taux de classification du problème de classification 3	67
4.4	Résultats de la classification de l'exemple synthétique.	72
4.5	Résultats de la classification des 3 ^{ème} et 4 ^{ème} paramètres de l'Iris	74
4.6	Résultats de la classification des données Iris	75
4.7	Résultats de la classification des données Wine	75
4.8	Résultats de la classification sur la base Iris	80
4.9	Résultats de la classification sur la base Wine	80
4.10	Résultats de la classification sur la base Wisconsin	81
4.11	Liste des ensembles de données utilisés et leurs spécifications	86
4.12	Les approches utilisées pour la comparaison	88
4.13	Résultats de la classification (précision%) à l'aide de 10 – CV	89
4.14	Résultats de la classification (précision % et nombre de règles) à l'aide de 2 – CV	90

4.15	Test de Wilcoxon pour comparer l'approche proposée $[R^+]$ et les autres méthodes $[R^-]$ en $10 - CV$	91
4.16	Test de Wilcoxon pour comparer l'approche proposée $[R^+]$ et la méthode de Chen et al (2016) [22] $[R^-]$ en $2 - CV$	91

Liste des abréviations

ACO	Optimisation par colonies de fourmis (Ant Colony Optimization)
AIS	Système immunitaire artificiel (Artificial Immune System)
CF	Grade de certitude (Certainty Factor)
CS	Recherche coucou (Cuckoo Search)
FRBCS	Système de classification à base des règles floues (Fuzzy Rule-Based Classification System)
FRBS	Système à base des règles floues (Fuzzy Rule-Based System)
FRM	Méthode de raisonnement floue (Fuzzy Reasoning Method)
GA	Algorithme génétique (Genetic Algorithm)
MF	Fonction d'appartenance (Membership Function)
PSO	Optimisation par essaim de particules (particle swarm optimization)
RDF	(Reconnaissance Des Formes)
SC	Soustractive clustering (Subtractive Clustering)

INTRODUCTION

Cadre générale et problématique

Les progrès technologiques actuels ont donné naissance à une abondance de données collectées. Ces données constituent d'importants outils pour le développement des systèmes automatique de reconnaissance des formes. En revanche, ces données présentent d'énormes imprécisions et incertitudes. Les approches floues sont l'une des solutions les bien connues pour la manipulation de ce type de problèmes.

Dans la classification floue un individu peut appartenir à plusieurs classes à divers degrés en appliquant un ensemble de règles floues de la forme (Si ... Alors ...), basées sur des valeurs linguistiques de ces attributs. Pour construire un système de classification floue, les tâches les plus difficiles et les plus importantes sont : l'initialisation des fonction d'appartenance, la génération, l'apprentissage, la sélection des règles, ainsi que le calcul des degrés de confiance de ces règles. Dans ce contexte, une grande variété de méthodes a été introduite. Ceci comprend : les méthodes heuristiques ([3]; [79]; [107]), les algorithmes génétiques (GA) ([8]; [27]; [72]; [110]), les systèmes immunitaires artificiels (AIS) ([7]; [67]), l'optimisation par les colonies de fourmis (ACO) ([43]; [92]), l'optimisation par essaim de particules (PSO) ([18]; [40]; [61]; [82]), les réseaux de neurones ([21]; [84]).

L'avantage des approches basées sur le clustering, dans lesquelles chaque règle corres-

pond à un cluster, réside dans le fait qu'elles permettent de générer des classificateurs efficaces avec peu de règles. En effet, dans les méthodes traditionnelles de grille-partitions, l'espace d'entrée est divisé en toutes les régions possibles et chacune d'elles est représentée par une règle floue. Ce processus génère un grand nombre de règles, en particulier dans les problèmes de grande dimension où le nombre de règles devient exponentiellement énorme (malédiction de dimensionnalité). Ces règles doivent ensuite être sélectionnées et réglées. D'autre part, dans les approches basées sur le clustering, les paramètres des fonctions d'appartenance, c'est-à-dire les centres et les largeurs, peuvent être automatiquement définis en projetant les clusters obtenus. Dans ce travail, nous proposons trois modèles de classification à base de règles floues en utilisant la méthode de soustractive clustering.

Objectif et Contributions

Les approches basées sur le clustering ont été largement proposées pour l'initialisation des fonctions d'appartenance. Dans ces approches, le processus de création est généralement réalisé en deux étapes : L'application de la méthode de clustering et le réglage des paramètres. Parmi ces approches, [19], [57], [59], [65], [69], [73]. Dans ce travail, nous proposons trois modèles de classification à base de règles floues en utilisant la méthode de soustractive clustering et les méthodes d'optimisation bio-inspirées. Les travaux effectués peuvent être résumés comme suit :

- ✓ Une étude théorique pour l'analyse de l'effet des degrés de confiance des règles (poids des règles) et des fonctions d'appartenance sur la performance des systèmes de classification à base des règles floues. Cette étude est basée sur l'analyse des frontières de décision et des taux de classification.

- ✓ Nous proposons un 1^{er} modèle de classification basé sur des règles floues dans lequel la génération des fonctions d'appartenance s'effectue automatiquement à l'aide de la méthode de soustractive clustering, les fonctions d'appartenance sont ensuite optimisées

par un algorithme de fusion.

✓ Nous proposons un 2^{ème} modèle qui comprend deux étapes. Dans la première, la méthode de soustractive clustering est appliquée aux données d'apprentissage pour obtenir différentes régions et pour l'initialisation des fonctions d'appartenance. Dans la deuxième étape, les poids des règles sont générés automatiquement par une méthode d'optimisation bio-inspirée.

✓ Nous proposons un 3^{ème} modèle, qui constitue notre contribution principale. Notre motivation se base sur le fait que la méthode de soustractive clustering dépend du rayon, dont sa valeur affecte le nombre de clusters obtenus. Par conséquent, cela affecte le nombre de règles générées et les fonctions d'appartenance. Généralement, ce paramètre est prédéfini par l'utilisateur dans l'intervalle [0 1]. L'idée clé de ce travail repose essentiellement sur la détermination automatique des valeurs optimales de ce paramètre afin d'obtenir une base de règles floues efficace et compacte. Le PSO est ainsi utilisé pour trouver l'ensemble de rayons optimaux. Pour obtenir un bon compromis précision-compacité, nous proposons d'utiliser une fonction multi-objective qui minimise le nombre de règles et maximise le taux de classification.

Organisation de la thèse

Cette thèse est organiser comme suit :

Le **premier chapitre** donne une présentation des concepts généraux de la reconnaissance des formes (RDF) et la classification. Il présente le processus général de la RDF, décrit la classification et le clustering, et présente quelques classificateurs de base.

Le **deuxième chapitre** constitue une introduction à la logique floue et ses concepts fondamentaux, les principes de la théorie des sous-ensembles flous, les opérations flous et présente la structure générale d'un système d'inférence floue, puis il présente la classification floue et ses applications.

Le **troisième chapitre** présente un état de l'art sur les méthodes bio-inspirées utilisées

dans la génération et l'optimisation des systèmes de classification basé sur les règles floues.

Le **quatrième chapitre** présente en détail les modèles proposés et l'évaluation de leurs performances en effectuant des tests sur différentes bases de données. Il présente et discute les résultats obtenus et les compare avec les différentes méthodes dans la littérature. Finalement, une **conclusion** conclue ce travail.

Reconnaissance des formes et classification

Sommaire

1.1 Introduction	5
1.2 Processus de la reconnaissance des formes	6
1.3 Techniques d'apprentissage	8
1.4 Classification supervisé	10
1.5 La classification non-supervisé : Clustering	15
1.6 Conclusion	17

1.1 Introduction

La reconnaissance des formes est l'une des branches très importantes et actives de l'intelligence artificielle. C'est la science qui tente à rendre les machines aussi intelligentes que l'homme pour reconnaître les formes et les classer dans les catégories souhaitées de manière simple et fiable. Ce chapitre présente les concepts de base de la reconnaissance des formes, l'architecture du système et fournit la compréhension de divers modèles d'apprentissage et méthodes de la classification.

1.2 Processus de la reconnaissance des formes

La reconnaissance des formes est une discipline scientifique qui a pour but d'automatiser le processus de la prise de décision. La reconnaissance de formes se définit comme l'étude de la manière dont les machines peuvent observer l'environnement, apprendre à distinguer les différents exemples d'intérêt de son arrière-plan et prendre des décisions raisonnables sur les catégories de exemples. Lors de la reconnaissance, les exemples donnés sont affectés à une catégorie prescrite. Les exemples à identifier dépendent de l'application : ils peuvent être des images, des lettres, des paroles, des cibles militaires, des formes d'onde d'un signal, etc. En des termes informatiques, un exemple est un ensemble de valeurs, appelé attributs, auquel est associé un nom (ou étiquette), qui est leur classe.

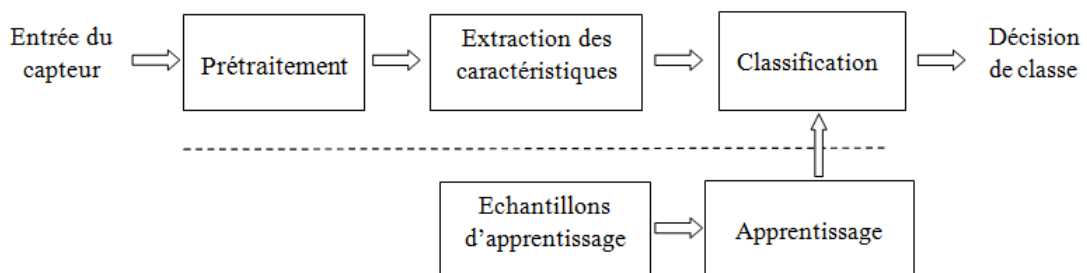


FIGURE 1.1 – Organigramme d'un système de reconnaissance des formes

L'approche classique de la reconnaissance des formes (par exemple, [33]) utilise des données provenant de capteurs arbitraires comme entrée dans un système de classification ; les mesures sont soumises (si nécessaire) à une étape de prétraitement afin d'améliorer les propriétés du signal. L'extraction ultérieure des caractéristiques fournit un vecteur de caractéristiques pour une classification ultérieure. Ce vecteur décrit les mesures d'entrée dans l'espace des caractéristiques.

1.2.1 Acquisition des données

Pour pouvoir une décision, il est généralement nécessaire d'avoir suffisamment d'informations. Pour les systèmes de reconnaissance automatique des formes, ces informations sont souvent fournies par des capteurs [33]. Ces capteurs mesurent des quantités physiques et fournissent une caractéristique analogique ou numérique des objets qui doivent être classés. L'étape de détection/acquisition utilise un capteur tel qu'une caméra ou un microphone. Le signal acquis (par exemple, une image) doit être d'une qualité suffisante pour que les "caractéristiques" distinctives puissent être correctement mesurées. Cela dépendra des caractéristiques du capteur. Ici, les données du milieu environnemental sont prises en compte et transmises au système de reconnaissance des formes. Les données brutes sont ensuite prétraitées soit en éliminant le bruit des données, soit en extrayant le motif d'intérêt de l'arrière-plan afin de rendre l'entrée lisible par le système.

1.2.2 Prétraitement

Les étapes de prétraitement comprennent la standardisation (ou normalisation) des données, et l'élimination ou la réduction de l'effet des valeurs hors-normes. Les données du capteur sont prétraitées en termes de transformation de signal à signal. L'objectif de cette étape est d'améliorer le signal, pour alléger l'analyse ultérieure (réduction de la complexité du signal d'entrée) ou pour augmenter la performance de la classification (amélioration de la classification obtenue). Le prétraitement est un aspect important de l'organigramme de reconnaissance des formes, mais ce n'est pas l'objet de la recherche sur la reconnaissance des formes [35]

1.2.3 Extraction de caractéristiques

L'objectif de l'extraction de caractéristiques est de fournir une représentation des caractéristiques mesurées de l'objet qui facilite la prise de décision ultérieure. Cet objectif peut être atteint en représentant les objets de la même catégorie par des valeurs de carac-

téristiques similaires, et les objets de catégories différentes par des valeurs de caractéristiques distinctes [33].

1.2.4 Classification

La classification est l'étape finale et principale du système de la RDF. Le système de classification ou de description est généralement basé sur la disponibilité a priori d'informations sur un ensemble des échantillons déjà classés ou décrits. Cet ensemble des échantillons est appelé l'ensemble d'apprentissage, et la stratégie d'apprentissage qui en résulte est caractérisée comme un apprentissage supervisé. La classification peut également être non supervisé, dans le sens où le système des échantillons n'est pas étiqueté a priori, mais qu'il établit lui-même les classes sur la base des régularités statistiques des échantillons. C'est ce qu'on appelle l'apprentissage non supervisé.

1.3 Techniques d'apprentissage

D'une façon générale, toute méthode qui intègre des informations provenant d'échantillons d'apprentissage dans la conception d'un classificateur fait appel à l'apprentissage. Du fait que les problèmes réels de la reconnaissance des formes sont généralement si difficiles que nous ne pouvons pas développer facilement le classificateur approprié, l'apprentissage constitue l'étape la plus importante. Le développement d'un classificateur consiste à poser une forme générale de modèle, et d'utiliser des exemples d'apprentissage pour apprendre ou estimer les paramètres inconnus du modèle. L'apprentissage fait référence à une forme d'algorithme permettant de réduire l'erreur sur un ensemble de données d'apprentissage. L'apprentissage se présente sous plusieurs formes générales : Supervisé, non-supervisé, semi-supervisé et par renforcement.

1.3.1 Apprentissage supervisé

Dans l'apprentissage supervisé, un enseignant fournit une étiquette de catégorie pour chaque exemple dans un ensemble d'apprentissage, et nous cherchons à réduire la somme des coûts pour ces exemples. L'ensemble de données d'apprentissage supervisé est un ensemble de données étiqueté. L'apprentissage supervisé tente de trouver les relations entre l'ensemble de caractéristiques et l'ensemble d'étiquettes, c'est-à-dire les connaissances et les propriétés que nous pouvons apprendre à partir de l'ensemble de données étiqueté. Si chaque vecteur de caractéristique x correspond à une *étiquette* $y \in L$, $L=l_1, l_2, \dots, l_c$, le problème d'apprentissage est défini comme un problème de classification. Les connaissances extraites de l'apprentissage supervisé sont souvent utilisées pour la prédiction et la reconnaissance [16].

1.3.2 Apprentissage non-supervisé

L'apprentissage non supervisé peut être défini comme le problème consistant à essayer de trouver la structure cachée dans un ensemble de données non étiquetées. Comme les exemples donnés à l'apprenant ne sont pas étiquetés, chaque algorithme classe lui-même l'ensemble de tests. En termes simples, ici, aucun ensemble d'apprentissage étiqueté n'est fourni et le système applique un regroupement ou une classification spécifique aux ensembles de données non étiquetés sur la base de certains critères de similarité. Ici, l'entrée consiste en quelques valeurs non étiquetées dont la caractéristique distinctive n'est pas connue au départ [98].

1.3.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé a récemment attiré une attention croissante. Il est défini entre l'apprentissage supervisé et non supervisé, contient à la fois des données étiquetées et non étiquetées, et apprend conjointement les connaissances de leur part.

1.3.4 Apprentissage par renforcement

L'apprentissage par renforcement est un domaine de l'apprentissage automatique. Il s'agit de prendre des mesures appropriées pour maximiser la récompense dans une situation particulière. Il est utilisé par divers logiciels et machines pour trouver le meilleur comportement ou la meilleure voie à suivre dans une situation spécifique. L'apprentissage par renforcement diffère de l'apprentissage supervisé dans la mesure où, dans ce dernier, les données d'apprentissage sont accompagnées de la clé de correction, de sorte que le modèle est formé avec la bonne réponse elle-même, tandis que dans l'apprentissage par renforcement, il n'y a pas de réponse mais l'agent de renforcement décide de ce qu'il doit faire pour accomplir la tâche donnée. En l'absence d'un ensemble de données d'apprentissage, il est tenu d'apprendre de son expérience.

1.4 Classification supervisé

Les algorithmes de classification se déroulent en une phase d'apprentissage et une phase de travail [105]. Dans la phase d'apprentissage, les caractéristiques spécifiques des classes sont apprises à partir d'un échantillon représentatif. A ce propos, les objets sont étiquetés avec l'information de la classe à laquelle ils appartiennent et les algorithmes de classification sont fournis avec ces informations. Sur la base de ces informations, un critère de décision est établi. Ce critère peut être représenté par une limite de décision dans l'espace de caractéristiques. Dans la phase de classification, le critère de décision, ou la limite de décision, est utilisé pour les nouvelles entrées du capteur. Les objets nouvellement mesurés sont donc affectés à la classe ou à la catégorie à laquelle ils appartiennent le plus probablement. Différents classificateurs utilisant diverses stratégies d'apprentissage peuvent être utilisés, y compris par exemple les classificateurs de Machine à vecteurs de support et K-plus proche voisin [35].

1.4.1 K-plus proche voisin (KNN)

Dans le domaine de la reconnaissance des formes, l'algorithme du plus proche voisin (k-NN) est une méthode de classification des exemples basée sur des exemples d'apprentissage les plus proches dans l'espace des caractéristiques. K-NN est un type d'apprentissage basé sur des exemples, ou apprentissage paresseux où la fonction est approximée au niveau local seulement et tout le calcul est reporté jusqu'à la classification. Cet algorithme est l'un des algorithmes d'apprentissage automatique les plus simples dans lequel un exemple est classé en utilisant un vote majoritaire de ses voisins et l'objet est ensuite affecté à la classe la plus fréquente parmi ses voisins les plus proches. Ici, les voisins sont pris dans un ensemble d'exemples dont on connaît la classification correcte. Ces voisins peuvent être considérés comme un ensemble d'apprentissage pour cet algorithme, bien que pas d'étape d'apprentissage explicite est nécessaire. L'apprentissage dans ce modèle est basé sur le stockage de tous les exemples d'apprentissage qui correspondent à des points dans un espace euclidien n-dimensionnel ainsi que leurs étiquettes de classe et la classification est retardée jusqu'à l'arrivée d'un nouvel exemple. Lorsque le nouvel exemple à classer ou le nouveau vecteur sans étiquette arrive, la classification est effectuée en attribuant l'étiquette la plus fréquente parmi les k échantillons d'apprentissage les plus proches de cet exemple à classifier, il s'agit de déterminer les distances d_q entre X et chaque vecteur $X^{(q)}$ (figure 1.2), données par : [98].

$$d_q = X - X^{(q)}$$

En suite d'assigner à cet exemple la classe C_m de l'exemple (m) correspondant à la distance la plus petite (d_m), les fonctions discriminantes sont alors :

$$\mathcal{O}_m = \min(d_i) \text{ pour tout } X^{(i)} \in C_m$$

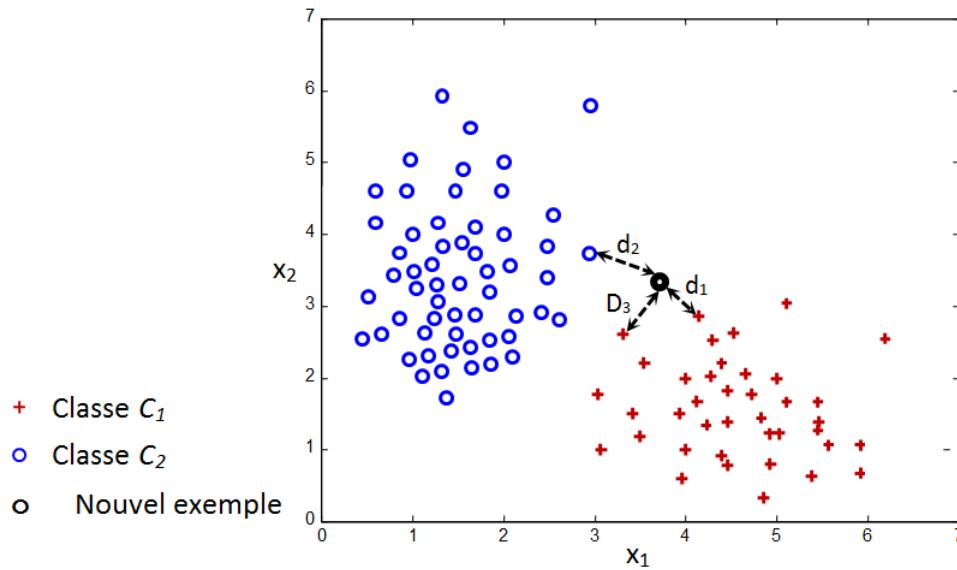


FIGURE 1.2 – Classification par la méthode des k-plus proche voisin

1.4.2 Classificateur Naive Bayes

Le classificateur Naive Bayes est un classificateur simple, probabiliste et statistique qui est basé sur le théorème de Bayes (à partir des statistiques bayésiennes) avec de fortes hypothèses d'indépendance (naïves) et une hypothèse maximale a posteriori. Les classificateurs bayésiens étant de nature statistique, ils peuvent prédire la probabilité qu'un échantillon donné appartienne à une classe particulière. Le modèle de probabilité sous-jacent à ce classificateur peut être qualifié plus justement de "modèle de caractéristique indépendant", car un classificateur Bayes naïf suppose que l'effet d'une valeur d'attribut sur une classe donnée est indépendant des valeurs des autres attributs. Une telle hypothèse est appelée indépendance conditionnelle de classe. Elle est faite pour simplifier le calcul impliqué et, dans ce sens, est considérée comme "naïve". Le classificateur naïf de Bayes est formé en utilisant une approche d'apprentissage supervisé qui nécessite simplement de considérer chaque attribut dans chaque classe séparément. L'apprentissage au classificateur naïf de Bayes est donc considérée comme très facile et rapide. Pour estimer les

paramètres dans le modèle Bayes naïf, il utilise le principe de la méthode du maximum de vraisemblance dans de nombreuses applications pratiques. Les tests dans cet algorithme sont également très directs et simples ; il suffit de regarder les tableaux et de calculer les probabilités conditionnelles avec des distributions normales. L'avantage du modèle Naive Bayes est qu'il ne nécessite qu'une petite quantité de données d'apprentissage pour estimer les paramètres, c'est-à-dire les moyennes et les variances des variables qui sont nécessaires pour la classification [98].

1.4.3 Machine à vecteurs de support

Une machine à vecteurs de support (SVM) effectue la classification en construisant un hyperplan à N dimensions qui sépare de manière optimale les données en deux catégories. Une machine à vecteur de support (SVM) est utilisée en informatique pour un ensemble de méthodes d'apprentissage supervisées connexes qui analysent les données d'entrée et apprennent à partir de celles-ci, puis les utilisent pour effectuer de classification et des analyses de régression. Le SVM standard est un SVM à deux classes qui prend un ensemble de données d'entrée et prédit la classe possible, pour chaque entrée, parmi les deux classes possibles dont l'entrée est membre, ce qui en fait un classificateur linéaire binaire non probabiliste. Étant donné l'ensemble des exemples d'apprentissage où chacun d'entre eux est marqué comme appartenant à l'une des deux classes, l'algorithme d'apprentissage SVM construit un modèle qui attribue de nouveaux exemples dans une classe ou l'autre (figure 1.3).

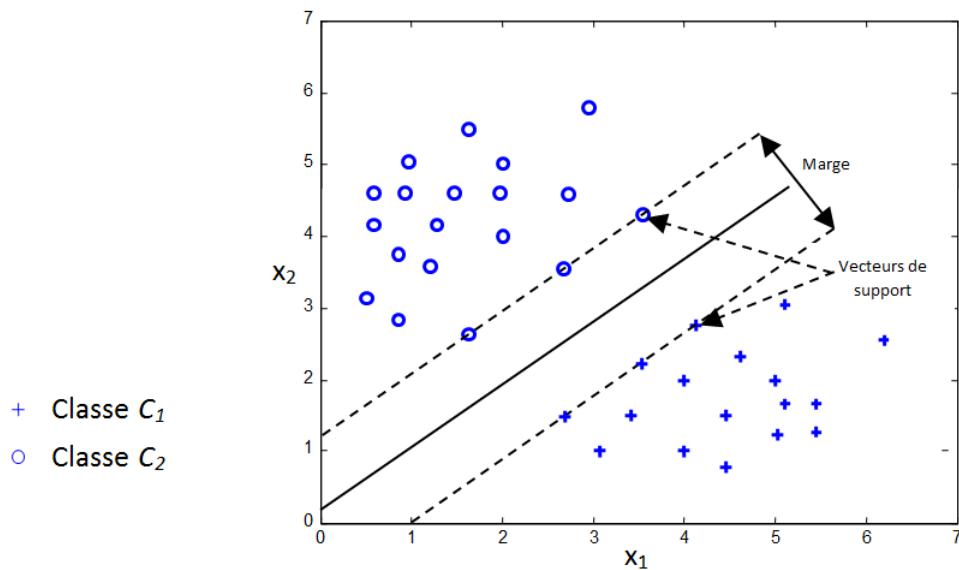


FIGURE 1.3 – Classification par la méthode des Machines à vecteurs de support

1.4.4 Classificateur à base de règles

Dans les problèmes où les classes peuvent être caractérisées par des relations générales, plutôt que par des exemples (instances), il devient intéressant de construire des classificateurs basés sur des règles. Les humains aiment généralement les explications pour la plupart des décisions, et parfois il y a des exigences légales et éthiques pour cela (par exemple, rejet d'une demande de carte de crédit, diagnostic médical). Il est possible d'extraire des règles d'un arbre de décision. Chaque chemin de la racine à la feuille peut être écrit comme une série de règles SI ... ALORS. Lorsque plusieurs feuilles sont étiquetées avec la même classe, alors les chemins peuvent être combinés avec un OU logique. Il peut être possible d'élaguer les règles qui en résultent, mais il est possible qu'elles ne puissent plus être réécrites sous forme d'arbre après cela. Un autre moyen est d'apprendre les règles directement à partir des données. Ce type d'induction de règles est similaire à l'induction d'un arbre de décision, sauf que l'induction de règles effectue une recherche en profondeur et génère une règle (chemin) à la fois, tandis que l'induction d'un arbre va

en largeur et génère tous les chemins simultanément. Les règles sont apprises une par une. Chaque règle est une combinaison de conditions, qui sont ajoutées une à la fois pour minimiser certains critères. Une fois qu'une règle est développée et ajustée, elle est ajoutée à la base de règles et tous les exemples d'apprentissage couverts par cette règle sont retirés de l'ensemble d'apprentissage : et le processus se poursuit jusqu'à ce qu'un nombre suffisant de règles soit généré. Il y a une boucle extérieure qui consiste à ajouter une règle à la fois à la base de règles et une boucle intérieure qui consiste à ajouter une condition à la fois à la règle en cours. Les deux étapes sont gourmandes et ne garantissent pas l'optimalité ; et les deux boucles incorporent un élagage pour une meilleure généralisation.

1.5 La classification non-supervisé : Clustering

Le clustering des données est le processus d'identification des groupements naturels ou des clusters au niveau des données multidimensionnelle sur la base de certaines mesures de similarité (par exemple la distance euclidienne). Il s'agit d'un processus important dans la reconnaissance des formes et l'apprentissage non-supervisé. En effet, le clustering des données est un processus central de l'intelligence artificielle (IA). Les algorithmes de clustering sont utilisés dans de nombreuses applications, comme la segmentation d'images ; la quantification d'images vectorielles et en couleur, la fouille de données, la compression, l'apprentissage automatique, etc. Un cluster est généralement identifié par un centre de cluster (ou centroïde). Le clustering des données est un problème difficile dans la reconnaissance de formes non supervisée car les groupes de données peuvent avoir des formes et des tailles différentes [80].

1.5.1 Clustering hiérarchique

Les algorithmes de cette catégorie génèrent un arbre de cluster(ou dendrogramme) en utilisant des techniques heuristiques de division ou de fusion. Un arbre de cluster est défini comme "un arbre montrant une séquence de clustering, chaque clustering étant une

partition de l'ensemble de données" [68]. Les stratégies de clustering hiérarchique se divisent généralement en deux types [98] :

✓ Agglomération : il s'agit d'une approche "ascendante" du clustering hiérarchique où chaque observation commence par un seul cluster, puis des paires de clusters sont fusionnées en remontant dans la hiérarchie. Par conséquent, l'algorithme se traduit par une séquence de regroupement, puis il appartient à l'utilisateur de choisir un regroupement naturel à partir de cette séquence.

✓ Diviseur : Il s'agit d'une approche "descendante" du clustering hiérarchique où toutes les observations commencent dans un cluster, et les divisions sont effectuées de manière récursive à mesure que l'on descend dans la hiérarchie. Ici, une mesure de dissimilarité est nécessaire entre les ensembles d'observations pour décider quels clusters doivent être combinés pour un regroupement aggloméré, ou où un cluster doit se diviser pour un regroupement diviseur. Principalement dans le regroupement hiérarchique, cette mesure est obtenue en utilisant une métrique appropriée et un critère de liaison qui spécifie la dissimilarité des ensembles en fonction des distances par paire d'observations dans les ensembles.

1.5.2 Clustering par K-means

L'algorithme le plus utilisé est l'approche itérative K-means [39]. Le clustering k-means est défini comme une méthode d'analyse de cluster qui vise à partitionner n observations différentes en k clusters différents dans lesquels chaque observation appartient au cluster avec la moyenne la plus proche. Ainsi, l'algorithme K-means minimise la distance intra-groupe. L'algorithme K-means commence par K centroïdes (les valeurs initiales des centroïdes sont choisies au hasard ou dérivées d'informations a priori). Ensuite, chaque motif de l'ensemble de données est attribué au cluster le plus proche (c'est-à-dire le centroïde le plus proche). Enfin, les centroïdes sont recalculés en fonction des modèles associés.

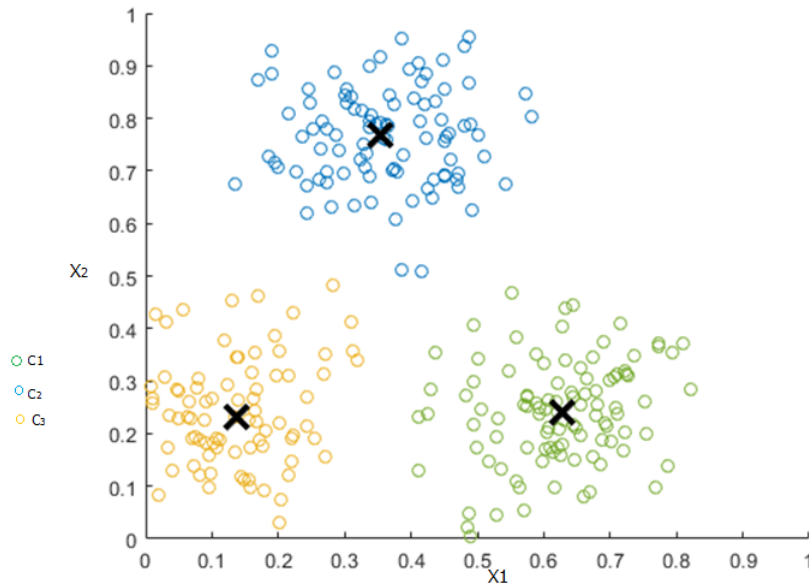


FIGURE 1.4 – Clustering par K-means

1.5.2.1 L'algorithme C-means flou

Une version floue de K-means, appelée C-means flou (FCM) (parfois appelée flou K-means), a été proposée par Bezdek [12] [13]. La FCM est basée sur une extension floue le critère d'erreur du moindre carré. L'avantage de FCM par rapport à K-means est que FCM attribue chaque exemple à chaque cluster avec un certain degré d'appartenance (c'est-à-dire un regroupement flou). Cette méthode est plus adaptée aux applications réelles lorsqu'il y a des chevauchements entre les clusters dans l'ensemble de données

1.6 Conclusion

Dans ce chapitre, nous avons présenté le concept de la reconnaissance des formes, les techniques d'apprentissage, les différents types de classification, et quelques algorithmes de classification et de clustering couramment utilisés dans la littérature.

Dans le chapitre suivant, nous présentons les notions et les connaissances prélimi-

naires sur la théorie des ensembles flous, la structure interne du système d'inférence flou, et la classification à base des règles floue.

La classification à base des règles floues

Sommaire

2.1 Introduction	20
2.2 Concepts fondamentaux du système logique flou	20
2.3 Structure générale des systèmes flous	31
2.4 Le système d'inférence de Mamdani	36
2.5 Le système d'inférence de Sugeno	37
2.6 Classification à base des règles floues	39
2.7 Conclusion	43

Ce chapitre vise à présenter les principaux concepts et notions mathématiques de la théorie des ensembles flous (également appelés logiques floues ou théorie de la logique floue) qui sont nécessaires à la compréhension de ce travail. On commence d'abord par une introduction à la logique floue suivie par une description des concepts fondamentaux des ensembles flous (opérateurs, fonction d'appartenance,...). Ensuite, nous présentons la structure générale des systèmes flous, après nous présentons les systèmes d'inférences de Mamdani et de sugeno. La dernière partie présente la classification floue et présente un

aperçu non exhaustif des principaux domaines d'application de la classification floue.

2.1 Introduction

La logique floue est une extension de la logique classique qui permet la modélisation des imperfections des données, elle est basée sur le raisonnement intuitif en prenant en compte la subjectivité et l'imprécision humaine. Le système de logique floue est l'une des techniques d'apprentissage automatique les plus populaires qui a été appliquée avec succès à la classification, à cause de son style de raisonnement similaire à celui de l'être humain. L'un des principaux avantages de la logique floue est qu'elle permet de déployer dans la structure algorithmique une connaissance qualitative du domaine sur une tâche de classification. Il est bien connu que le concept d'ensemble flou est d'abord issu de l'étude des problèmes liés à la classification [111]. Cela n'est pas surprenant, car le processus de reconnaissance des formes, qui est un aspect important de la perception humaine, est un processus flou. Le flou peut inclure les changements dans l'orientation et la taille de l'objet, le degré d'incomplétude et de distorsion, la quantité de bruit de fond, les descriptions vagues, les mesures imprécises, les informations contradictoires ou ambiguës, les événements aléatoires et autres.

La classification englobe un large éventail de problèmes de traitement de l'information qui sont d'une grande importance pratique, allant de la classification des caractères manuscrits à la détection des fautes dans les machines et au diagnostic médical.

2.2 Concepts fondamentaux du système logique flou

2.2.1 Théorie des ensembles flous

La théorie de la logique floue est basée sur les ensembles flous qui sont une extension naturelle de la théorie classique des ensembles. Un ensemble net est défini par une fonction de vérité bivalente qui n'accepte que les valeurs 0 et 1, ce qui signifie qu'un élément

appartient entièrement à un ensemble ou pas du tout, tandis qu'un ensemble flou est déterminé par une fonction d'appartenance qui accepte toutes les valeurs intermédiaires entre 0 et 1. Les valeurs d'une fonction d'appartenance, appelées degrés ou grades d'appartenance, spécifient précisément dans quelle mesure un élément appartient à un ensemble flou, c'est-à-dire au concept qu'il représente [112]. La théorie des ensembles flous est une méthodologie pour montrer comment aborder l'incertitude, et de traiter des informations imprécises dans une situation complexe. Les ensembles flous sont l'élément central de la logique floue. Ils se caractérisent par des fonctions d'appartenance qui sont associées à des termes ou des mots utilisés dans l'antécédent et les conséquences des règles, et avec entrée et sortie vers le système de logique floue [111].

Un ensemble net A dans un univers de discours U peut être défini comme

$$A \rightarrow \mu_A(x) = \begin{cases} 1, & \text{si } x \in A \\ 0, & \text{si } x \notin A \end{cases}$$

Des ensembles flous peuvent être utilisés pour définir des expressions imprécises et ambiguës telles que "arbre haut" ou "voiture rapide". Les ensembles flous sont définis dans ce qu'on appelle l'univers du discours, notée par U , qui est la gamme des éléments possibles de l'ensemble flou.

Un ensemble flou B est une généralisation d'un ensemble net. Il est défini sur un univers du discours U et est caractérisé par une fonction d'appartenance $\mu_B(x)$ qui prend des valeurs dans l'intervalle $[0,1]$. Une fonction d'appartenance fournit une mesure du degré de similarité d'un élément en U avec l'ensemble flou [111].

$$B = \{(x, \mu_B(x)) | x \in U\}$$

Dans lequel

$$\mu_B(x) : U \rightarrow [0, 1]$$

est la fonction d'appartenance à un ensemble flou B . Cette fonction attribue à chaque élément $x \in U$ son degré d'appartenance à l'ensemble flou B , et on peut distinguer 3 cas :

1) $\mu_B(x) = 1$ signifie l'appartenance complète de l'élément x à l'ensemble flou B , c'est-à-dire $x \in B$,

2) $\mu_B(x) = 0$ signifie l'absence d'appartenance de l'élément x à l'ensemble flou B , c'est-à-dire $x \notin B$,

3) $0 < \mu_B(x) < 1$ signifie une appartenance partielle de l'élément x à l'ensemble flou B .

Par exemple : si $\mu_B(x) = 0.30$; x appartient à l'ensemble flou B avec un degré d'appartenance de 30

2.2.2 Opérateurs flous

Les opérations de complément, d'intersection et d'union de la théorie des ensembles classiques peuvent également être généralisées pour les ensembles flous, ils sont définis en fonction de leurs fonction d'appartenance. Pour ces opérations, il existe plusieurs définitions ayant des implications différentes. Cette section ne présente que les opérateurs les plus courants de la proposition initiale de Zadeh [113].

2.2.2.1 L'opérateur Négation (Complément)

Le complément d'un ensemble flou est de 1 moins les degrés d'appartenance des éléments de l'univers. Cette définition respecte la notion de forte négation [112].

Definition 1. *Le complément d'un ensemble flou A défini sur un ensemble de référence U est défini comme*

$$\neg A = \mu_{\neg A}(x) = 1 - \mu_A(x), x \in U$$

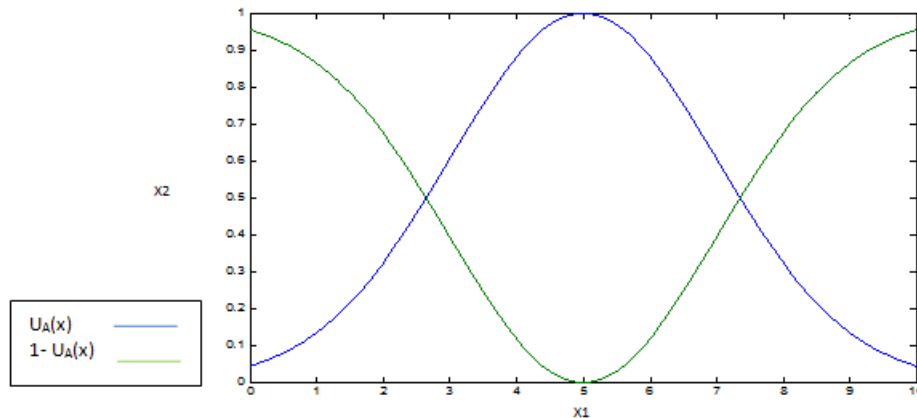


FIGURE 2.1 – Le complément d’une fonction gaussienne

2.2.2.2 L’opérateur ET (intersection)

Pour l’intersection, Zadeh propose d’utiliser l’opérateur minimum. Ces opérateurs ont l’avantage d’être facilement compréhensibles et très rapides à calculer. L’intersection de deux ensembles flous est la valeur minimale des degrés d’appartenance des deux ensembles flous pour tous les éléments de l’ensemble de référence. La figure (2.2) illustre un exemple de l’intersection de deux fonctions d’appartenances gaussiennes en utilisant l’opérateur minimum.

Definition 2. *L’intersection de deux ensembles flous A et B définis sur une référence L’ensemble U est défini comme*

$$A \cap B = \mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)), x \in U$$

2.2.2.3 L’opérateur OU (union)

Pour l’union, Zadeh propose d’utiliser l’opérateur maximum. L’union de deux ensembles flous est la valeur maximale des degrés d’appartenance des deux ensembles flous pour tous les éléments de l’ensemble de référence. La figure (2.2) illustre un exemple de l’union de deux fonctions d’appartenances gaussiennes en utilisant l’opérateur maximum.

Definition 3. L'union de deux ensembles flous A et B définis sur un ensemble de référence U est définie comme

$$A \cup B = \mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)), x \in U$$

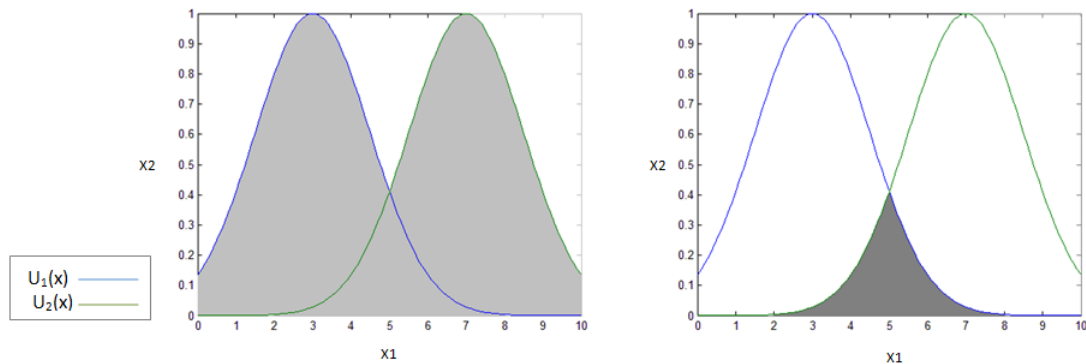


FIGURE 2.2 – Exemple des opérateur ET et OU de deux fonctions d'appartenance gaussienne

2.2.2.4 Norme et conorme triangulaire

Plus généralement, la famille d'opérateurs mettant en œuvre l'intersection est appelée norme triangulaire (abrégée t-norme) et conorme triangulaire (abrégée t-conorme ou s-norme) pour l'union.

L'intersection d'ensembles flous peut être définie plus généralement comme [96]

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x))$$

où la fonction T est la norme dite t-norme.

Donc, $\min(\mu_A(x), \mu_B(x)) = T(\mu_A(x), \mu_B(x))$ est un exemple de fonctionnement de la t-norme.

De même, l'union des ensembles flous est définie comme suit :

$$\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x))$$

où la fonction S est ce que l'on appelle la "t-conorme". Dans ce cas, $\max(\mu_A(x), \mu_B(x)) = S(\mu_A(x), \mu_B(x))$ est un exemple de la forme t-conorme. (W. Siler, J.J. Buckley) Étant donné les ensembles flous A, B, C, \dots tous les sous-ensembles flous de U , nous souhaitons calculer $A \cup B, B \cap C$, et ainsi de suite. Ce que nous utilisons dans la logique floue, ce sont les opérateurs ET et OU généralisés de la logique classique. Ils sont appelés t-normes (pour ET) et t-conormes (pour OU). Nous définissons d'abord les t-normes. Une t-norme T est une fonction de $[0, 1] \times [0, 1]$ à $[0, 1]$. Autrement dit, si $z = T(x, y)$, alors x, y et z appartiennent tous à l'intervalle $[0, 1]$. Toutes les normes t ont les quatre propriétés suivantes [96] :

1. $T(x, 1) = x$ (frontière)
2. $T(x, y) = T(y, x)$ (commutativité)
3. si $y_1 \leq y_2$, alors $T(x, y_1) \leq T(x, y_2)$ (monotonie)
4. $T(x, T(y, z)) = T(T(x, y), z)$ (associativité)

Les équations suivantes sont toutes des exemples de t-normes. Les t-normes de base sont les suivantes

$$T_m(x, y) = \min(x, y)$$

$$T_L(x, y) = \max(0, x + y - 1)$$

$$T_p(x, y) = xy$$

et $T^*(x, y)$ définis comme x si $y = 1$, y si $x = 1$, 0 sinon.

T_m est appelée l'intersection standard ou zadehienne, et est la plus couramment utilisée ; T_L est l'intersection de la différence bornée ; T_p est le produit algébrique ; et T^* est l'intersection radicale.

Les t-Conorms généralisent l'opération OU à partir de la logique classique. Comme pour les t-normes, une t-conorm $S(x, y) = z$ a toujours x, y et z toujours dans $[0, 1]$. Les propriétés de base de tout t-conorm S sont :

1. $S(x, 0) = x$ (limite)
2. $S(x, y) = C(y, x)$ (commutativité)
3. Si $y_1 \leq y_2$, alors $S(x, y_1) \leq S(x, y_2)$ (monotonie)
4. $S(x, S(y, z)) = S(S(x, y), z)$ (associativité)

Les t-conorms de base sont

- 1 : $S_m(x, y) = \max(x, y)$ appelé union standard
- 2 : $S_L(x, y) = \min(1, x + y)$ appelé somme bornée
- 3 : $S_p(x, y) = x + y - xy$ appelé somme algébrique
- 4 : $S^*(x, y)$ appelée union drastique qui se définit comme : x si $y = 0$, y si $x = 0$, et un sinon

2.2.3 Fonctions d'appartenance

La fonction d'appartenance d'un ensemble flou est une généralisation de la fonction de contrôle dans les ensembles classiques. En logique floue, elle représente le degré de vérité comme une extension de l'évaluation. Les degrés de vérité sont souvent confondus avec les probabilités, bien qu'ils soient conceptuellement distincts, parce que la vérité floue représente l'appartenance à des ensembles vaguement définis, et non la probabilité d'un événement ou d'une condition. Les fonctions d'appartenance ont été introduites par Zadeh dans le premier article sur les ensembles flous [113]

Zadeh a donné dans son célèbre article [113] la définition suivante : Un ensemble flou est une classe avec un continuum de degrés d'appartenance. Ainsi, un ensemble flou (classe) B dans un référentiel U est caractérisé par une fonction d'appartenance qui associe à chaque élément x dans U un nombre réel dans l'intervalle $[0, 1]$. La valeur de la fonction d'appartenance à l'élément x représente le " degré d'appartenance " de x dans B . [32]

$$B : U \rightarrow [0, 1]$$

et c'est une sorte de généralisation de la fonction caractéristique traditionnelle d'un sous-ensemble A

$$A : U \rightarrow \{0, 1\}$$

Notez que les ensembles flous sont en fait des sous-ensembles flous de U , comme le souligne Kaufmann [62].

Soit F et B deux ensembles flous, le degré d'appartenance $\mu_F(x_0)$ d'une fonction d'appartenance $\mu_F(x)$ décrit pour l'élément spécial $x = x_0$, à quel degré il appartient à l'ensemble flou F . Cette valeur est dans l'intervalle unitaire $[0,1]$. Bien entendu, x_0 peut simultanément appartenir à un autre ensemble flou B , de sorte que $\mu_B(x_0)$ caractérise le degré d'appartenance de x_0 à B .

2.2.3.1 Notions de base sur les fonctions d'appartenance floue

Support : Le support de l'ensemble flou B est l'ensemble de tous les points $x \in U$ tels que $\mu_B(x) > 0$

Mathématiquement, nous pouvons exprimer le support(B) = $\{(x, \mu_B(x)) | \mu_B(x) > 0\}$

Noyau : Le noyau d'un ensemble flou B est l'ensemble de tous les $x \in U$ tel que

$$\mu B(x) = 1.$$

Mathématiquement, nous pouvons exprimer le noyau $(B) = \{x \in U | \mu B(x) = 1\}$

Croisement : Un point de croisement d'un ensemble flou "B" est un point $x \in U$ auquel $\mu B(x) = 0.5$.

Mathématiquement, nous pouvons exprimer $\text{Croisement}(B) = \{x \in U | \mu B(x) = 0.5\}$

La normalité : Un ensemble flou 'B' est une normale si son noyau est non vide, c'est-à-dire que le noyau $(B) \Rightarrow \phi \neq B$ est un ensemble flou normal.

Mathématiquement, nous pouvons exprimer la Normalité $(B) = 1$ si $\mu B(x) = 1$, pour tout $x \in U$ et $(x, \mu B(x)) \in B$

2.2.3.2 Les formes des fonctions d'appartenance

En principe, toute fonction de la forme $B : U \rightarrow [0, 1]$ décrit une fonction d'appartenance associée à un ensemble flou B qui dépend non seulement du concept à représenter, mais aussi du contexte dans lequel il est utilisé. Les courbes des fonctions peuvent avoir des formes différentes et peuvent avoir des propriétés. Ci-dessous, nous allons préciser ces fonctions et présenter leurs représentations graphiques [96].

1. La fonction d'appartenance gaussienne

La figure 2.3 montre des fonctions d'appartenance gaussienne

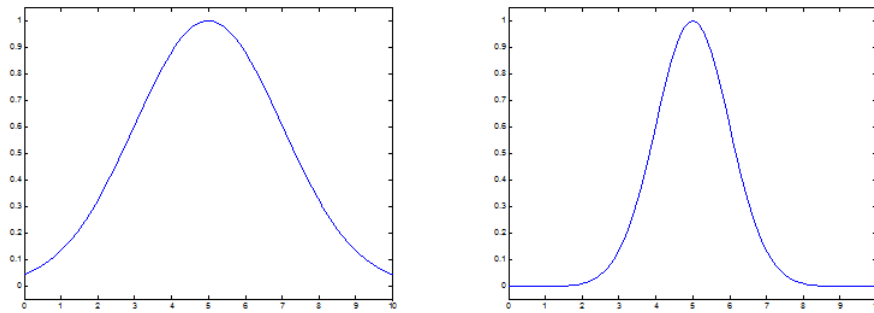


FIGURE 2.3 – Exemple des fonctions d'appartenance gaussienne

Pour un exemple d'entrée x la fonction d'appartenance $\mu(x)$ peut être définie comme suit :

$$\mu_A(x) = \exp\left(-\left(\frac{x-c}{\sigma}\right)^2\right) \quad (2.2.1)$$

Où c est le centre et σ définit la largeur de la courbe gaussienne. C'est la fonction d'appartenance la plus courante.

2. La fonction d'appartenance de la cloche (Bellshape)

La figure 2.4 illustre la forme de la fonction d'appartenance de la cloche

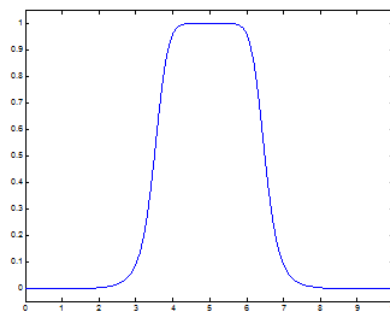


FIGURE 2.4 – Exemple de la fonction d'appartenance de la cloche

La fonction d'appartenance $\mu(x)$ est donnée par

$$\mu(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \quad (2.2.2)$$

Où le paramètre a définit sa largeur, le paramètre b ses pentes, et le paramètre c son centre.

3. La fonction d'appartenance triangulaire

Nous considérons une fonction d'appartenance triangulaire pour la variable d'entrée x , [4] comme le montre la figure 2.5.

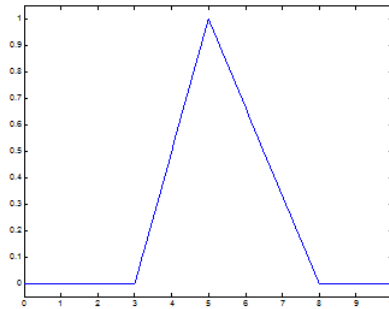


FIGURE 2.5 – Exemple de la fonction d'appartenance triangulaire

À $x = c$, le degré d'appartenance est 1 où c est le centre de la fonction d'appartenance et il diminue à mesure que x s'éloigne de c et atteint 0 à $x = a$ ou $x = b$. Lorsque $x \geq b$ ou $x \leq a$, le degré d'appartenance est 0. Le centre c n'est pas nécessairement le point médian de a et b . La fonction d'appartenance $\mu(x)$ pour la variable d'entrée x peut être définie comme suit :

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{c-a}, & a < x \leq c \\ \frac{b-x}{b-c}, & c < x < b \\ 0, & x \geq b \end{cases}$$

4. Fonctions d'appartenance trapézoïdale

La figure 2.6 montre une fonction d'appartenance trapézoïdale $\mu(x)$.

Définie par sa limite inférieure a et sa limite supérieure d , et les limites inférieure et supérieure de son noyau, b et c respectivement. Le degré d'appartenance est 1 pour

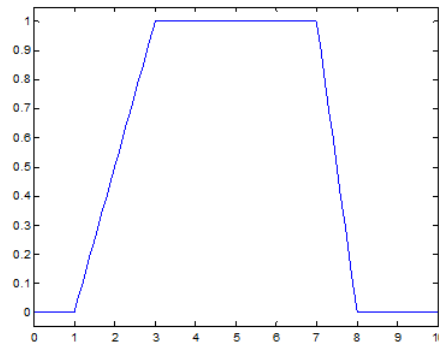


FIGURE 2.6 – Exemple de la fonction d'appartenance trapézoïdale

$b \leq x \leq c$ et le degré d'appartenance diminue jusqu'à ce qu'il atteigne 0 lorsque $x \leq a$ et $x \geq d$. La fonction d'appartenance $\mu(x)$ est donnée par

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x < d \\ 0, & x \geq d \end{cases}$$

$$\mu_{trapezoid} = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}, 0 \right) \right)$$

2.3 Structure générale des systèmes flous

Un système d'inférence floue est composé d'une base de connaissances, contenant des règles Si-Alors floues, et d'un moteur d'inférence, effectuant un raisonnement approximatif [88]. L'un des principaux avantages des systèmes d'inférence [88] est la possibilité de construire des modèles qui imitent le raisonnement humain et qui sont relativement simples et faciles à interpréter. Ces modèles sont peut-être moins précis que les modèles

classiques et plus formels, mais lorsqu'il s'agit d'applications du monde réel, l'interprétabilité, la signification et l'efficacité du calcul peuvent pallier un certain manque de précision.

En raison de sa nature pluridisciplinaire, le système d'inférence flou est connu sous plusieurs noms, tels que système à base de règles floues, système expert flou, modèle flou, mémoire associative floue, contrôleur de logique floue, et simplement système flou [47]

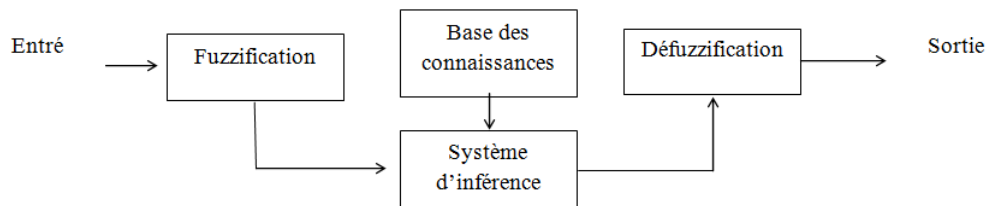


FIGURE 2.7 – Structure générale d'un système basé sur la logique floue

Le système à base de la logique floue est composé de quatre blocs principaux (Fig.2.7).

- ◇ Fuzzification
- ◇ Base de connaissances floues,
- ◇ Moteur d'inférence floue,
- ◇ Défuzzification.

2.3.1 Fuzzification

La fuzzification est le processus de conversion d'une valeur d'entrée numérique en une valeur floue qui est effectuée par l'utilisation des informations de la base de connaissances. Elle permet la détermination de degré d'appartenance d'une donnée d'entrée à chacun des ensembles flous appropriés via les fonctions d'appartenance. Bien que l'on puisse voir différents types de courbes dans la littérature, les fonctions d'appartenance gaussiennes, triangulaires et trapézoïdales sont les plus couramment utilisées dans le processus de fuzzification. Chaque fonction d'appartenance évalue dans quelle mesure la

variable linguistique peut être décrite par un qualificatif flou particulier.

Les valeurs numériques des entrées/sorties sont transformées en ensembles flous. Elles sont donc exprimées dans des variables linguistiques caractérisées par une appartenance appelée sous-ensemble, qui représente chaque point de l'espace d'entrée, appelé l'univers du discours.

✓ Variables linguistiques

Pour définir les règles de la base de règles, l'expert utilisera une "description linguistique" ; par conséquent, des expressions linguistiques sont nécessaires pour les entrées et les sorties et les caractéristiques des entrées et des sorties. Nous utiliserons des "variables linguistiques" (descriptions symboliques constantes de ce qui est en général des quantités variables dans le temps) pour décrire les entrées et les sorties du système flou [70].

Certains ensembles flous discrets décrivent des quantités numériques. Les ensembles flous discrets numériques ont été formalisés sous forme de variables linguistiques, qui consistent en un nom d'ensemble flou discret ; les noms de ses membres, appelés valeurs linguistiques ; et pour chaque valeur linguistique, une fonction d'appartenance comme celles des nombres flous. (W. Siler, J.J. Buckley)

2.3.2 Base de connaissances floues

La base de connaissances se compose d'une base de règles (BR) et d'une base de données (BD). Une base de règles contient un certain nombre de règles floues Si-Alors. Une base de données définit les MF des ensembles flous utilisés dans les règles floues [27]

✓ Base de données :

La BD contient la définition des ensembles flous liés aux termes linguistiques utilisés dans le BR. Ce fait nous amène à spécifier les points suivants :

- le nombre de termes linguistiques pour chaque variable considérée.
- La fonction d'appartenance des ensembles flous liés à ces termes linguistiques.

✓ La base des règles Si-Alors :

Le processus de raisonnement est basé sur une série de règles "Si - Alors" comme une sorte de connaissance d'expert . La déclaration conditionnelle (ou proposition) contient une prémisse, la partie "Si", et une conclusion, la partie "Alors" [30]. La connaissance incluse dans un système de règle flou est constituée d'un groupe de plusieurs règles de la forme

$$\text{"Si } X \text{ est } A \text{ Alors } Y \text{ est } B\text{"}$$

ou, plus généralement,

$$\text{"Si } X_1 \text{ est } A_1 \text{ et ...et } X_n \text{ est } A_n \text{ Alors } Y \text{ est } B\text{"}$$

où A, A_n, B sont des variables linguistiques définies par des sous-ensembles flous.

2.3.3 Moteur d'inférence flou

Le mécanisme d'inférence flou est le processus de raisonnement en logique floue qui détermine les sorties correspondant aux entrées floues. en effectuant les opérations d'inférence sur les règles. La base de règles floue est composée de règles SI-ALORS, chaque règle floue définit une implication floue entre les parties de la règle de condition (Si) et de conclusion(Alors). Il existe plusieurs types d'inférence floue, et les deux principaux types les plus récemment utilisés sont , le type Mamdani flou [71] et Sugeno [103]. La base de connaissances d'un système d'inférence Mamdani contient des règles où les antécédents et les conséquences sont des ensembles flous. Les systèmes d'inférence Sugeno, d'autre part, utilisent des règles avec des antécédents flous et des conséquents nets (ordinaire).Une brève description de chaque type est présentée ci-dessous.

2.3.4 Défuzzification

La dernière étape est le processus de défuzzification qui consiste à obtenir un nombre unique à partir de la sortie de l'ensemble flou agrégé.Cette étape est une transformation

inverse par rapport au processus de fuzzification, s'effectue toujours à l'aide des fonctions d'appartenance. Elle est utilisée pour transférer les résultats d'inférences floues dans une sortie numérique (précise), à partir des degrés d'appartenance. Il existe plusieurs formes de défuzzification, y compris les méthodes du centre de gravité (COG), de la moyenne du maximum (MOM) et de la moyenne du centre.

Méthode du centre de gravité (COG) ou centre de gravité de la zone (COA)

Cette méthode fournit une valeur précise basée sur le centre de gravité de l'ensemble flou.

Pour la fonction d'appartenance discrète, la valeur de défuzzification désignée par x^* en utilisant le COG est définie comme suit [89]

$$x^* = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \quad (2.3.3)$$

Ici x_i indique l'élément échantillon, $\mu(x_i)$ est la fonction d'appartenance, et n représente le nombre d'éléments dans l'échantillon.

Pour la fonction d'appartenance continue, x^* est définie par

$$x^* = \frac{\int x \mu_A(x) dx}{\int \mu_A(x) dx} \quad (2.3.4)$$

Méthode du centre des sommes (COS)

Il s'agit de la technique de défuzzification la plus utilisée. Dans cette méthode, la zone de chevauchement est comptée deux fois.

La valeur défuzzifiée x^* est définie comme :

$$x^* = \frac{\sum_{i=1}^N x_i \cdot \sum_{k=1}^n \mu_{A_k}(x_i)}{\sum_{i=1}^N \sum_{k=1}^n \mu_{A_k}(x_i)} \quad (2.3.5)$$

Ici, n est le nombre d'ensembles flous, N est le nombre de variables floues $\mu_{(A_k)}(x_i)$ est la fonction d'appartenance pour le $k - i\grave{e}me$ ensemble flou.

Méthode de la moyenne des maxima (MOM)

Dans cette méthode, la valeur défuzzifiée est prise comme l'élément ayant les valeurs d'appartenance les plus élevées. Lorsqu'il y a plus d'un élément ayant des valeurs d'appartenance maximales, la valeur moyenne des maxima est prise. Soit A un ensemble flou avec la fonction d'appartenance $\mu_A(x)$ définie sur $x \in X$, où X est un univers de discours. La valeur défuzzifiée est disons x^* d'un ensemble flou et est définie comme suit [89]

$$x^* = \frac{\sum_{x_i \in M} x_i}{|M|} \quad (2.3.6)$$

2.4 Le système d'inférence de Mamdani

La méthode Mamdani est considérée comme la technique de système d'inférence flou la plus généralement utilisée. La méthode a été introduite pour la première fois par le professeur Ebrahim Mamdani en 1975 [71], lorsqu'il a appliqué un ensemble de règles floues fournies par des opérateurs des experts humains pour construire le premier système flou pour contrôler une combinaison de moteur à vapeur et de chaudière. Une règle typique du modèle flou de type Mamadi avec deux entrées et une sortie unique se présente sous la forme

Si x est A et y est B , Alors z est C

où x, y et z sont des variables linguistiques ; A, B et C sont des ensembles flous.

Les résultats de la méthode de système d'inférence flou de Mamdani peuvent être facilement transformés en une forme linguistique comme résultat d'inférence avant la défuzzification car le résultat de l'inférence est un ensemble flou . Le processus de la méthode Mamdani se déroule en six étapes [50] :

Tout d'abord, (1) la détermination des règles floues nécessaires.

Ensuite, (2) il y a la conversion des entrées nettes en entrées floues en utilisant la fonction d'appartenance.

Ensuite, (3) les entrées floues doivent être rassemblées selon les règles floues.

Ensuite, (4) le conséquent de la règle est déterminé en joignant le poids de la règle et (4) la sortie de la fonction d'appartenance.

Ensuite, (5) toutes les conséquences doivent être combinées pour obtenir la distribution de la sortie.

Enfin, (6) la distribution de la sortie est défuzzifiée. La figure 2.8 illustre le système d'inférence floue de Mamdani.

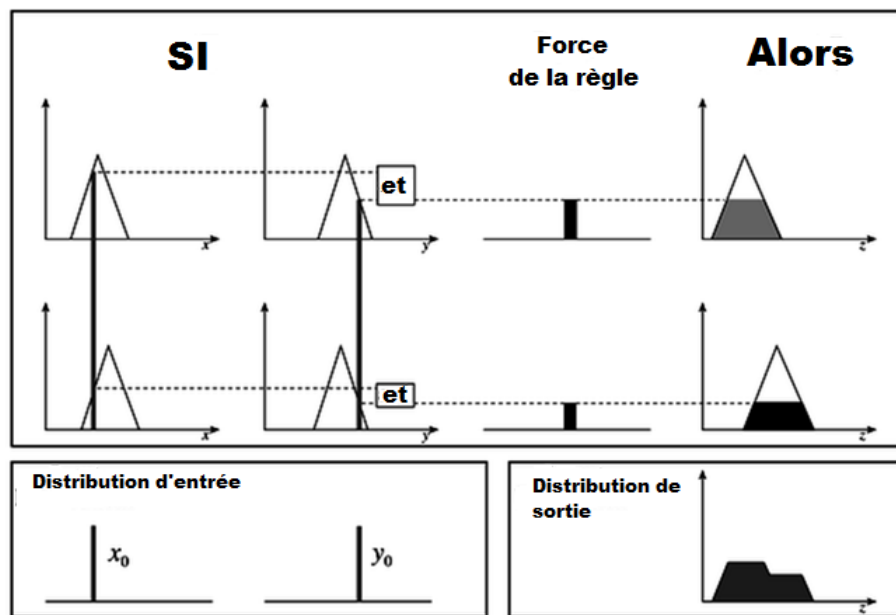


FIGURE 2.8 – Exemple d'un système d'inférence de Mamdani

2.5 Le système d'inférence de Sugeno

Cette méthode floue a été présentée en 1985 par Takagi, Sugeno et Kang [103]. Elle est très similaire à la méthode Mamdani, sauf que la conséquence de la règle est modifiée, où une fonction mathématique est utilisée pour la variable de sortie au lieu de l'ensemble flou [15]. Le format général de la règle pour la méthode floue de Sugeno est le suivant :

Si x est A et y est B Alors z est $f(x, y)$

où x, y et z sont des variables linguistiques ; A et B sont des ensembles flous ; et $f(x, y)$ est la fonction mathématique. La figure 2.9 illustre le système d'inférence floue de Sugeno. La méthode Sugeno est efficace sur le plan des calculs et fonctionne bien avec les techniques d'optimisation et d'adaptation, ce qui la rend très intéressante dans les problèmes de contrôle, en particulier pour les systèmes dynamiques non linéaires.

Habituellement, $f(x, y)$ est un polynôme dans les variables d'entrée x et y mais il peut s'agir de n'importe quelle fonction tant qu'elle peut décrire de manière appropriée la sortie du modèle dans la région floue spécifiée par l'antécédent de la règle. Lorsque $f(x, y)$ est un polynôme du premier ordre, le système d'inférence floue qui en résulte est appelé modèle flou Sugeno du premier ordre qui a été proposé dans [103]. Lorsque $f(x, y)$ est une constante, il s'agit du modèle flou d'ordre zéro de Sugeno, qui peut être considéré comme un cas particulier du modèle flou de Mamdani, dans lequel la conséquence de chaque règle est spécifiée par un singleton flou ou par une conséquence pré-défuzzifiée

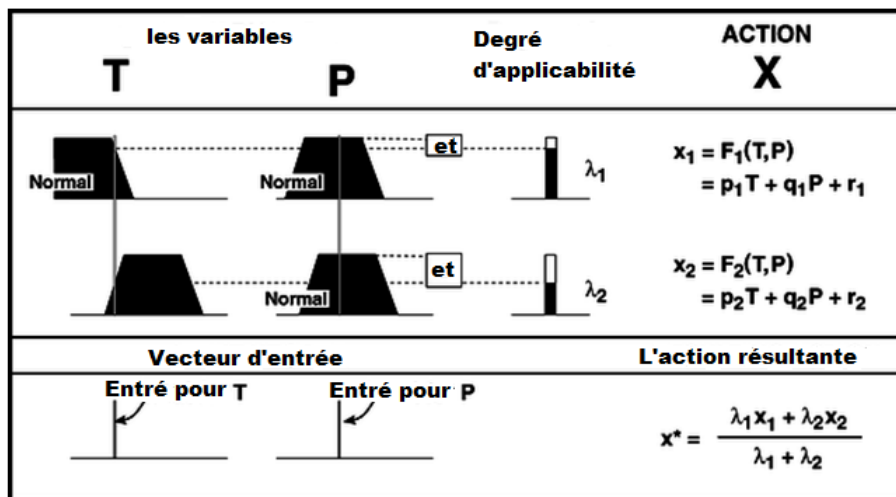


FIGURE 2.9 – Exemple d'un système d'inférence Sugeno

2.6 Classification à base des règles floues

2.6.1 Principe de classification floue

La classification floue est une extension naturelle de la classification traditionnelle, de la même manière que les ensembles flous étendent les ensembles classiques. Dans une classification précise, chaque objet est affecté à une classe exacte, ce qui signifie que le degré d'appartenance de l'objet est de 1 dans cette classe et de 0 dans toutes les autres. L'appartenance des objets dans les classes est donc mutuellement exclusive. En revanche, une classification floue permet aux objets d'appartenir à plusieurs classes en même temps ; De plus, chaque objet a des degrés d'appartenance qui expriment dans quelle mesure cet objet appartient aux différentes classes [112].

Dans les classificateurs flous conventionnels, l'espace de chaque exemple d'entrée est divisée en plusieurs intervalles. Ensuite, chaque intervalle est considéré comme un ensemble flou et une fonction d'appartenance associée est définie. Ainsi, l'espace d'entrée est divisé en plusieurs sous-régions qui sont parallèles aux axes d'entrée. Pour chaque sous-région, une règle floue est définie ; si l'entrée se trouve dans la sous-région, l'entrée appartient à la classe associée à la sous-région. Ensuite, les degrés d'appartenance d'une entrée inconnue pour tous les ensembles flous sont calculés et l'entrée est classée dans la classe avec le degré d'appartenance maximum. Par conséquent, les fonctions d'appartenance influencent directement les performances du classificateur flou . Lorsque le système de classification est composé d'un ensemble de règles floues, il est appelé système de classification basé sur des règles floues, ou Fuzzy rule based classification system (FRBCS) en anglais [4].

En peut générer des base de règles d'un système de classification avec l'un des trois types de règles suivants [56] :

A. Les règles floues avec une classe dans le conséquent. Ce type de règle a la structure suivante :

$$R_q : \text{Si } x_1 \text{ est } A_{q1} \text{ et } \dots \text{ et } x_n \text{ est } A_{qn} \text{ Alors } C_q$$

B. Des règles floues avec une classe et un degré de certitude dans le conséquent

$$R_q : \text{Si } x_1 \text{ est } A_{q1} \text{ et } \dots \text{ et } x_n \text{ est } A_{qn} \text{ Alors } C_q \text{ avec } CF_q$$

Où $x = (x_1 \dots x_n)$ est un vecteur de caractéristique à n-dimensions, A_{qi} est un ensemble flou antécédent pour le $i^{\text{ème}}$ attribut, C_q est une classe conséquente, et CF_q est un degré de certitude (poids de la règle)

C. Le troisième type avec plusieurs classes conséquentes est écrit pour les problèmes de classification de M-classes comme suit :

$$R_q : \text{Si } x_1 \text{ est } A_{q1} \text{ et } \dots \text{ et } x_n \text{ est } A_{qn} \text{ Alors classe1 avec } CF1 \text{ et } \dots \text{ classeM avec } CF_{qM}$$

Où CF_{qM} est le degré de certitude de la classe M .

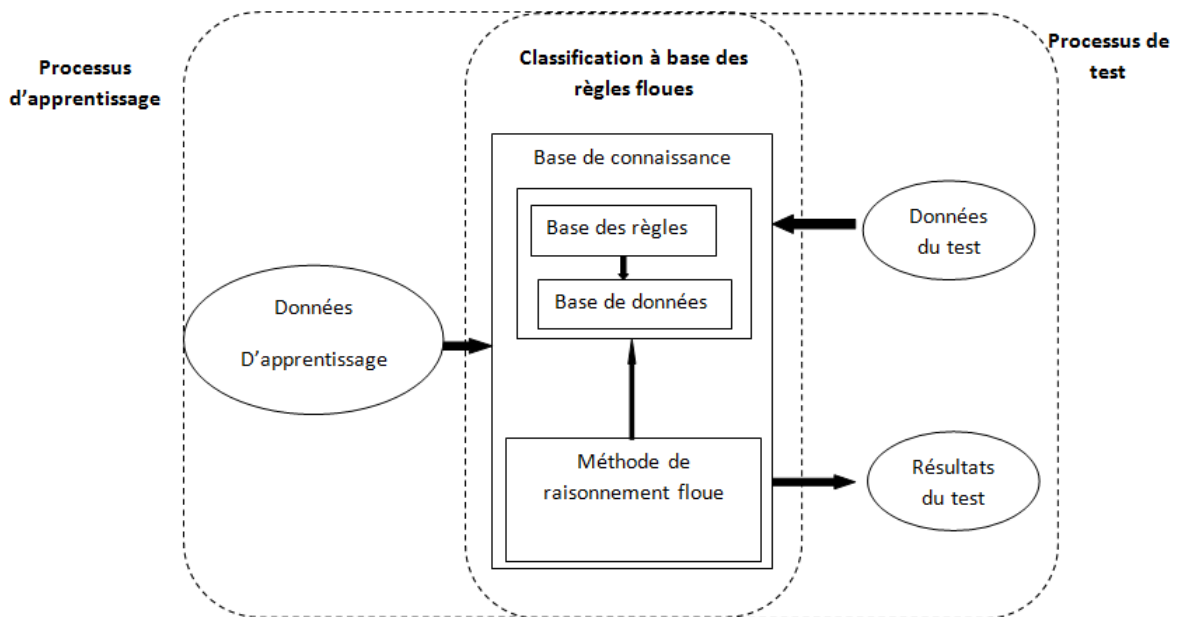


FIGURE 2.10 – Le processus de classification à base des règles floues

2.6.2 Méthodes de raisonnement (décision)

Il existe plusieurs méthodes de raisonnement floues pour classer de nouveaux exemples à l'aide de règles floues. Dans ce travail, nous nous intéressons aux deux méthodes les plus utilisées. La première méthode est basée sur un seul gagnant, où une seule règle gagnante est utilisée pour classer les exemples présentés. La deuxième méthode, quant à elle, se base sur le vote où la classification est effectuée par une procédure de vote selon toutes les règles. Pour plus de détail sur les méthodes de raisonnement flou pour les problèmes de classification, le lecteur peut consulter Cordon et al [28].

Méthode basée sur un seul gagnant

Soit S un ensemble de règles floues de la forme (B) de la section (2.6.1). Ce système peut être utilisé comme un système de classification à base de règles floues. Dans ce système, la règle gagnante, R_g , pour un nouvel exemple $x_p = (x_{p1}, \dots, x_{pn})$, est celle qui satisfait :

$$\mu_g(x_p).CF_g = \max\{\mu_q(x_{pn}).CF_q | R_q \in S\}$$

Autrement dit, la règle gagnante a le produit maximal du degré de compatibilité et du poids de la règle. Si plusieurs règles linguistiques ont le même produit maximal mais des classes conséquentes différentes pour le nouvel exemple x_p , la classification de x_p est rejetée. La classification est également rejetée si aucune règle linguistique n'est compatible avec le nouvel exemple x_p . Lorsque nous utilisons la méthode du gagnant unique, chaque règle linguistique a sa région de décision. En d'autres termes, chaque règle linguistique est responsable de la classification des nouveaux exemples dans sa région de décision. Cela conduit à une grande transparence du processus de classification des nouveaux exemples [55].

Comme nous savons qu'une seule règle est responsable de la classification d'un nouvel exemple, nous pouvons expliquer pourquoi cet exemple est assigné à une classe particulière par notre système de règles. C'est un avantage de la méthode basée sur un seul

gagnant par rapport aux méthodes de raisonnement flou.

Méthode basée sur le vote

Selon la méthode basée sur le vote ; pour classer un nouvel exemple, chaque règle vote pour sa classe conséquente. Le produit du degré de compatibilité et du poids de la règle est utilisé comme le degré de certitude du vote par cette règle. Lorsqu'un nouvel exemple x_p est classé par un ensemble de règles selon la méthode du vote, le vote total pour chaque classe est calculé comme suit :

$$V_{Classh} = \sum_{\substack{R_q \in S, \\ C_q = h}} \mu_{A_q}(x_p) \cdot CF_q$$

Dans le cas de la méthode du seul gagnant, une seule règle est identifiée pour la classification d'un nouvel exemple, tandis que dans la méthode du vote toutes les règles compatibles sont responsables et avec différents degrés de responsabilité [55]. Cela rend difficile d'expliquer comment un nouvel exemple est classé avec l'ensemble de règles.

2.6.3 Domaine d'application de la classification floue

La classification floue a été largement utilisée dans de divers domaines. Ceci est dû à la capacité des système flous a traité les imprécisions dans les problèmes réels. Par exemple dans le domaine médical, ou le diagnostic est souvent considéré comme un problème de classification, les données actuellement collectés sont très massive et présentent beaucoup d'incertitude et d'imprécisions. L'application des techniques de logique floue offre ainsi de précieux outils d'une part, pour le traitement des données, et d'autre part, pour modéliser les connaissances acquises des experts. Le tableau (2.1) montre les différents domaines d'applications de la classification floue.

Domaine	Applications	Références
Médicale	-Cancer du sein -Diabet - Cardiotocographie - Cancer de la thyroïde	[85], [78], [45], [76].
Traitement d'image	-Reconnaissance d'écriture -Segmentation -Reconnaissance des empreintes digitales, des visages -La segmentation des images IRM -La segmentation des images satellitaires -Localisation d'objets dans des images	[61], [100], [48], [58].
Commerce en ligne	-La catégorisation des documents -Le marketing -la gestion de la clientèle -Base de données clients en ligne	[104], [17], [10].

TABLE 2.1 – Quelques domaines d'applications de la classification floue

2.7 Conclusion

Dans ce chapitre, les concepts de la théorie des ensembles flous nécessaires pour comprendre le travail présenté dans cette thèse ont été introduits. Le concept de sections 2.1 est le concept de base de la logique floue et des systèmes d'inférences. Des représentations des fonctions d'appartenance les plus utilisées sont données, ainsi que les opérateurs logiques. Nous nous sommes intéressés aux concepts de base de la classification à base des règles floues, et nous avons présenté quelque méthode de raisonnement pour classer un nouvel élément.

Comme nous avons vu dans les sections du chapitre la construction d'un système flou

passer par plusieurs étapes, chacune peut influencer la performance du système. Pour cela plusieurs méthodes dans la littérature ont été consacrées, des méthodes d'optimisation des fonctions d'appartenance, le poids de la règle, le nombre des règles floues et d'autres feront l'objet du chapitre suivant.

**Optimisation des règles floue par les méthodes
bio-inspirées.**

Sommaire

3.1 Introduction	45
3.2 Des algorithmes bio-inspirés pour la classification basée sur des règles floues (FRBCSs)	46
3.3 Conclusion	53

3.1 Introduction

De nombreux problèmes peuvent être exprimés comme des problèmes de reconnaissance des formes. C'est pourquoi le développement de classificateurs efficaces est devenu un domaine de recherche important. Diverses techniques ont été proposées pour produire des classificateurs [95] Dans le domaine de la classification, l'objectif est d'assigner l'exemple ou l'échantillon à la classe associée, sur un ensemble de classes prédéfinies, en fonction des valeurs de certains caractéristiques des exemples. Les problèmes de

classification peuvent être traités en utilisant différentes approches. Les approches floues sont devenues l'une des solutions bien connues pour les problèmes de classification. Dans la classification floue, un échantillon peut être associé à de nombreuses classes différentes avec des degrés différents. Dans les problèmes de classification, de nombreuses approches ont été proposées pour la génération et l'apprentissage des règles floues SI-Alors à partir de données numériques [54]. L'une de ces dernières est l'approche bio-inspirée comme l'algorithme génétique, le système immunitaire, l'optimisation par colonies de fourmis, l'optimisation par l'essaim de particules.....etc.

3.2 Des algorithmes bio-inspirés pour la classification basée sur des règles floues (FRBCSs)

3.2.1 Algorithme génétique pour les FRBCS

3.2.1.1 Brève description de l'algorithme génétique

Les algorithmes génétiques GA sont des algorithmes d'optimisation stochastique basé sur l'évolution avec un potentiel de recherche global proposé par Holland en 1975. L'algorithme commence par l'initialisation d'une population de solution (chromosome). De nouvelles solutions (progéniture/enfants) sont formées en sélectionnant des solutions en fonction de leur fitness (les meilleurs). Les trois principaux opérateurs génétiques en GA sont : la sélection, le croisement et la mutation [14], [75]. Les étapes de l'algorithme génétique sont les suivantes [75] :

1. [Début] Générer une population aléatoire de n chromosomes (solutions possibles pour le problème)
2. [Fitness] Évaluez la fitness $f(x)$ de chaque chromosome x dans la population
3. [Nouvelle population] Créez une nouvelle population en répétant les étapes suivantes jusqu'à ce que la nouvelle population soit complète.

- a. [Sélection] Sélectionner les deux chromosomes parents d'une population en fonction de leur forme physique (fitness) (plus ils sont en bonne forme physique, plus ils ont de chances d'être sélectionnés).
- b. [Croisement] Avec une probabilité de croisement, croiser les parents pour former une nouvelle progéniture (génération) (enfants). Si aucun croisement n'a été effectué, la progéniture est une copie exacte des parents.
- c. [Mutation] Avec une probabilité de mutation, muter les nouveaux progénitures à chaque locus (position dans le chromosome).
- d. [Acceptation] Placer une nouvelle progéniture dans une nouvelle population.

3.2.1.2 Application de l'algorithme génétique aux FRBCS

De nombreux travaux dans la littérature ont utilisé des algorithmes génétiques pour générer des règles floues Si-Alors et régler (ajuster) les fonctions d'appartenance des ensembles flous, antécédents et conséquents [51]. Dans [110] ils ont développé un algorithme génétique flou pour générer un ensemble de règles de classification floue, la performance a été meilleure en termes de précision prédictive de l'ensemble de règles découvert. Pour la phase d'apprentissage, il existe plusieurs applications avec des algorithmes évolutifs comme dans [27] O.Cordon et al. ont décrit un processus d'apprentissage génétique en multi-étape qui intègre des méthodes de raisonnement flou (FRMs) en coopération avec les bases de connaissances (KBs) ce processus apprend le meilleur ensemble des protections linguistiques pour des termes linguistiques variables, en conservant ainsi les fonctions descriptives du système de classification. D'autres recherches [11] les auteurs ont également utilisé la programmation génétique pour l'apprentissage des FRBCS avec une grande capacité d'interprétation. . Dans la littérature, plusieurs algorithmes ont été proposés pour la sélection des exemples et des caractéristiques : J.J.Aguilera et al. [5] ont analysé différents algorithmes génétiques de niching pour la sélection des caractéristiques, l'ensemble des solution optimal qu'il fournissent sont regroupées dans un multi-

classificateur. Dans ce travail, ils prouvent que l'AG avec compensation est l'AG de niching qui présente une meilleure performance en précision, réduction et convergence . Dans [102] D.G.Stavrakoudis et al. ont proposé l'utilisation d'un schéma de sélection des caractéristiques locales pour la sélection efficace des caractéristiques pertinentes pendant la conception des FRBCSs génétiques selon l'approche IRL(Iterative Rule Learning). Pour la sélection des exemples : B.Giglio et al. [46] analysé l'influence des méthodes de sélection des exemples pour la sélection des ensembles d'apprentissage avec système génétique flou pour la classification et ils utilisent la technique TSS (Training Set Selection) dans le but de réduire le nombre de règles tout en conservant la plus grande précision. De plus, les auteurs dans [38] ont également utilisé la technique des TSS comme technique de sélection avant d'appliquer le GA pour la génération des FRBCS afin de réduire la complexité des FRBCS générés et diminuer le nombre des exemples dans les ensembles de données en préservant ou augmentant leur précision. Pour faire face au problème d'un degré d'incertitude dans la définition des partitions floues J.Sanz et al. ont utilisé le concept IVFS (Interval-Valued-Fuzzy-Set) pour caractériser les étiquettes linguistiques qui composent les attributs du problème. ils améliorent les performances des FRBCS grâce à la coopération entre l'IVFS et à l'étape de réglage génétique post-traitement au cours de laquelle l'amplitude du support des IVFS est réglée, ce qui permet d'améliorer la précision de la classification [93]. Dans [94], ils ont appliqué cette méthodologie à la classification des patients en fonction de leur risque de souffrir de maladies cardiovasculaires. Dans [83], les auteurs ont proposé un algorithme génétique intégré à la logique floue pour la génération de règles de classification floue (FCR) à partir d'ensembles de données contenant un attribut mixte (attribut catégorique et continu). Un système co-évolutif est proposé par R.R.F. Mendes et al. dans [74] pour le développement des FCR, ce système utilise deux algorithmes évolutifs : Un GP pour l'évolution d'une population de d'ensemble de règles floues, une population d'appartenance simple et évolutive. Basé sur l'algorithme génétique et la formulation de programmation entière J. Sanz et al. ont proposé une approche heuristique hybride (hGA) qui fournit un mécanisme plus précis et

plus robuste pour les problèmes linguistiques (ils ont utilisé GA pour générer différentes règles pour chaque classe et IPF pour sélectionner une règle parmi celles obtenues par GA) [9]. K. Trawinski et al. ont proposé un système de classification basé sur des règles linguistiques floues jouant le rôle de la méthode de combinaison de système de multi-classificateur (un FRBCS-CM) il a été construit sous une AG spécifique qui est utilisée avec la possibilité de sélectionner des caractéristiques et des termes linguistiques dans les parties antécédentes des règles, effectuée à la fois la fusion du classificateur et la sélection du classificateur au niveau de la classe [106].

3.2.2 Système immunitaire artificiel pour FRBCSs

3.2.2.1 Brève description de Système immunitaire artificiel

Proposé par Dasgupta, en 1999, AIS consiste en des méthodes qui s'inspirent du système immunitaire biologique et conçues pour résoudre des problèmes du monde réel, Les principales caractéristiques du système immunitaire sont l'apprentissage, l'adaptabilité et les mécanismes de la mémoire, qui en font une source d'inspiration riche [60]. Les types standard d'AIS sont : la sélection négative, la sélection clonale et les réseaux immunitaires. Les étapes du Système immunitaire artificiel par la sélection clonale sont les suivantes [44] :

1. Initialisation : créer une population initiale aléatoire d'anticorps, P_0 . Répétez les étapes 2 à 7 si une condition de terminaison prédéfinie n'est pas remplie.
2. Évaluation et sélection 1 : Sélectionner un sous-ensemble, F , des anticorps les plus adaptés de P_t en fonction d'une fonction d'aptitude ($f(ab_i)$)
3. Clonage : Pour chaque $ab \in F$, créer un ensemble de clones, C_i de sorte que $|C_i| = nc(ab_i)$. L'ensemble de tous les clones, $C = \bigcup_i C_i$
4. Mutation : Muter chaque clone $c \in C$ par une fonction $am(f(ab_i), \rho)$ Ajouter le clones, C' , à P_t pour donner P'_t
5. Évaluation et sélection 2 : sélectionner un sous-ensemble, F' des anticorps les plus adaptés de P'_t

6. La diversité : Ajouter r nouvelles cellules-B générées de manière aléatoire à F' donne une nouvelle population P_t''
7. La mort : Ne retenez que les meilleurs membres $|P_t|$ de P_t'' pour donner P_{t+1} ; toutes les autres cellules-B sont considérées comme mortes.

3.2.2.2 Application de Système immunitaire artificiel pour FRBCSs

R.T. Alves et al [7] ont proposé un algorithme utilisant le Système Immunitaire Artificiel (SIA) pour générer des FRBCS de la forme "Si-Alors" basée sur le principe de sélection clonale appelé (IFRAIS). Cet algorithme est composé de deux parties : une procédure séquentielle de recouvrement et une procédure d'évolution des règles. Ils représentent chaque règle de classification par un anticorps. Pour classer les nouveaux exemples (antigènes), ils tiennent compte de l'adéquation d'une règle floue et de l'affinité entre la règle et la nouvelle instance. Dans [67] Z.Lei et al. ont proposé un algorithme pour concevoir un système de classification floue représenté sous la forme "Si-Alors" basé sur le principe du système immunitaire (FAIS). Ils trouvent un bon compromis entre le nombre de règles floues "Si-Alors" et la précision de la classification. Cet algorithme a également inclus une étape pour trouver la redondance entre les ensembles flous dans le système de simplification et de réduction base de règles. E. Kamaloo et al [60] ont proposé une méthode (FAIS) similaire à [67]. Ils ont combiné la logique floue et les concepts AIS, la logique floue est appliquée pour extraire les règles floues. Cette approche permet de prédire la probabilité que les clients obtiennent une cote de crédit.

3.2.3 L'optimisation par colonies de fourmis (ACO) pour FRBCSs

3.2.3.1 Brève description de L'optimisation par colonies de fourmis

L'optimisation des colonies de fourmis (ACO) est une technique probabiliste proposée par Dorigo & Di Caro en 1999 [34] qui peut être utilisée pour résoudre des problèmes

qui peuvent être réduits à trouver de bons chemins le long des graphiques. Cette méthode s'inspire du comportement présenté par les fourmis dans les trajets les plus courts entre le nid des fourmis et les sources de nourriture en traçant des traces de phéromones [14]. Par une décision probabiliste, les fourmis choisissent le chemin à suivre avec un biaisement de la quantité de phéromone, la trace de phéromone la plus forte est suivie. Les étapes de l'algorithme ANT sont les suivantes [26] :

1. [Initialisation] Initialiser la population de fourmis, Le niveau initial de phéromones entre deux statistiques est fixé à une petite constante positive.
2. [Transition] Calculez la probabilité de transition de stat r à stat s pour la k-ème fourmi.
3. [Mis à jours] Mettez à jour le niveau de phéromone entre les statistiques. Outre le renforcement des traces de phéromones, ACO inclut également l'évaporation des traces de phéromones, afin d'aider les fourmis à oublier les mauvaises solutions.
4. Continuer les étapes (2)-(4) jusqu'à ce que le comportement stagne ou que le nombre maximum de cycles soit atteint.

3.2.3.2 Application de L'optimisation par colonies de fourmis pour FRBCS

Dans [1], [2] les auteurs proposent une génétique hybride avec une recherche locale basée sur l'heuristique ACO qui est utilisée dans la structure du système flou évolutif du michigan. Cet algorithme a une grande performance pour les problèmes de classification de détection d'intrusion. Ganji et al. [41] ont proposé un système de classification qui appliquait l'algorithme ACO pour extraire les règles floues Si-Alors pour le diagnostic du diabète, qu'ils ont appelé FADD. Les résultats montrent que le FADD peut classer ou détecter le diabète avec une précision acceptable. Ganji et al [42] ont utilisé une hybridation de l'ACO et de la logique floue pour classer les données médicales sous le nom de Fc-AntMiner. Héritier d'un ACO coopératif pour l'apprentissage des règles floues est utilisé dans l'étape d'apprentissage. Ensuite, dans l'étape de test, ils utilisent un moteur de détection floue pour classer les exemples. Dans [43] Ganji et al. combinent aussi l'ACO

avec la logique floue pour extraire un ensemble de règles floues pour le diagnostic du diabète, ils l'ont appelé FCS-ANTMINER. Les applications de cet algorithme donnent un taux de classification élevé avec un petit nombre de règles floues 'Si-Alors' de courte longueur.

3.2.4 L'optimisation par essaim de particules (PSO) pour FRBCS

3.2.4.1 Brève description de L'optimisation par essaim de particules

L'optimisation de l'essaim de particules (PSO) a été introduite pour la première fois par Kennedy et Eberhart en 1995 [63], il s'agit d'une technique d'optimisation stochastique basée sur la population et inspirée du comportement social du flocage d'oiseaux ou du banc de poissons. Chaque individu en essaim de PSO s'appelle particule, cela correspond à la solution potentielle. Chaque particule a une valeur de fitness (pbest) : meilleure position, position actuelle, meilleure position trouvée par son voisinage (gbest) [26], [14]. Les étapes de l'algorithme PSO sont décrites comme suit :

1. Initialisation : Initialiser les particules avec des vecteurs de position et de vitesse aléatoires.
2. Pour la position de chaque particule (p) évaluer la condition physique (fitness).
3. Comparez la valeur de la fitness de chaque particule avec sa valeur pbest. Si la fitness (p) est meilleure que la fitness (pbest) alors pbest = p.
4. Déterminez la particule avec la meilleure valeur de la fitness, Définissez le meilleur des pBests comme gBest .
5. Mettre à jour la vitesse et la position des particules à l'aide d'équations appropriées.
6. Ces étapes récursives 2 à 5 se poursuivent jusqu'au critère d'arrêt rencontré.

3.2.4.2 Application de l'algorithme d'optimisation par essaim de particules pour FRBCSs

Récemment, certains articles utilisant PSO pour des travaux de recherche utilisant un système combiné basé sur PSO et des règles floues pour la classification Chia-Chong Chen [18] a proposé une méthode basée sur l'optimisation par l'essaim de particules pour sélectionner un système de classification floue avec un nombre adéquat de règles floues et minimiser le nombre des exemples mal classés. Cette sélection s'effectue en guidant la procédure de recherche par l'implémentation d'une fonction fitness. Dans [61] les auteurs ont proposé une méthode d'optimisation par l'essaim de particules (PSO) pour générer un nombre adéquat de classificateurs de règles floues. Dans ce travail, ils ont appliqué cet algorithme pour récupérer des images en fonction de leur couleur dominante. Ils ont défini des ensembles flous sur les composantes H, S et L de l'espace couleur HSL. C.Rania [86] a intégré des opérateurs de mutation avec pso pour la conception d'un système de classificateur flou, cette base de règles évoluée et les fonctions d'appartenance ayant pour objectif de minimiser le nombre de règles et le nombre des exemples mal classifiés, ainsi que de maximiser le nombre des exemples correctement classés. Dans [6], ils proposent un PSO pour générer un système basé sur des règles floues si-alors pour la classification du cancer du sein.

3.3 Conclusion

Ce chapitre donne un aperçu de certaines méthodes bio-inspirées et de leur hybridation avec la classification basée sur les règles floues pour générer ou sélectionner des règles floues. Les résultats obtenus avec ces approches ont prouvé leurs performances. La plupart d'entre eux augmentent la précision et le nombre de modèles correctement classés et ou diminuent le nombre de règles et le nombre des exemples mal (incorrectement) classés. L'axe de recherche dans ce domaine est ouvert pour plus d'optimisation de la règle floue et à d'autres méthodes bio-inspirées.

Contribution et résultats

Sommaire

4.1 Introduction	54
4.2 Méthodes et techniques utilisé	55
4.3 Etude théorique sur l'effet des fonctions d'appartenance et des poids des règles sur la classification basée sur des règles floues	61
4.4 Les modèles proposés	69
4.5 Conclusion	92

4.1 Introduction

Les systèmes basés sur des règles floues (FRBS) sont l'une des méthodes les plus populaires en matière de reconnaissance des formes et d'apprentissage automatique. Les FRBS ont été appliqués avec succès à une grande variété de problèmes, y compris les problèmes de classification.

Pour construire un système de classification basé sur des règles floues, de nombreuses tâches importantes sont effectuées : la génération, l'apprentissage, la sélection des règles,

la construction des fonctions d'appartenance et l'apprentissage des poids des règles. Dans de tels systèmes flous, le choix du nombre de règles floues et la définition des fonctions d'appartenance et l'apprentissage du poids des règles constituent une tâche très importante.

Dans ce travail nous avons effectué une étude analytique sur l'effet des poids des règles (CF) et les fonctions d'appartenance (MF) sur la performance du système de classification basé sur les règles floues on se basant sur ces résultats nous avons proposé quelques modèles de systèmes de classification à base des règles floues optimale.

4.2 Méthodes et techniques utilisé

4.2.1 Clustering en utilisant soustractive clustering

4.2.1.1 Initialisation des fonctions d'appartenance par le clustering

Le clustering est le processus qui consiste à identifier des groupes de données similaires répondant à certains critères. Les méthodes de clustering ont été utilisées pour l'initialisation des règles floues. Le choix du nombre de règles floues et la définition des fonctions d'appartenance constituent les tâches difficiles dans le processus de conception des systèmes flous et des classificateurs à base des règles floues. L'utilisation des méthodes de clustering pour trouver une base de règles compacte constitue une approche alternative aux méthodes conventionnelles basées sur les partitions en grille et la sélection de règles.

L'avantage d'utilisation des approches basées sur le clustering pour l'initialisation des fonctions d'appartenance, dans lesquelles chaque règle correspond à un cluster, réside dans le fait qu'elles permettent de générer des classificateurs efficaces avec peu de règles. En effet, dans les méthodes de partition en grille, l'espace d'entrée est divisé en toutes les régions possibles et chacune d'entre elles est représentée par une règle floue. Ce processus génère un grand nombre de règles, en particulier dans les problèmes de grandes

dimensions où le nombre de règles devient exponentiellement important (malédiction de la dimension).

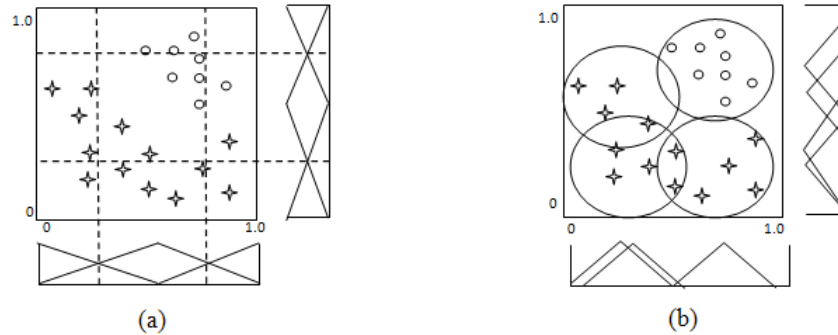


FIGURE 4.1 – Initialisation des fonctions d'appartenance

La figure 4.1 présente exemple de l'application de la partition en grille et l'application du clustering pour trouver les régions, ou chaque région représente une règle floue. On peut remarquer que dans le cas de la partition en grilles c.à.d fig.4.1(a) le nombre de règles égale au nombre de fonction d'appartenance (K) à la puissance nombre de dimension. Dans cet exemple $k = 3$ et le problème est de deux dimension alors le nombre de règles $= 3^2 = 9$ règles.

Par contre dans le cas de clustering le nombre des règles égale au nombre de cluster c-à-d 4 règles seulement.

4.2.1.2 Soustractive clustering (SC)

La méthode soustractive clustering Chiu (1994) [23] est une modification du clustering de montagne introduit par Yager et Filev (1994) [108]. Elle permet de déterminer à la fois le nombre de clusters et leurs paramètres qui peuvent être utilisées pour initialiser d'autres méthodes de clustering basées sur l'optimisation. Dans notre système, cette méthode est également utilisée pour déterminer le nombre de règles floues et les paramètres des fonctions d'appartenances.

Supposons que nous avons une collection de M exemples dans l'espace de N dimensions

$x_i = x_1, x_2, \dots, x_N$. Dans le soustractive clustering, chaque exemple est considéré comme un centre potentiel de cluster. La mesure du potentiel d'un exemple x_i étant un centre de cluster est calculée par l'équation suivante :

$$P_i = \sum_{j=1}^N e^{-\alpha \|x_i - x_j\|^2}, \alpha = 4/r_a^2 \quad (4.2.1)$$

Où $\|x_i - x_j\|$ est la distance euclidienne. r_a est le rayon, une constante positive qui définit le voisinage. La valeur potentielle de chaque exemple dépend alors du nombre de ses proches. L'influence d'un exemple voisin diminue de façon exponentielle avec le carré de sa distance, et le rayon, r_a , définit la quantité décroissante. Par conséquent, le rayon définit la région contenant les exemples les plus influents.

Après avoir défini le potentiel de tous les exemples, l'exemple ayant la valeur maximale est sélectionné comme premier centre. Considérons x_1^* en tant que premier centre de cluster et P_1^* est sa valeur potentielle. Après avoir sélectionné le $k^{i\text{ème}}$ centre de cluster, le potentiel de chaque exemple est calculé par l'équation :

$$P_i = P_i - P_k^* e^{-\beta \|x_i - x_k^*\|^2}, \beta = 4/r_b^2 \quad (4.2.2)$$

r_b , est une constante positive qui définit le rayon du voisinage où la fonction potentielle du point de données sera réduit. Ainsi, une quantité potentielle de chaque exemple est soustraite en fonction de sa distance par rapport au centre de cluster actuel. Les exemples proches du centre du cluster ont un potentiel réduit et sont donc peu susceptibles d'être sélectionnés comme centre du prochain cluster. Le processus d'acquisition de nouveaux centres de clusters et de révision des potentiels est répété jusqu'à ce que le critère d'arrêt soit satisfait. Dans le cas d'une grande région homogène, le point central sera choisi comme premier centre du cluster. Ensuite, le prochain centre de cluster sera sélectionné parmi les exemples éloignés du premier centre et ainsi de suite. Pour plus de détails sur le soustractive clustering, voir [23].

4.2.2 Optimisation par essais particuliers (OEP ou PSO)

Le PSO est proposée par Kennedy et Eberhart (1995) [63]. C'est une méta-heuristique inspirée de la nature et méthode d'optimisation stochastique. Dans le PSO, chaque particule dans l'espace de recherche représente une solution au problème d'optimisation et il possède ces trois propriétés : une position actuelle, une vitesse, et une meilleure position personnelle (expérience) dans l'espace de recherche. La position de la particule est influencée par la vitesse.

Chaque particule ajuste sa position et sa vitesse en fonction de sa meilleure position trouvée (p_{best}), et la meilleure expérience globale (g_{best}) trouvée par les autres particules pendant le processus de recherche. Le p_{best} est la position qui présente la meilleure valeur de fitness et le g_{best} est la position qui donne la meilleure solution parmi toutes les positions. La vitesse (v_{id}) et la position (x_{id}) sont mises à jour à l'aide de l'équation (4.2.3) et de l'équation (4.2.4) respectivement :

$$v_{id}(t+1) = w.v_{id}(t) + c_1.rand_1(pbest_{id}(t) - x_{id}(t)) + c_2.rand_2(gbest_{id}(t) - x_{id}(t)) \quad (4.2.3)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (4.2.4)$$

Où c_1 et c_2 sont deux valeurs constantes positives, $rand_1$ et $rand_2$ sont deux valeurs aléatoires dans la plage $[0, 1]$. w est un paramètre appelé poids d'inertie utilisé pour ajuster l'équilibre entre la recherche globale et locale [99]. Le poids d'inertie est calculé par l'équation(4.2.5) :

$$w = (T - t)/T \quad (4.2.5)$$

4.2.3 Optimisation en utilisant l'Algorithme de la recherche Coucou

L'Algorithme de la recherche Coucou (cuckoo search (CS)) est un algorithme basé sur l'intelligence en essaim qui a été développé par Yang et Deb (2009) [109], inspi-

rée du comportement naturel des coucous, en particulier des parasites obligatoires de la couvée de certaines espèces de coucous qui pondent leurs oeufs dans les nids d'autres oiseaux hôtes. Les auteurs ont défini l'algorithme CS en fixant trois règles qui idéalisent le comportement des coucous afin de devenir appropriées pour une implémentation en tant qu'algorithme informatique :

- Chaque coucou pond un oeuf à la fois et le dépose dans un nid choisi au hasard.
- Les meilleurs nids avec des oeufs de haute qualité seront transmis aux générations suivantes.
- Le nombre de nids hôtes disponibles est fixé et l'oeuf pondu par un coucou peut être découvert par l'oiseau hôte avec une probabilité $p_a \in (0, 1)$. Dans ce cas, l'oiseau hôte peut soit se débarrasser de l'oeuf, soit simplement abandonner le nid et construire un tout nouveau nids.

La première étape de l'algorithme génère une population initiale, où les nids d'hôtes sont positionnés dans l'espace de recherche de façon aléatoire. Dans la boucle principale qui suit, l'algorithme obtient une nouvelle position du coucou de façon aléatoire par le vol de Lévy et évalue son fitness. Ensuite, une solution aléatoire j est choisie, qui peut être remplacée lorsque la $i^{\text{ème}}$ solution est meilleure. Les pires nids peuvent être abandonnés et les nouveaux sont construits. Cependant, le maintien de la meilleure solution est effectué. Lors de la génération de nouvelles solutions $x(t + 1)$ pour, disons, un coucou est un vol de Lévy est effectué.

$$x_i(t + 1) = x_i(t) + \alpha \text{Lévy}(\gamma) \quad (4.2.6)$$

4.2.4 Génération des règles floue

4.2.4.1 Formes des règles utilisées

Dans nos travaux nous utilisons deux formes de règle avec CF et sans CF .

La première forme de la règle floue SI-ALORS pour un problème de classification des formes de H classe avec n attributs est la suivante :

$$\text{Règle } R_j : \text{ Si } x_1 \text{ est } A_{j1} \text{ et } x_2 \text{ est } A_{j2} \text{ et...et } x_n \text{ est } A_{jn} \quad (4.2.7)$$

$$\text{Alors class } C = C_j$$

Où :

R_j est l'étiquette de la $j^{\text{ème}}$ règle floue Si-Alors

$x_n = (x_1, \dots, x_n)$ est un vecteur des exemples à n dimensions

A_{j1}, \dots, A_{jn} sont des sous-ensembles flous sur l'intervalle unitaire $[0, 1]$,

C_j est la classe conséquente ou résultante

La deuxième forme de la règle floue SI-ALORS utilisé dans notre travail est comme suit :

$$\text{Règle } R_j : \text{ Si } x_1 \text{ est } A_{j1} \text{ et } x_2 \text{ est } A_{j2} \text{ et...et } x_n \text{ est } A_{jn} \quad (4.2.8)$$

$$\text{Alors class } C = C_j \text{ avec } CF = CF_j$$

Où CF est le degré de certitude de la règle floue SI-ALORS R . Habituellement, CF est un nombre réel dans l'intervalle d'unités $[0,1]$ (c.-à-d. $0 \leq CF \leq 1$).

4.2.4.2 La procédure de génération des règles floues

Après le partitionnement de l'espace du problème de classification en sous régions on utilisant le clustering ou bien les méthodes de partition en grille, la partie antécédente (si x_1 est A_1 et x_2 est A_2 ... et x_n est A_n) de la règle est déterminé par la projections des régions pour trouver les degrés d'appartenance de chaque exemple x . Pour générer la partie conséquente nous utilisons la procédure présentée dans [53] . Pour une partie donné j représenté par la règle R_j , la procédure de détermination de la partie conséquente (c'est-à-dire la classe C_j et le poids CF_j est donnée comme suit :

Etape 1 : Calculer pour chaque classe C_h la somme des degrés de compatibilité de toutes les exemples d'apprentissage. Exemple : $x_p, x_p = x_{p1}, x_{p2}, \dots, x_{pN}$ de cette classe

$$\beta_{jh} = \sum_{x_p \in C_h} (\mu_{j1}(x_{p1}) * \mu_{j2}(x_{p2}) * \dots * \mu_{jN}(x_{pN})) \quad (4.2.9)$$

Où β_{jh} est la somme des degrés de compatibilité correspondant à la classe C_j et μ_{ji} est le $i^{\text{ème}}$ MF correspondant à la règle j .

Etape 2 : Trouver la classe conséquente C_j pour la règle R_j . C_j est donné par :

$$C_j = \operatorname{argmax}_h(\beta_{jh}) \quad (4.2.10)$$

Etape 3 : Si la règles contiens un poids dans ça partie conséquente, il existe plusieurs méthodes heuristiques pour le calculer.

Après qu'un ensemble de règles S ait été donné, un exemple d'entrée $x_p = (x_{p1}, x_{p2})$ est classé par une règle gagnante unique R_j dans S . La règle gagnante floue SI-ALORS avec l'utilisation des degrés de certitude (c.-à-d. les poids des règles), est déterminée comme la règle avec le produit maximum du degré de compatibilité et du degré de certitude.

$$\mu(x_p).CF_j = \max\{\mu(x_p).CF_j | R_j\} \quad (4.2.11)$$

4.3 Etude théorique sur l'effet des fonctions d'appartenance et des poids des règles sur la classification basée sur des règles floues

Le but de cette étude est d'examiner l'effet des degrés de certitude et des fonctions d'appartenance sur la performance des systèmes de classification basés sur des règles floues. Nous présentons une étude théorique basée sur l'analyse des frontière de décision et du taux de classification des exemples synthétiques. Supposons que notre espace de caractéristique A est l'unité carrée $[0, 1] \times [0, 1]$. Nous supposons également que $x_p = (x_{p1}, x_{p2}), p = 1, 2, \dots, M$ sont considérés comme des exemples d'apprentissage

de la classe H . Dans cette étude, nous utilisons une fonction d'appartenance triangulaire symétrique pour $A_i, i = 1, 2, \dots, K$ où K est le nombre de variables linguistiques dans chaque axe. Chaque sous-espace a sa propre règle floue.

Fonction triangulaire : définie par une limite inférieure a , une limite supérieure b et une valeur m , où

$$a < m < b$$

$$\mu_A = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{m-a}, & a < x \leq m, \\ \frac{b-x}{b-m}, & m < x < b \\ 0, & x > b, \end{cases} \quad (4.3.12)$$

La génération des règles floues pour les problèmes de classification se compose de deux phases : partition floue de l'espace des exemples et identification d'une règle floue pour chaque sous-espace flou.

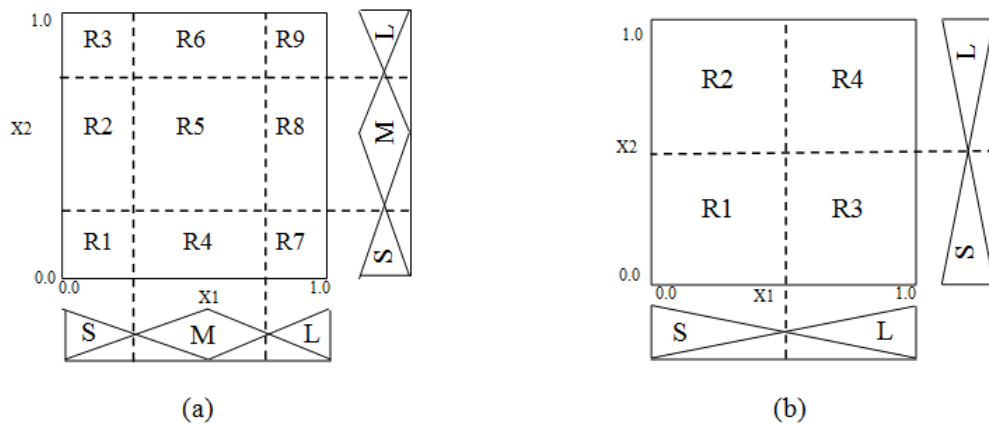


FIGURE 4.2 – Zone de décision de chaque règle floue SI-ALORS avec : (a) $K=3$ et (b) $K=2$

Après avoir partitionné l'espace des exemples en K^2 sous-espaces flous, nous avons déterminé une règle pour chaque sous-espace flou, nous utilisons la méthode présentée dans la section (4.2.4).

Les règles utilisées dans ce travail sont de la forme (2) présenté dans la sous-section (4.2.4.1). Les règles floues générées correspondant aux sous-espaces présentés dans la Fig 4.2.(b) peuvent être écrites sous la forme :

R_1 : Si x_1 est S et x_2 est S Alors class $C = C_1$ avec CF_1

R_2 : Si x_1 est S et x_2 est L Alors class $C = C_2$ avec CF_2

R_3 : Si x_1 est L et x_2 est S Alors class $C = C_2$ avec CF_3

R_4 : Si x_1 est L et x_2 est L Alors class $C = C_1$ avec CF_4

4.3.1 Effet des poids des règles (CF)

Dans cette section, nous examinons l'effet du poids des règles sur les frontières de décision obtenues à l'aide de règles floues SI-ALORS avec ou sans degré de certitude. Nous considérons deux problèmes de classification. Ces problèmes sont bi-dimensionnels et ils sont de 2 classes ('x','o') présentées dans la figure 4.3.



FIGURE 4.3 – Deux problèmes synthétiques de classification (a) 1^{er} problème : linéairement séparable (b) 2^{ème} problème : non linéairement séparable.

Le tableau 4.1 présente le taux de classification du 1^{er} et 2^{ème} modèle avec et sans CF , avec différentes valeurs de K .

le taux de classification pour le 1 ^{er} problème		
K	sans CF	avec CF
K=2	75	86.36
K=3	72.72	81.81
K=4	81.81	95.45

le taux de classification pour le 2 ^{ème} problème		
K	sans CF	avec CF
K=2	92	67.5
K=3	67.5	90
K=4	95	97.5

TABLE 4.1 – Le taux de classification des deux problèmes de classification avec divers K

La TABLE 4.1, illustre le taux de classification, on note que l'utilisation du degré de certitude améliore le taux de classification sauf dans le cas K=2 dans le 2^{ème} problème de classification.

Après avoir partitionné l'espace des exemples du problèmes de classification en K^2 sous-espaces flous ou K=3, nous avons déterminé une règle pour chaque sous-espace flou. Les règles floues générées correspondant aux neuf sous-espaces peuvent être écrites sous comme suit :

R_1 : Si x_1 est S et x_2 est S Alors class $C = 1$ avec $CF = 1$

R_2 : Si x_1 est S et x_2 est M Alors class $C = 1$ avec $CF = 0.3$

R_3 : Si x_1 est S et x_2 est L Alors class $C = 2$ avec $CF = 0.37$

R_4 : Si x_1 est M et x_2 est S Alors class $C = 2$ avec $CF = 0.16$

R_5 : Si x_1 est M et x_2 est M Alors class $C = 1$ avec $CF = 0.17$

R_6 : Si x_1 est M et x_2 est L Alors class $C = 1$ avec $CF = 0.015$

R_7 : Si x_1 est L et x_2 est S Alors class $C = 2$ avec $CF = 1$

R_8 : Si x_1 est L et x_2 est M Alors class $C = 2$ avec $CF = 0.07$

R_9 : Si x_1 est L et x_2 est L Alors class $C = 1$ avec $CF = 1$

La Fig.4.4 et la Fig.4.5 illustrent les frontières de décision obtenues à l'aide de neuf règles floues si-alors sans et avec CF respectivement.

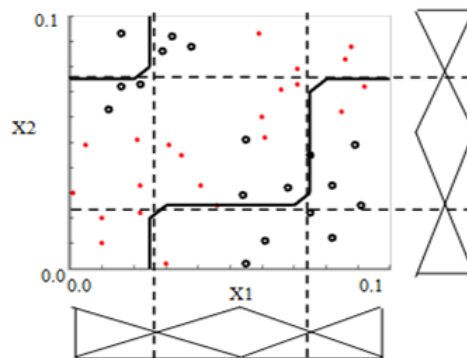


FIGURE 4.4 – Frontières de décision sans CF, K=3

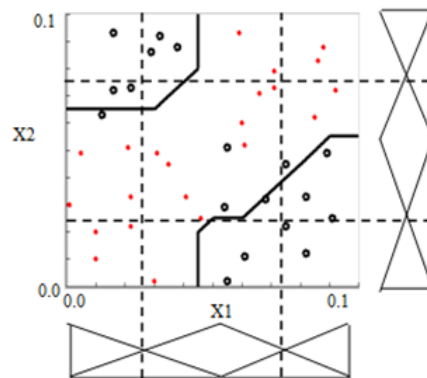


FIGURE 4.5 – Frontières de décision avec les CF

D'après la Fig.4.4 et la Fig. 4.5 Nous pouvons noter que l'utilisation du degré de certitude avec des fonctions d'appartenance fixe améliore les frontières de décision de cet exemple.

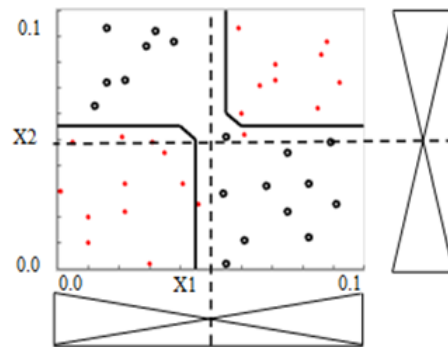


FIGURE 4.6 – Frontières de décision avec CF ajusté

	le taux de classification		
	sans CF	avec CF	CF ajuster
K=2	67.5	95	97.5

TABLE 4.2 – Taux de classification avec CF ajusté pour problème de classification 2

Le taux de classification obtenu pour ce problème est de 95% avec des degrés de certitude heuristiques calculés par la formule (4.4.15) et 67,5% sans degrés de certitude. Nous pouvons améliorer la performance de classification (97,5%) et les frontières de décision en ajustant le degré de certitude de chaque règle floue SI-ALORS, comme le montrent la Figure 4.6 et le TABLEAU 4.2 .

4.3.2 Effet des fonctions d'appartenance (MF)

Dans cette section, nous utilisons des règles floues (forme 1) sans CF et nous avons modifié les fonctions d'appartenance, nous utilisons un autre problème de classification 3.

La Fig.4.7 et la Fig.4.8 illustrent les frontières de décision obtenues à l'aide de quatre règles floues si-alors ou les fonctions d'appartenance sont fixées et modifiées respectivement.

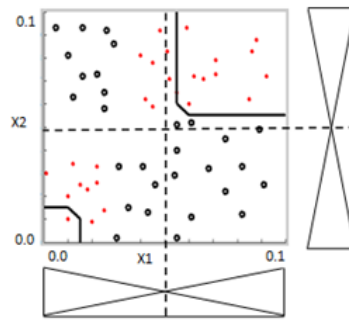


FIGURE 4.7 – Les frontières de décision avec CF, $K = 2$

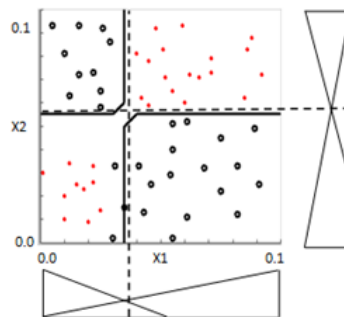


FIGURE 4.8 – Les frontières de décision avec fonction d’appartenance modifiée

Nous avons remarqué que plusieurs exemples sont mal classifiés comme le montre la figure 4.7. Mais la modification de la fonction d’appartenance de chaque valeur linguistique antécédente peut améliorer les frontières de décision même sans degré de certitude comme le montre la figure 4.8.

	le taux de classification		
	MF non-modifier	MF modifier sans CF	MF modifier avec CF
$K=3$	87.71	91.22	92.98

TABLE 4.3 – Taux de classification du problème de classification 3

Nous constatons que l’utilisation du degré de certitude améliore le taux de classifi-

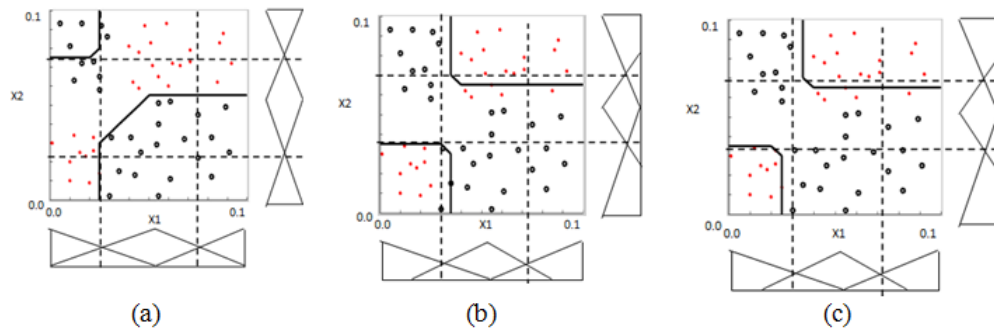


FIGURE 4.9 – Les frontières de décision : (a) avec CF, (b) fonction d'appartenance modifiée sans CF (c) fonctions d'appartenance modifiées avec CF

cation et les frontières de décision, même lorsque nous modifions la fonction d'appartenance. Mais l'utilisation des deux donne un meilleur résultat.

Dans cet étude, nous avons analysé l'effet de la fonction d'appartenance et du degré de certitude dans les systèmes de classification basés sur des règles floues. L'analyse était basée sur l'examinations du taux et des frontières de décision des problèmes synthétiques bidimensionnelles. Pour cette tâche, nous avons effectué des tests, d'une part, avec la fonction d'appartenance centrée et modifiée et, d'autre part, avec et sans degré de certitude. Les résultats obtenus ont montré que : Premièrement, l'utilisation de la fonction d'appartenance modifiée sans pondération de la règle a un effet significatif sur la performance de la classification. Deuxièmement, l'utilisation d'une fonction d'appartenance fixe avec les poids de la règle améliore également les performances de la classification. Enfin, l'ajustement de ces deux fonctions améliore davantage les performances de la classification.

4.4 Les modèles proposés

4.4.1 1^{er} modèle :

4.4.1.1 L'idée principale

Ce modèle est un système de classification basé sur des règles floues dans lequel la génération des fonctions d'appartenance s'effectue automatiquement à l'aide d'une méthode d'apprentissage non supervisée. La génération des fonctions d'appartenance gaussiennes se fait en deux étapes. Premièrement, la méthode de clustering soustractive est utilisée pour définir à la fois le nombre et les centres de clusters qui sont utilisés pour l'initialisation des fonctions d'appartenances. Deuxièmement, les fonctions d'appartenance obtenues sont fusionnées afin d'améliorer la compréhension des règles obtenues. Pour évaluer cette approche, des tests ont été effectués sur un exemple synthétique, la base de données Iris et la base de données Wine.

4.4.1.2 Processus

Ce modèle inclut une méthode d'initialisation des paramètres de la fonction d'appartenance à l'aide d'une méthode de clustering non supervisée (soustractive clustering). Nous utilisons d'abord les centres de clusters et leur espace d'influence sur chacune des dimensions de données pour générer les fonctions d'appartenance, puis nous fusionnons les fonctions d'appartenance proches. Notre approche se compose de trois phases principales :

1. Dans la première phase : les données sont regroupées en utilisant la méthode soustractive clustering ,
2. Dans la deuxième phase, nous utilisons les résultats de la première phase pour la construction des fonctions d'appartenance, et nous fusionnons les plus proches.
3. Dans la troisième phase, les fonctions d'appartenance générées sont utilisées dans la génération des règles floues.

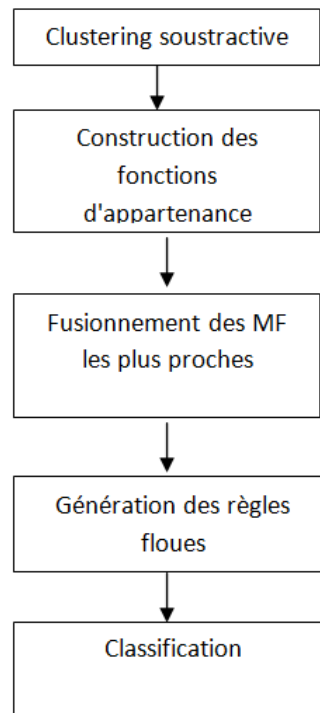


FIGURE 4.10 – Shéma générale du premier modèle proposé

Les étapes précédentes sont détaillées comme suit :

- Utiliser la méthode de soustractive clustering pour trouver les clusters de la base de données.

- Les centres (C) des clusters sont considérés comme les centres des MFs et le nombre de clusters est considéré comme le nombre des MFs.

- Le paramètre sigma (σ) de cluster est utilisé pour initialiser la largeur (σ) de MF gaussienne.

- Les fonctions d'appartenance proches sont ensuite fusionnées comme suit :

Lorsque le sommet de l'intersection de deux MF gaussiennes avec des paramètres (c_1, σ_1) et (c_2, σ_2) dépasse un seuil, γ . Nous les fusionnons dans un nouveau MF avec des paramètres $(c_{fusion}, \sigma_{fusion})$ dont le centre est calculé par la formule suivante [101] :

$$c_{fusion} = \frac{c_1\sigma_1 + c_2\sigma_2}{\sigma_1 + \sigma_2} \quad (4.4.13)$$

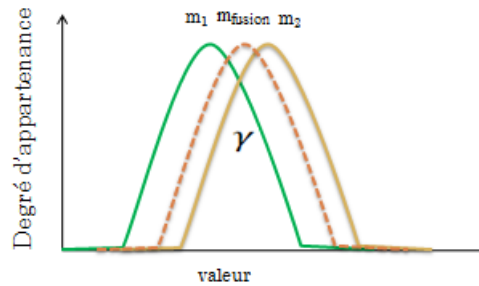


FIGURE 4.11 – Exemple de fusion de deux MF

De même, la largeur de la fonction d'appartenance résultante est obtenue par [101] :

$$\sigma_{fusion}^2 = \frac{\sigma_1^3 + \sigma_2^3}{\sigma_1 + \sigma_2} \quad (4.4.14)$$

Les fonctions d'appartenance générées sont utilisées dans la construction de la partie antécédente des règles floues.

La partie conséquente se construit en suivant les étapes expliquées dans la sous-section (4.2.4.2)

4.4.1.3 Test et résultats

Pour évaluer ce modèle, nous avons effectué des tests sur des exemples synthétiques et sur les bases de données Iris et Wine et nous les avons comparé avec une autre méthode heuristique.

Après que les exemples synthétique ont été regroupés avec la méthode de soustractive clustering , les centres de clusters obtenus sont les centres (C) de sept fonctions d'appartenance gaussiennes pour la 1^{ère} caractéristique, les sept centres sont :

0.12 0.16 0.80 0.45 0.50 0.85 0.82

Pour la 2^{ème} caractéristique, les sept centres sont :

0.16 0.80 0.26 0.21 0.70 0.66 0.90

Les écarts types des deux caractéristiques sont :

(σ) : 0.1039 et 0.1029

La figure 4.12 présente les sept fonctions d'appartenances gaussienne générées par la méthode proposée et leurs fusionnement. Ou (b) et (d) sont les résultats de fusionnement des fonctions d'appartenance présentée dans (a) et (c) respectivement

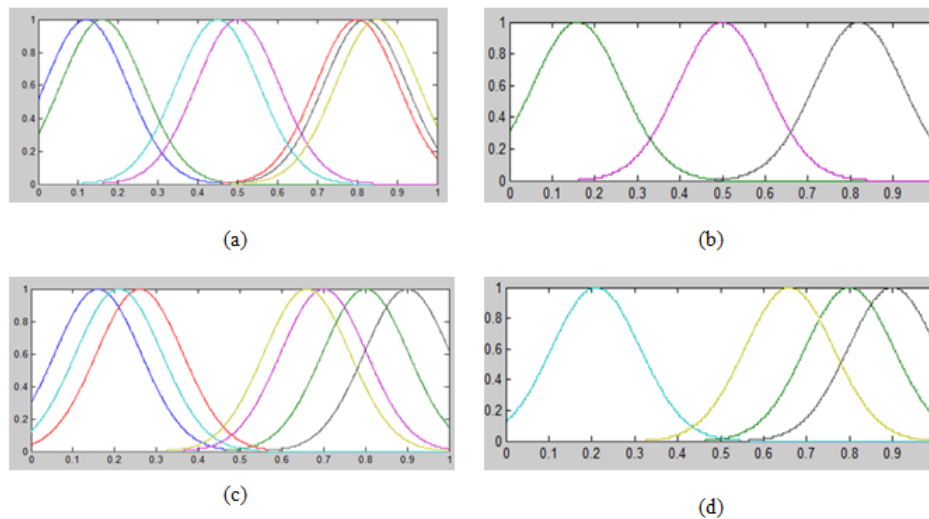


FIGURE 4.12 – Fusion des fonctions d'appartenance générées (MF) (a). MFs générées pour la caractéristique 1 (b). MFs fusionnées pour la caractéristique 1 (c). MFs générées pour la caractéristique 2 (d). MFs fusionnées pour la caractéristique 2

Dans le tableau 4.4. nous présentons le nombre de règles et le taux de classification obtenus par l'utilisation de la méthode proposée sur un exemple synthétique en comparaison avec la méthode heuristique présentée dans [53].

Les méthodes	Méthode heuristique	Méthode de clustering	Après la fusion
Nombre de règle	9	7	7
Taux de classification	85.83%	94.16%	96.66 %

TABLE 4.4 – Résultats de la classification de l'exemple synthétique.

Dans la figure 4.13 (a), nous présentons les frontière de décision obtenue par l'uti-

lisation de la méthode heuristique présentée dans [53] et la figure 4.13 (b) présente Les frontières de décision obtenue par l'utilisation de la méthode proposée sur un exemple synthétique. La figure.4.13 (c) présente les frontières de décision obtenue après fusion.

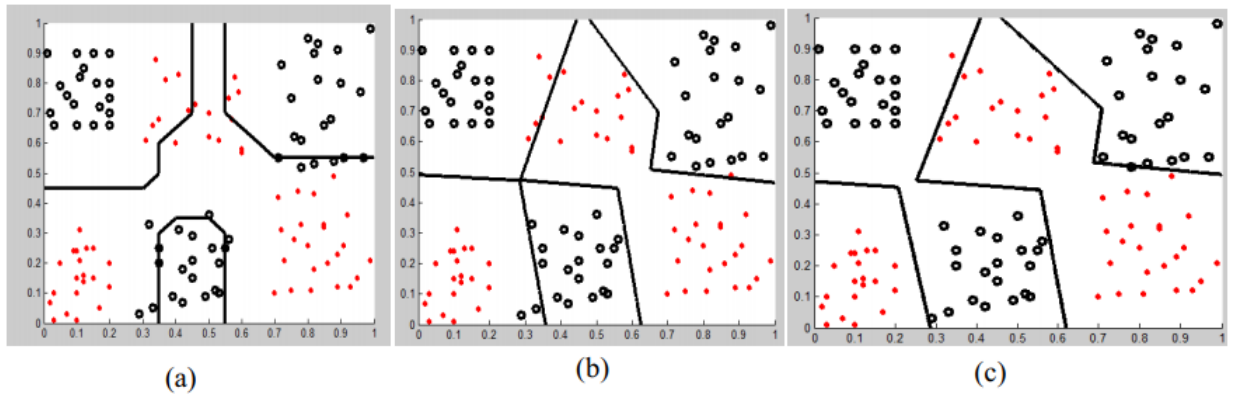


FIGURE 4.13 – Les frontières de décision de l'exemple synthétique par : (a) la méthode heuristique [53], (b) méthode proposée, (c) fusion des MF

D'après figure .4.13 et le tableau 4.4 on peut noter que l'initialisation de la fonction d'appartenance par la méthode de clustering et la fusion améliorent les frontière de décision et le taux de classification de cet exemple.

Application sur la base de données Iris

Dans un premier test, nous n'avons utilisé que les deux caractéristiques ($3^{\text{ème}}$ et $4^{\text{ème}}$) pour visualiser les frontières de décision.

Après le clustering, les centres des clusters obtenus sont utilisés pour initialiser les centres des fonctions d'appartenance gaussiennes. Pour la $3^{\text{ème}}$ caractéristique , les centres calculés sont : 0,2059 0,6324 0,7941 et pour la $4^{\text{ème}}$ caractéristique les centres correspondants sont : 0.0147 0.1912 0.2941. Par ailleurs, le paramètre (σ) des Gaussiens calculés est : 0,0920 et 0,0374 pour la $3^{\text{ème}}$ et $4^{\text{ème}}$ caractéristique respectivement

Le tableau 4.5. présente le nombre de règles et le taux de classification obtenus à l'aide de la méthode heuristique et de la méthode proposée pour l'initialisation de la fonction

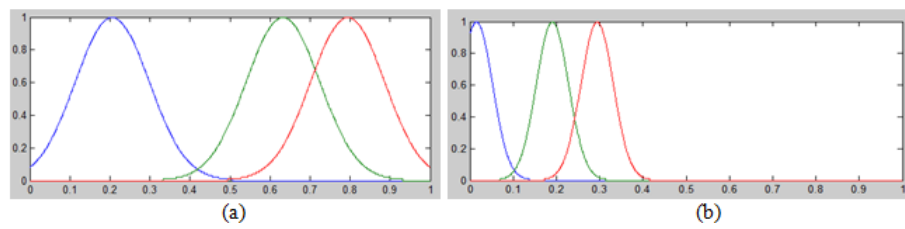


FIGURE 4.14 – Fonctions d’appartenance générées de : (a) la 3^{ème} caractéristique de l’Iris (b) la 4^{ème} caractéristique de l’Iris

d’appartenance des 3^{ème} et 4^{ème} paramètres Iris.

	Les méthodes	Méthode heuristique	Méthode proposée
3 ^{ème} et 4 ^{ème} paramètres Iris	Nombre de règle	10	3
	Le taux de classification	94	96

TABLE 4.5 – Résultats de la classification des 3^{ème} et 4^{ème} paramètres de l’Iris

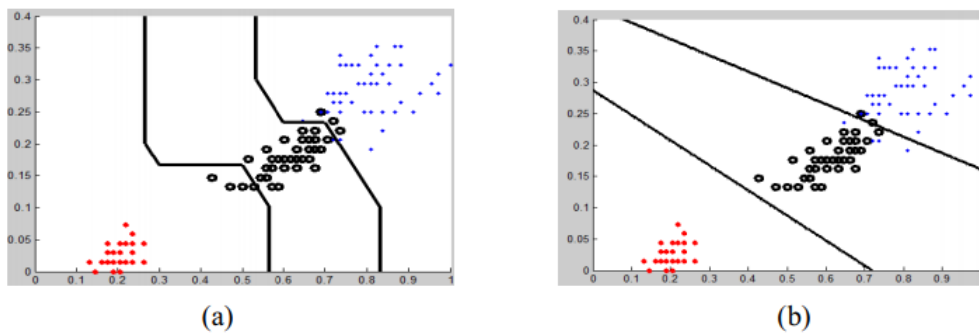


FIGURE 4.15 – Les frontière de décision des 3^{ème} et 4^{ème} paramètres de l’iris : (a) avec la méthode heuristique [53],(b) méthode proposée

Tableau 4.5. et figure 4.15. montrent que la méthode proposée améliore le taux de classification avec moins de règles et améliore également les frontière de décision.

Dans le second test, nous utilisons toutes les caractéristiques de la base de données Iris. Le tableau 4.6 présente le nombre de règles et le taux de classification obtenu à l'aide de la méthode heuristique et de la méthode proposée pour l'initialisation de la fonction d'appartenance.

Du tableau 4.6 le taux de classification obtenu des données sur l'iris est de 94 % avec la

Les méthodes	Méthode heuristique	Méthode proposée
Nombre de règle	28	5
Le taux de classification	94	95.66

TABLE 4.6 – Résultats de la classification des données Iris

méthode heuristique et de 95,66 % avec la méthode proposée. Et le nombre de règles est de 28 règles utilisant la méthode heuristique et 5 règles utilisant la méthode proposée. Les tests montrent que notre approche de la construction automatique permet d'améliorer la performance de la classification et réduire le nombre de règles floues.

Application sur la base de données Wine

Nous avons également évalué notre approche à l'aide de la base de données "Wine".

Les résultats obtenus sur cette base de données ont démontré une fois de plus que notre

Les méthodes	Méthode heuristique	Méthode proposée
Nombre de règle	12	7
Le taux de classification %	93.26	96.06

TABLE 4.7 – Résultats de la classification des données Wine

approche de construction automatique améliore les performances de classification et réduit le nombre de règles floues.

Dans ce modèle, nous avons présenté une méthode pour construire une fonction d'appartenance basée sur la méthode de clustering. Le système proposé comprend deux phases distinctes. Dans le premier cas, les fonctions d'appartenance sont déterminées à l'aide de la méthode soustractive et d'une technique de fusion. Dans le second cas, les grades des règles sont déterminés. Le système proposé a démontré avec succès sa compétence à générer les fonctions d'appartenance. Il a amélioré à la fois le taux de classification et les frontières de décision avec peu de règles.

4.4.2 2^{ème} modèle :

4.4.2.1 Idée principale

Ce modèle consiste en un système de classification basé sur les règles floues dans lequel la génération des fonctions d'appartenance et des poids des règles sont effectués, respectivement, par une méthode d'apprentissage non supervisée et d'une méthode d'optimisation bio inspirée. Ce modèle comprend donc deux étapes. Dans la première, la méthode de soustractive clustering est appliquée aux données d'apprentissage pour obtenir de différentes régions. Ensuite, chacune de ces régions sera représentée par une règle floue. Les paramètres des fonctions d'appartenance, c'est-à-dire les centres et les largeurs, sont également obtenus dans cette étape par la projection des clusters. Dans la deuxième étape, les poids des règles sont générés automatiquement par une méthode d'optimisation bio-inspirée. Selon la méthode d'optimisation utilisée, ce modèle comporte de deux versions : La première est basée sur l'optimisation par essaims de particules quant la deuxième, elle est basée sur l'algorithme de recherche coucou (cuckoo search). Les deux versions sont basées sur l'utilisation des règles floues SI-ALORS de la 2^{ème} forme présentée dans la sous-section (4.2.4.1).

Nous évaluons le premier modèle en utilisant les base de données Iris et " wine " et nous analysons les performances en comparant la méthode proposée avec d'autres différentes méthodes de calcul du CF. Puis, nous appliquons le deuxième modèle pour la classifica-

tion de la base de données médicale : tumeurs du cancer du sein (Wisconsin)

4.4.2.2 Processus

Les différentes phases de l'approche proposée peuvent être résumées comme suit :

- Dans la première étape : les données sont regroupées à l'aide de la méthode de soustractive clustering.
- Dans la deuxième étape, nous utilisons les résultats de la première phase pour la construction des fonctions d'appartenance.
- Dans la troisième étape, les fonctions d'appartenance obtenues sont utilisées dans la génération des règles floues.
- Finalement, nous calculons le poids de la règle en utilisant l'optimisation par essaims de particules pour la première version et l'algorithme de recherche coucou pour la deuxième version.

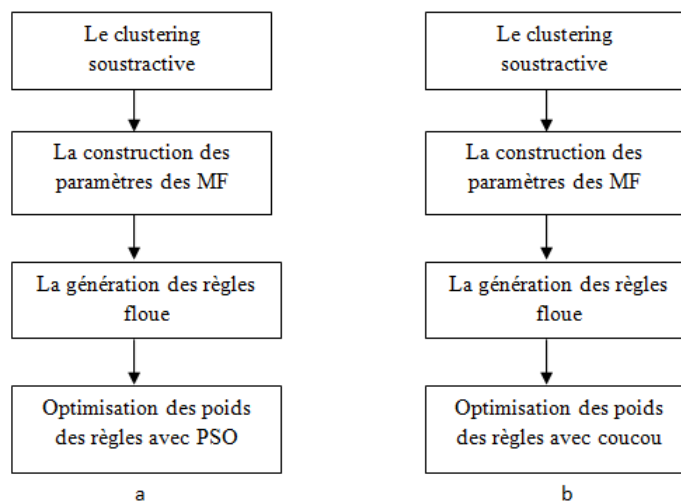


FIGURE 4.16 – Schéma générale du 2^{ème} modèle proposée (a) la première version (b) la deuxième version

Pour la construction de la fonction d'appartenance pour les variables d'entrée, nous

suivons les étapes suivantes :

- ▷ Appliquer la méthode de soustractive clustering pour trouver des clusters de la base de données.
- ▷ Les centres (C) des clusters sont considérés comme les centres de MF et le nombre de clusters est le nombre de MF.
- ▷ Le paramètre (σ), taille du cluster, est utilisé pour initialiser le (σ) de la MF gaussienne. Les fonctions d'appartenance finales générées sont utilisées dans la construction de la partie antécédente des règles floues. La partie conséquente est déterminée à partir des exemples d'apprentissage donnés par la procédure présentée dans la sous-section (4.2.4.2). Dans la première version le CF est calculé par le PSO où chaque particule dans l'espace de recherche représente une solution (CF(nb règle)) au problème d'optimisation. La valeur initiale de la position et de la vitesse sera initialisée de façon aléatoire. Nous mettons à jour la vitesse et la position en utilisant respectivement eq (4.2.3) et (4.2.4), jusqu'à atteindre l'objectif qui est la maximisation de la fitness ou atteindre le nombre maximum d'itérations.

Dans la deuxième version, le CF est calculé par recherche coucou (cuckoo search) comme suit : Une génération est représentée par un ensemble de nids d'hôtes. Chaque nid porte un oeuf qui représente une solution (CF(nb règle)). Chaque génération porte les meilleurs oeufs à la génération suivante (meilleure fitness). La qualité des oeufs est améliorée en générant un nouvel oeuf à partir de l'oeuf d'un nid sélectionné au hasard, qui est formé par un mouvement aléatoire sur l'oeuf sélectionné. Si le nouvel oeuf est supérieur à un oeuf existant dans un autre nid choisi au hasard, alors l'ancien oeuf est remplacé par le nouveau. Le nombre de nids reste fixe à chaque génération. Avec une certaine probabilité P_a , un nouvel oeuf généré aléatoirement remplace l'oeuf du nid le moins bien classé. Lors de la génération de nouvelles solutions $x(t + 1)$ pour un coucou i , un vol de prélèvement est effectué en utilisant l'équation (4.2.6).

4.4.2.3 Tests et résultats

Version 01 : Pour évaluer la méthode proposée dans la première version de ce modèle, nous avons effectué des tests sur les bases de données Iris et Wine et nous avons les comparer avec d'autre méthodes heuristique utiliser pour le calcul des poids des règles. Les poids des règle floues utilisés dans la comparaison sont calculés comme suit :

$$CF^I = (\beta_{classC} - \beta) / \sum_{C=1}^H \beta_{classC} \quad (4.4.15)$$

Où

$$\beta = \sum_{C=1, C \neq 1}^H \beta_{classC} / (c - 1) \quad (4.4.16)$$

La valeur de confiance est calculée comme suit :

$$c(A_q \Rightarrow C_q) = \max(c(A_q \Rightarrow Class_h)) \mid h = 1, 2, \dots, H \quad (4.4.17)$$

Où A_q est les conditions antécédentes de la règle floue R_q et C_q est la classe conséquente de la règle floue R_q .

$$CF^{II} = c(A_q \Rightarrow C_q) \quad (4.4.18)$$

$$CF^{III} = c(A_q \Rightarrow C_q) - \frac{1}{H-1} \sum_{h=1, h \neq 1}^H c(A_q \Rightarrow Class_h) \quad (4.4.19)$$

$$CF^{IV} = c(A_q \Rightarrow C_q) - \max(c(A_q \Rightarrow Class_h)) \mid h = 1, 2, \dots, H; h \neq C_q \quad (4.4.20)$$

$$CF^V = c(A_q \Rightarrow C_q) - \sum_{h=1, h \neq 1}^H c(A_q \Rightarrow Class_h) \quad (4.4.21)$$

Le tableau 4.8. présente le taux de classification obtenus de la méthode de clustering pour l'initialisation de la fonction d'appartenance et le pso pour le calcul de CF en comparaison avec les cinq différents poids de règles heuristiques appliqué sur la base de données Iris.

les CFs	CF^I	CF^{II}	CF^{III}	CF^{IV}	CF^V	PSO
Le taux %	96	96.66	96	96	96	97.33

TABLE 4.8 – Résultats de la classification sur la base Iris

Du tableau 4.8 On peut noter que le taux de classification obtenu des données de Iris est de 96 % en utilisant la méthode de regroupement avec toutes les CF sauf en utilisant la CF^{II} qui donne 96,66 % et le PSO est de 97,33 %.

Les tests démontrent que notre approche de construction automatique peut améliorer la performance de la classification avec un nombre réduit de règles floues.

Nous avons également évalué notre approche en utilisant la base de données Wine. Le tableau 4.9. présente le taux de classification obtenus de la méthode de clustering pour l'initialisation de la fonction d'appartenance et les cinq différents poids de règles heuristiques sur la base de données Wine en comparaison avec notre proposition.

les CFs	CF^I	CF^{II}	CF^{III}	CF^{IV}	CF^V	PSO
Le taux	96.06	96.06	96.06	96.06	96.06	97.99

TABLE 4.9 – Résultats de la classification sur la base Wine

Du tableau 4.9 On peut noter que le taux de classification obtenu des données de Wine est de 96.06% en utilisant la méthode de regroupement avec toutes les CF par contre le taux est de 97.99% pour le CF calculer par le PSO.

Les résultats obtenus sur cette base de données ont démontré une fois de plus que notre approche de construction automatique améliore les performances de classification.

Version 02 [90] : Ce modèle est un classificateur amélioré appliqué pour le diagnostic des tumeurs du cancer du sein. L'algorithme a été appliqué à des cas cliniques de cancer du sein obtenus de l'Université du Wisconsin.

Le tableau 4.10. présente le taux de classification obtenus par notre modèle, en comparant avec d'autres méthodes dans la littérature.

Méthode	Proposée	SNN [85]	EMEM-PCA-CART	NFS [45]
			Fuzzy Rule Based [78]	
Le taux %	97.72	96.38	93.2	97.8

TABLE 4.10 – Résultats de la classification sur la base Wisconsin

Du tableau 4.10 On peut noter que le taux de classification obtenu des données de Wisconsin est de 97.72%. Les résultats obtenus sur cette base de données ont démontré une fois de plus que notre approche de construction automatique améliore les performances de classification et donne des résultats compétitive.

4.4.3 3^{ème} Modèle [91] :

4.4.3.1 Idée principale

L'extraction des règles floues à partir de données à l'aide de soustractive clustering ou de toute autre méthode de clustering repose généralement sur deux étapes. La première étape consiste à former des régions. La seconde est la génération et l'apprentissage des règles en fonction des régions obtenues. Les méthodes de clustering sont généralement appliquées à un ensemble d'échantillons non étiquetés. Par conséquent, certains des clusters générés peuvent inclure des échantillons de différentes classes. La figure 4.17-a illustre un exemple d'une telle situation : les deux clusters générés incluent des exemples des deux classes.

Dans ce cas, il est difficile de définir les parties conséquentes des règles floues correspondantes, c'est-à-dire la classe conséquente. Pour faire face à ce problème, nous proposons d'appliquer le clustering sur chaque classe séparément (Figure 4.17 -b-). Dans ce cas,

chaque cluster comprend des échantillons de la même classe, ce qui simplifie la définition des parties conséquentes des règles floues correspondantes.

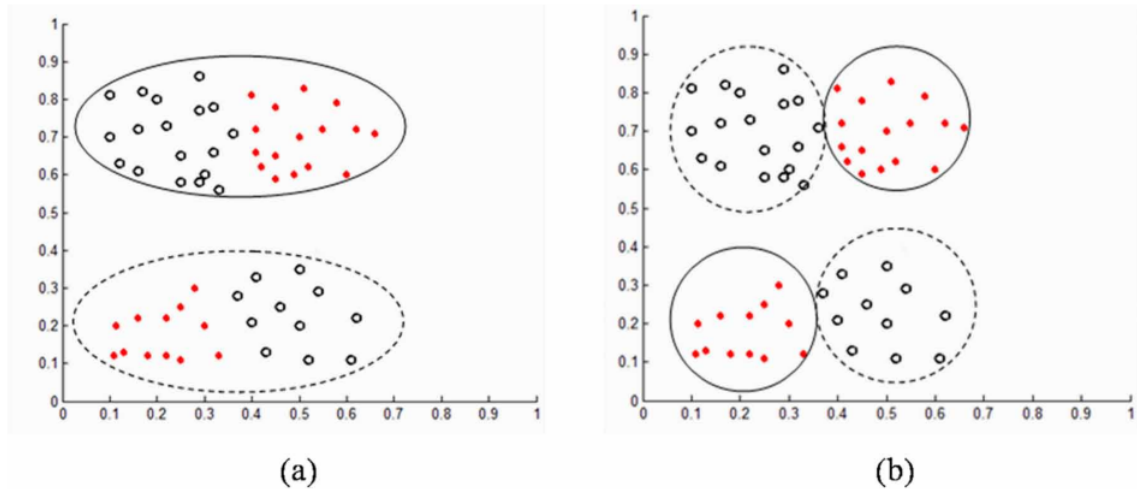


FIGURE 4.17 – Application de soustractive clustering sur : (a) tous les ensembles de données, (b) et sur chaque classe séparément

La méthode de soustractive clustering dépend du rayon. Ce paramètre influence le nombre, les centres et la taille des clusters obtenus. Les petites valeurs de rayon produisent un grand nombre de clusters et par conséquent un grand nombre de règles, et vice versa. Figure.4.18 illustre un exemple de clustering utilisant différentes valeurs de rayon. Le nombre et les formes des clusters dépendent fortement de ce paramètre. Dans cette figure, la partie (a) correspond a une grande valeur du rayon ce qui est traduit par un nombre insuffisant de clusters. La partie (c) correspond a une petite valeur du rayon et par conséquent un très grand nombre de clusters. Finalement la partie (b) correspond a la valeur optimale du rayon qui permet d’avoir le nombre optimale de clusters. Ainsi, définir les valeurs optimales des rayons permet d’obtenir une base de règles floue optimale.

Pour ce faire, nous proposons d’utiliser PSO et, pour obtenir un bon compromis précision/compacité, nous suggérons d’utiliser une fonction multi objectif qui inclut à la fois la précision et le nombre de règles. L’objectif est d’avoir un classificateur efficace avec un

nombre minimum de règles.

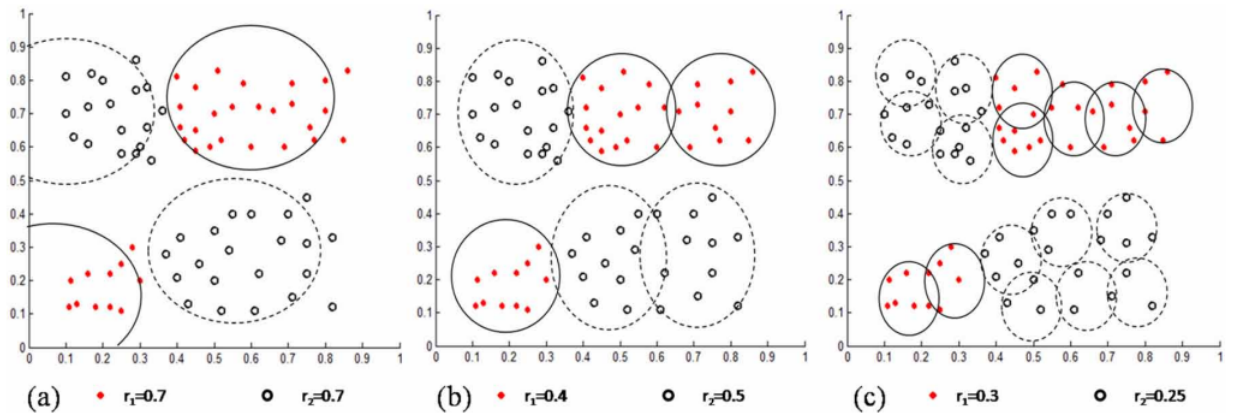


FIGURE 4.18 – Clustering soustractive avec différentes valeurs de rayon

4.4.3.2 Processus

La méthode proposée repose essentiellement sur l'application du PSO pour déterminer les rayons optimaux permettant d'obtenir des bases de règles plus précises. Le clustering et l'extraction des règles sont alors considérés comme la procédure d'évaluation du PSO pour les valeurs de rayon données, cette procédure est la suivante :

• Procédure de construction de la base de règles

Le soustractive Clustering est d'abord appliqué à chaque classe séparément, puis les fonctions d'appartenance sont définies par la projection des clusters obtenus sur chaque dimension de l'espace de caractéristique. Nous utilisons des fonctions gaussiennes et nous définissons leurs paramètres en considérant les centres des clusters comme leurs centres et les valeurs sigma, en spécifiant l'espace d'influence des clusters, comme leurs largeurs. Les fonctions d'appartenance gaussiennes sont données par :

$$\mu_{ji}(x) = \exp \left[- \left(\frac{x - c_{ji}}{2\sigma} \right)^2 \right] \quad (4.4.22)$$

Où c_{ji} et σ_{ji} sont, respectivement, le centre et l'écart-type des gaussiens qui correspond à la $j^{\text{ème}}$ règle et la $i^{\text{ème}}$ entrée.

Les fonctions d'appartenance calculées sont alors utilisées dans la génération des règles floues. Chaque cluster obtenu est représenté par une règle de la forme (2) expliquée dans la sous-section(4.2.4.1). Pour déterminer les parties conséquentes on se base sur les étapes présentées dans la sous-section (4.2.4.2).

• **La procédure principale** Dans le processus principal, le PSO est utilisé pour définir les rayons optimaux afin de construire une base de règles plus précise avec un nombre minimum de règles. Pour cette tâche, nous proposons d'utiliser une fonction multi-objectif similaire à la fonction de coût utilisée par Ishibuchi et Murata [52] pour l'algorithme génétique. La fonction d'objectif proposée, à la $t^{ième}$ itération, est :

$$fitness(t) = m_{cp}.CP(t) - m_r.R(t) \tag{4.4.23}$$

Où m_{cp} et m_r sont deux poids constants positifs, $CP(t)$ est le nombre des échantillons classifiés correctement, et $R(t)$ est le nombre de règles. On maximise le $CP(t)$ et on minimise le $R(t)$.

Le processus d'apprentissage de la méthode proposée est illustré à la figure 4.19. Le système proposé se compose de deux parties : Générateur de la base de règles et optimiseur des rayons (PSO).

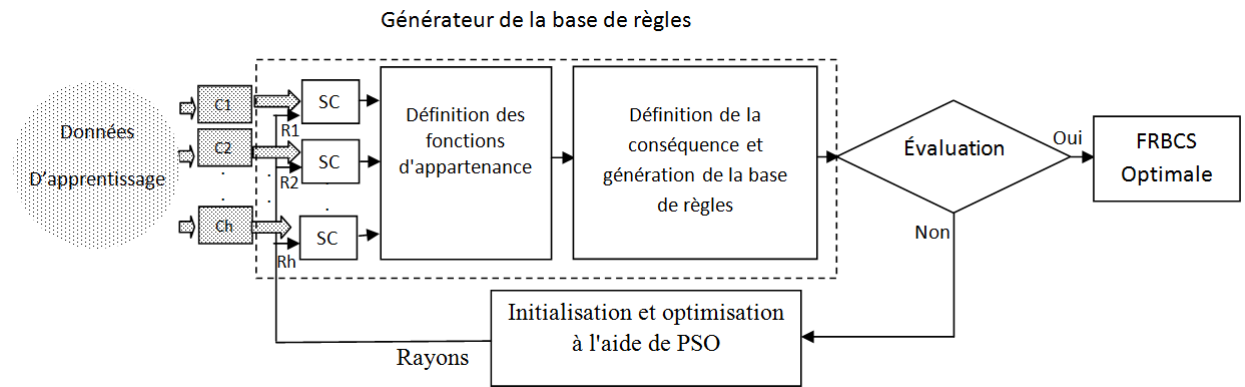


FIGURE 4.19 – Le schéma général du 3^{ème} modèle

Les étapes suivantes donnent une explication détaillée du processus proposé :

Étape 1 : Initialisation des paramètres PSO :

Ils sont initialisés de façon aléatoire. La position est un vecteur de rayons, dans lequel chaque rayon représente une classe.

Étape 2 : Construction de la base de règles :

Cette étape est décrite dans la sous-section précédente (4.2.4.2). Cela consiste en : Clustering par SC, Définition de la fonction d'appartenance, Définition des parties conséquentes des règles.

Étape 3 : Évaluation du système :

La base de règles obtenue est utilisée pour classer les exemples d'apprentissage et la fonction fitness (eq.4.7.18) est calculée. Si le critère d'arrêt (erreur minimale ou nombre d'itérations) est satisfait : Fin. Sinon, passez à l'étape suivante.

Étape 4 : Optimisation par PSO :

Cette étape se compose de deux parties :

- Définissez la meilleure position (p_{best}) de chaque particule et la meilleure position globale (g_{best}).
- Mise à jour de la vitesse et de la position de chaque particule selon les équations (4.2.3) et (4.2.4) respectivement.

Revenez à l'étape 2 pour mettre à jour la base de règles, utilisez la valeur de position (vecteur de rayons) dans la génération des règles floues, et définissez (la fitness).

4.4.3.3 Test et résultat

Dans cette sous-section, nous présentons les résultats expérimentaux du 3^{ème} modèle proposé, qui constitue notre contribution principale, sur différents bases de données avec attributs numériques de la machine d'apprentissage UCI (from the UCI machine learning) <http://archive.ics.uci.edu/ml> et le dépôt de données KEEL-Dataset : <http://sci2s.ugr.es/keel/datasets.php>. Pour une évaluation crédible, nous avons comparé notre méthode avec les travaux les plus similaires. Ainsi, nous avons considéré des approches floues basées sur l'évolution et nous

utilisons les mêmes ensembles de données impliqués dans ces travaux. Ces ensembles de données présentent une variété de classes (de 2 à 11), de caractéristiques (de 3 à 36) et d'échantillons (de 150 à 6435). Nous utilisons ces ensembles de données dans leur format original en tant qu'entrées dans le système proposé. Il s'agit notamment de : Iris, Tae, Wine, Glass, Pima, Bupa, Haberman, Newthyroid and Satimage dataset.

Leur spécification en termes de nombre d'attributs, des exemples et de classes de sortie est indiquée dans le tableau 4.11 :

Base de données	Nb d'attributs	Nb d'exemples	Nb de classes
Iris	4	150	3
Tae	5	151	3
Wine	13	178	3
Glass	9	214	7
Pima	8	768	2
Bupa	6	345	2
Haberman	3	306	2
Newthyroid	5	215	3
Satimage	36	6435	7
Vowel	13	990	11
Yeast	8	1484	10
Segment	19	2310	7

TABLE 4.11 – Liste des ensembles de données utilisés et leurs spécifications

Dans le premier test, nous utilisons une validation croisée d'ordre dix (10 – *CV*). Chaque ensemble de données est divisé en dix sous-ensembles, neuf sous-ensembles pour l'apprentissage et un sous-ensemble pour le test. Cette procédure est répétée dix fois pour que le FRBCS soit évalué à l'aide de tous les exemples.

Nous avons effectué des tests sur les ensembles de données ci-dessus où m_{cp} et m_r d'éq. (4.4.23) sont généralement égales à 0,7 et 0,3 respectivement.

Il est à noter que le PSO n'est pas sensible à la taille de la population [99]. Dans notre travail, nous avons utilisé des populations de 25 particules et un nombre maximal d'itérations égal à 200 était suffisant pour la convergence dans tous les tests. Nous avons comparé nos résultats avec sept approches similaires. Les méthodes utilisées dans la comparaison sont présentées dans le tableau 4.12.

Méthode	Titre	Auteur et référence
(1)	Design of Accurate Classifiers With a Compact Fuzzy-Rule Base Using an Evolutionary Scatter Partition of Feature Space	Ho et al., 2004 [49]
(2)	Constructing Accurate Fuzzy Rule-based Classification Systems Using A priori Principles and Rule-weighting	Fakhrmahmad et al., 2007 [36]
(3)	SGERD : A steady-state genetic algorithm for extracting fuzzy classification rules from data.	Mansoori et al., 2008 [72]
(4)	GPFIS-CLASS : A Genetic Fuzzy System based on Genetic Programming for classification problems	Koshiyama et al., 2015 [64]
(5)	Enhancing evolutionary fuzzy systems for multi-class problems : Distance-based relative competence weighting with truncated confidences (DRCW-TC)	Fernandez et al., 2016 [?]
(6)	An Integer Programming Approach for Fuzzy Rule-Based Classification Systems	Shahab and Alice, 2017 [97]
(7)	Inductive Learning of Fuzzy Rule-Based Classifier with Self-Constructing Clustering	Lee et al., 2017 [66]

TABLE 4.12 – Les approches utilisées pour la comparaison

Le tableau 4.13 illustre nos résultats et ceux des approches citées dans le tableau 4.12. Les résultats de chaque base sont les résultats moyens des 10 exécutions. Les résultats les plus élevés sont en gras.

On peut voir que le meilleur taux obtenu pour toutes les données n'est pas trouvé par une approche unique, mais que chaque approche a un degré d'efficacité différent d'une base à

	Proposé	(1)	(2)	(3)	(4)	(5)FARC-HD-v	(6)	
Iris	98.66	95.09	96.4	96.93	95.56	95.33	96.67	96.22
Tae	97.71	93.7	94.9	96.19	95.63	96.07	95.51	96.53
Wine	61	/	/	/	58.43	57.85	57.62	/
Glass	78.59	62.19	66.7	63.38	67.58	70.95	70.56	72.15
Pima	68.09	75.19	85.7	74.64	75.24	/	74.48	/
Bupa	65.8	/	62.2	/	68.39	/	/	/
Haberman	75.13	73.90	/	/	74.15	/	74.84	/
Newthyroid	96.73	93.97	/	/	95.23	97.16	95.35	/
Satimage	99.39	/	/	/	82.65	87.99	/	/
Vowel	96.76	/	/	58.53	/	99.09	/	75.65
Yeast	50.66	/	/	56.53	/	62.15	/	62.10
Segment	96.80	/	/	84.94	88.36	96.84	93.85	/

TABLE 4.13 – Résultats de la classification (précision%) à l'aide de 10 – CV

une autre.

Nos résultats montrent des améliorations en ce qui concerne les résultats de classification dans la majorité des bases de données à l'exception de Bupa et Pima qui ne possèdent que deux classes. Cela peut s'expliquer par le fait que le système proposé est plus efficace pour les problèmes avec un grand nombre de classes : PSO optimise plus de paramètres dans ces problèmes.

Afin de faire plus de comparaisons, c'est-à-dire en tenant compte à la fois de la précision et du nombre de règles, nous avons effectué un deuxième test en utilisant la validation croisée d'ordre deux (2 – CV). Cette fois-ci, chaque ensemble de données est divisé en deux sous-ensembles, un pour l'apprentissage et l'autre pour le test. Cette procédure est répétée deux fois.

	Méthode proposée		(chen et al.,2016) [22]	
	taux	Nb de règles	taux	Nb de règles
Iris	98.66	8.5	93.02	22.38
Tae	96.06	49.5	/	/
Wine	57.63	25.5	/	/
Glass	61.21	21	60.67	23.98
Pima	74.34	15.5	73.19	36.32
Bupa	64.05	38.5	/	/
Haberman	75.81	17	74.18	13.47
Newthyroid	96.75	12	91.13	16.88
Satimage	99.14	63.5	/	/

TABLE 4.14 – Résultats de la classification (précision % et nombre de règles) à l'aide de $2 - CV$

Le tableau 4.14 montre le taux de classification obtenu et le nombre de règles pour chaque méthode. Les résultats sont comparés avec (Chen et al., 2016) [22] qui utilise l'optimisation par l'essai de particules dans la recherche d'un ensemble de poids de règles optimales.

Les résultats obtenus montrent que la méthode proposée est supérieure à l'autre méthode en termes de précision et de nombre de règles.

Nous complétons l'analyse en effectuant un test d'hypothèse statistique non paramétrique pour la comparaison de deux classificateurs. Nous utilisons le test Wilcoxon Signed-Ranks Test, qui est un test de comparaison par paires pour déterminer l'existence d'une différence significative de performance entre deux classificateurs. Considérons R^+ la somme des rangs pour les fonctions où le premier algorithme dépasse le second et R^- la somme des rangs où le second algorithme dépasse le premier.

R^+ et R^- sont calculés par eq. (4.7.24) et éq. (4.7.25) comme suit [31] :

$$R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i) \quad (4.4.24)$$

$$R^- = \sum_{d_i < 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i) \quad (4.4.25)$$

Où d_i est la différence entre les scores de performance des deux algorithmes sur $i^{\text{ème}}$ parmi N cas. Pour chaque comparaison, nous considérons les méthodes avec les résultats disponibles pour chaque ensemble de données.

Proposé vs.	R^+	R^-	valeur-p
(1)	16.0	5.0	0.208413
(2)	11.0	4.0	0.280713
(3)	21.0	7.0	0.204894
(4)	43.0	12.0	0.102918
(5)	30.0	15.0	0.343253
(6)	29.0	7.0	0.107328
(7)	11.0	4.0	0.280713

TABLE 4.15 – Test de Wilcoxon pour comparer l’approche proposée [R^+] et les autres méthodes [R^-] en 10 – CV.

Proposé vs.	R^+	R^-	valeur-p
(Chen et al., 2016) [22]	15.0	0.0	0.030971

TABLE 4.16 – Test de Wilcoxon pour comparer l’approche proposée [R^+] et la méthode de Chen et al (2016) [22] [R^-] en 2 – CV

Les tableaux 4.15. et 4.16. présentent l’analyse statistique réalisée à l’aide du test de Wilcoxon Signed-Ranks. En tenant compte de la précision dans le test, les meilleures valeurs sont indiquées en caractères gras. Ces tableaux montrent que l’algorithme que nous

proposons est statistiquement meilleur que les autres méthodes utilisées dans la comparaison.

La complexité temporelle de la méthode proposée est faible et elle augmente linéairement avec la dimension du problème et le nombre de clusters. Ceci s'explique par les trois points suivants. Premièrement, notre méthode est basée sur le clustering, qui génère moins de règles que les approches basées sur des partitions en grille. En effet, dans les méthodes de partitions en grille, l'espace des entités est divisé en toutes les régions possibles et chacune d'entre elles est représentée par une règle floue. Par conséquent, un nombre exponentiellement croissant de règles sont générées. Deuxièmement, les avantages du clustering soustractive par rapport aux autres méthodes de clustering en termes de complexité temporelle résident dans le fait qu'il n'implique aucune optimisation itérative non linéaire et que le calcul n'augmente que linéairement avec la dimension du problème Chiu (1994). Troisièmement, PSO est un algorithme d'optimisation simple avec seulement deux équations dans chaque itération. Par exemple, par rapport aux algorithmes génétiques, le PSO est beaucoup plus simple puisqu'il ne comporte ni opérations de croisement ni opérations de mutation. Dans notre travail, le nombre de paramètres optimisés utilisant PSO est égal au nombre de classes.

4.5 Conclusion

Dans ce chapitre, nous avons effectué une étude analytique sur l'effet des poids des règles et des fonctions d'appartenance, puis proposé trois modèles de classification. Dans le premier modèle nous avons optimisé les fonctions d'appartenance par fusion. Dans le deuxième modèle nous nous sommes basés sur le clustering pour générer les règles floues et les métaheuristiques bio-inspirées pour l'optimisation des poids des règles. Dans ce modèle nous avons utilisée l'optimisation par essaim de particules (PSO) et l'algorithme de recherche coucou (cuckoo search). Finalement, dans le troisième modèle, qui constitue notre principale contribution, nous avons optimisé et déterminé les valeurs adéquates des

paramètres d'entrée de la méthode de clustering en utilisant le PSO, afin de déterminer un nombre optimal de règles.

D'après les résultats obtenus nous pouvons déduire les points suivant :

- les fonctions d'appartenance et les poids des règles influencent sur les performances de la classification basée sur les règles floues.
- Le fusionnement des fonctions d'appartenance a amélioré les performances de classification et réduit le nombre de règles floues.
- L'utilisation des méthodes bio inspirées pour le calcul des poids des règles a montré son efficacité.
- L'Application du soustractive clustering sur chaque classe séparément et l'utilisation du PSO pour optimiser les rayons donne des résultats satisfaisants.
- Les systèmes proposés ont démontré avec succès leur capacité à générer les fonctions d'appartenance et les degrés de certitude.

CONCLUSION

Cette thèse s'est attachée à la classification floue qui a prouvée son efficacité face aux données imprécises, contradictoires et ambiguës. Notre objectif était de développer des classificateurs à base des règles floues qui permettent un bon compromis précision-compacité. Nous avons ainsi proposé trois modèles qui se basent essentiellement sur les méthodes de clustering et les méthodes d'optimisation bio-inspirées.

D'abord nous avons effectué une étude théorique sur l'effet des fonctions d'appartenance et les poids des règles sur les performances du classificateur à base de règles floues. Les tests de cette étude ont été effectués sur des problèmes synthétiques bidimensionnelles, et sur les bases de données Iris et Wine.

Nous avons ensuite proposé trois modèles :

Le premier modèle consiste à utiliser le soustractive clustering pour la génération des fonctions d'appartenance et à utiliser un algorithme de fusionnement afin de réduire le nombre des fonctions d'appartenance et le nombre des règles floues.

Dans le deuxième modèle nous avons utilisé aussi le soustractive clustering pour l'initialisation des fonctions d'appartenance. Pour le calcul des poids des règles, nous avons utilisé deux méthodes d'optimisation bio-inspirées, deux versions de ce modèle ont été ainsi proposées. La première version est basée sur l'optimisation par essaim de particules. Cette version a été évaluée sur les bases de données Iris et Wine. Quant la deuxième version, elle est basée sur l'algorithme de recherche coucou. Cette version a été utilisée pour

le diagnostic médical et appliquée sur la base de données du cancer du sein (wisconsin). Pour le troisième modèle, qui constitue notre contribution principale, l'idée principale se base sur le fait que la méthode de soustractive clustering dépend du rayon, dont sa valeur affecte le nombre de clusters obtenus. Généralement, ce paramètre est prédéfini par l'utilisateur. Le principe de ce modèle repose essentiellement sur la détermination automatique des valeurs optimales de ce paramètre avec le PSO. Pour obtenir un bon compromis précision-compacité, nous avons proposé d'utiliser une fonction multi-objective qui minimise le nombre de règles et maximise le taux de classification. Ce modèle a été testé sur les bases de données Iris, Tae, Wine, Glass, Pima, Bupa, Haberman, Newthyroid et Satimage. Après avoir implémenter nos modèles et les tester sur les différentes bases de données, ils ont permis d'avoir des solutions satisfaisantes dans la plupart des cas. En les comparant avec d'autres méthodes récentes, ils ont montré leur compétitivité en termes de précision et de nombre de règles.

Les résultats obtenus nous ont permis de dresser les perspectives de recherche suivantes :

- Proposer d'autres méthodes pour le fusionnement des fonctions d'appartenance.
- Utiliser d'autres méthodes d'optimisation bio-inspirés et de clustering afin d'améliorer davantage l'efficacité, la simplicité et la compréhensivité des classificateurs flous.
- Introduire des Systèmes d'Aide au Diagnostic Médical basés sur la classification floue et bénéficier des avantages des systèmes flous, notamment de leur compréhensivité et de leur capacité à gérer les données imprécises.

Bibliographie

- [1] Abadeh, M. S., Habibi, J., et Soroush, E. (2007, March). *Induction of fuzzy classification systems using evolutionary ACO-based algorithms*. In First Asia International Conference on Modelling & Simulation (AMS'07) (pp. 346-351). IEEE.
- [2] Abadeh, M. S., Habibi, J., et Soroush, E. (2008). Induction of fuzzy classification systems via evolutionary ACO-based algorithms. *computer*, 35, 37.
- [3] Abe, S. et Lan, M. S. (1995). *A method for fuzzy rules extraction directly from numerical data and its application to pattern recognition*. IEEE Transaction on Fuzzy Systems, 3(1), p. 18-28.
- [4] Abe, S. (2012). *Pattern classification : neuro-fuzzy methods and their comparison*. Springer Science & Business Media.
- [5] Aguilera, J. J., Chica, M., Del Jesus, M. J., et Herrera, F. (2007, July). *Niching genetic feature selection algorithms applied to the design of fuzzy rule-based classification systems*. In 2007 IEEE International Fuzzy Systems Conference (pp. 1-6). IEEE.

- [6] Alikar, N., Abdullah, S., Mousavi, S. M., et Niaki, S. T. A. (2013). *A hybrid particle swarm optimization and fuzzy rule-based system for breast cancer diagnosis*. *Int J Soft Comput*, 8(2), 126-133.
- [7] Alves, R. T., Delgado, M. R., Lopes, H. S., et Freitas, A. A. (2004, September). *An Artificial Immune System for Fuzzy-Rule Induction in Data Mining*, In proceeding of : Parallel Problem Solving from Nature-PPSN VIII, 8th International Conference (p.1011-1020). Birmingham, UK.
- [8] Antonelli, M., Ducange, P. et Marcelloni, F. (2014). *A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers*, *New Trend of Computational Intelligence in Human-Robot Interaction*, 283, p. 36-54.
- [9] Aydogan, E. K., Karaoglan, I., et Pardalos, P. M. (2012). *hGA : Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems*. *Applied Soft Computing*, 12(2), 800-806.
- [10] Azeem, M., Usman, M., et Fong, A. C. M. (2017). *A churn prediction model for prepaid customers in telecom using fuzzy classifiers*. *Telecommunication Systems*, 66(4), 603-614.
- [11] Berlanga, F. J., del Jesus, M. J., et Herrera, F. (2005). *Learning compact fuzzy rule-based classification systems with genetic programming*. In *EUSFLAT Conf.* (pp. 1027-1032).
- [12] Bezdek, J.(1980) . *A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2, p. 1-8,
- [13] Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.

- [14] Binitha, S., et Sathya, S. S. (2012). *A survey of bio inspired optimization algorithms*. International journal of soft computing and engineering, 2(2), 137-151.
- [15] Cavallaro, F., (2015), A Takagi-Sugeno fuzzy inference system for developing a sustainability index of biomass, Sustainability, 7 (9), 12359-12371.
- [16] Chao, W. (2011). *Machine learning tutorial*. Digital Image and Signal Processing.
- [17] Chaudhuri, A., et De, K. (2011). *Fuzzy support vector machine for bankruptcy prediction*. Applied Soft Computing, 11(2), 2472-2486.
- [18] Chen, C. C. (2006). *Design of PSO-based Fuzzy Classification Systems* , Tamkang Journal of Science and Engineering, 9(1), p. 63-70.
- [19] Chen, Ch., Li, Sh., Liu, Ta., Chen, K., et Wong, Ch., (2011). *A Clustering-based algorithm to extracting fuzzy rules for system modeling*. International Journal of Advancements in Computing Technology (IJACT), 3(11), p. 394-401.
- [20] Chen, G., Pham, T. T., et Boustany, N. M. (2001). *Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems*. Appl. Mech. Rev., 54(6), B102-B103.
- [21] Chen, L., Yan, J., et He, Y.(1995). *Automatic generation of fuzzy set membership functions and rules using a neural network*, Proceedings of the Third Australian and New Zealand Conference on Intelligent Information Systems (pp. 164 - 168). ANZIIS-95.
- [22] Chen, T., Shen, Q., Su, P., et Shang, Ch. (2016). *Fuzzy rule weight modification with particle swarm optimization*, Soft Computing, 20(8),p. 2923-2937.
- [23] Chiu, S. L. (1994). *Fuzzy model identification based on cluster estimation*, Journal of Intelligent and Fuzzy Systems -JIFS, 2,p. 267-278.

- [24] Chiu, S. L. (1997). *Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification*, Chapter 9 in *Fuzzy Information Engineering : A Guided Tour of Applications*, D. Dubois et al. (eds.) (pp. 149-162). Springer, Berlin.
- [25] Chopra, S., Mitra, R., et Kumar, V. (2004). *Identification of rules using subtractive clustering with application to fuzzy controllers*, Proceedings of the Third International Conference on Machine Learning and Cybernetics (pp. 26-29). Shanghai.
- [26] Chu, S. C., Huang, H. C., Roddick, J. F., et Pan, J. S. (2011, September). *Overview of algorithms for swarm intelligence*. In *International Conference on Computational Collective Intelligence* (pp. 28-41). Springer, Berlin, Heidelberg.
- [27] Cordon, O., Del Jesus, M. J., et Herrera, F. (1998). *Genetic Learning of Fuzzy Rule-Based Classification Systems Cooperating with Fuzzy Reasoning Methods*, *International Journal of Intelligent Systems*, 13(10-11),p. 1025-1053.
- [28] Cordon, O., Jesus, M.J. Del, Herrera, F. (1999) : *A Proposal on Reasoning Methods in Fuzzy Rule-based Classification Systems*. *International Journal of Approximate Reasoning*, 20, 21-45
- [29] Cordon, O., Del Jesus, M. J., Herrera, F., Magdalena, L., et Villar, P. (2003). *A Multiobjective Genetic Learning Process for joint Feature Selection and Granularity and Contexts Learning in Fuzzy Rule-Based Classification Systems*, J. Casillas et al. (eds.), *Interpretability Issues in Fuzzy Modeling* Springer-Verlag Berlin Heidelberg, 128, p. 79-99.
- [30] Cornelissen, A.M.G. ; van den Berg, J. ; Koops, W.J. ; Grossman, M. ; Udo, H.M.J.(2001) *Assessment of the contribution of sustainability indicators to sustainable development : A novel approach using fuzzy set theory*. *Agr. Eco. Env. ,* 86,p. 173-185

- [31] Demsar, J. (2006). *Statistical Comparisons of Classifiers over Multiple Data Sets*, Journal of Machine Learning Research, 7,p. 1-30.
- [32] Dubois, D., Ostasiewicz, W., et Prade, H. (2000). Fuzzy sets : history and basic notions. In *Fundamentals of fuzzy sets* (pp. 21-124). Springer, Boston, MA.
- [33] Duda, R. O., Hart, P. E. et Stork, D. G. (2001). *Pattern classification*, New York, NY, Wiley & Sons.
- [34] Dorigo, M., Maniezzo, V., et Colomi, A. (1996). *Ant system : optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29-41.
- [35] Eskofier, B. M. (2010). *Application of pattern recognition methods in biomechanics*. University of Calgary.
- [36] Fakhrahmad, S. M., Zare, A.,et Jahromi, M. Z. (2007). *Constructing Accurate Fuzzy Rule-based Classification Systems Using Apriori Principles and Rule-weighting*, Intelligent Data Engineering and Automated Learning - IDEAL 2007, 4881,p. 547-556.
- [37] Fernández, A., Elkano, M., Galar, M., Sanz, J. A., Alshomrani, S., Bustince, H.,et Herrera, F. (2016). *Enhancing evolutionary fuzzy systems for multi-class problems : Distance-based relative competence weighting with truncated confidences (DRCW-TC)*, International Journal of Approximate Reasoning, 73, p.108-122.
- [38] Fazzolari, M., Giglio, B., Alcalá, R., Marcelloni, F., et Herrera, F. (2013). *A study on the application of instance selection techniques in genetic fuzzy rule-based classification systems : Accuracy-complexity trade-off*. Knowledge-Based Systems, 54, 32-41.

- [39] Forgy, E. (1965). *Cluster Analysis of Multivariate Data : Efficiency versus Interpretability of Classification*. *Biometrics*, 21, p. 768-769
- [40] Ganeshkumar, P., Rani, C., et Deepa, S.N. (2013), *Formation of fuzzy if-then rules and membership function using enhanced particle swarm optimization*, *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 21(1), 103-126.
- [41] Ganji, M. F., et Abadeh, M. S. (2010, May). *Using fuzzy ant colony optimization for diagnosis of diabetes disease*. In 2010 18th Iranian Conference on Electrical Engineering (pp. 501-505). IEEE.
- [42] Ganji, M. F., et Abadeh, M. S. (2010). *Rule Based Classification System for Medical Data Mining Using Fuzzy Ant Colony Optimization*. In *Proceedings of the World Congress on Engineering and Computer Science (Vol. 1)*.
- [43] Ganji, M. F., et Abadeh, M. S. (2011). *A fuzzy classification system based on Ant Colony Optimization for diabetes disease diagnosis*, *Expert Systems with Applications*, 38 (12), p. 14650-14659.
- [44] Garrett, S. M. (2005). *How do we evaluate artificial immune systems*. *Evolutionary computation*, 13(2), 145-177.
- [45] Ghosh, S., Biswas, S., Chanda, D., et Sarkar, P. P. (2016). *A Neuro-fuzzy based Classification Method*. *Indian Journal of Science and Technology*, 9, 14.
- [46] Giglio, B., Marcelloni, F., Fazzolari, M., Alcalá, R., et Herrera, F. (2012, June). *A case study on the application of instance selection techniques for genetic fuzzy rule-based classifiers*. In 2012 IEEE International Conference on Fuzzy Systems (pp. 1-8). IEEE.
- [47] Gomes, M. S. M. M. (2009). *Reducing the number of membership functions in linguistic variables* (Doctoral dissertation, Faculdade de Ciências e Tecnologia).

- [48] Haddadnia, J., Faez, K., et Ahmadi, M. (2003). *A fuzzy hybrid learning algorithm for radial basis function neural network with application in human face recognition*. *Pattern Recognition*, 36(5), 1187-1202.
- [49] Ho, Sh. Y., Chen, H. M., Ho, Sh. J., et Chen, T. K. (2004). *Design of Accurate Classifiers With a Compact Fuzzy-Rule Base Using an Evolutionary Scatter Partition of Feature Space*, *IEEE Transactions On Systems, MAN, and Cybernetics_Part B : Cybernetics*, 34(2), p. 1031-1044.
- [50] Imran, M., et Alsuhaibani, S. A. (2019). *A Neuro-Fuzzy Inference Model for Diabetic Retinopathy Classification*. In *Intelligent Data Analysis for Biomedical Applications* (pp. 147-172). Academic Press.
- [51] Ishibuchi, H., Murata, T., et Türksen, I. B. (1997). *Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems*. *Fuzzy sets and systems*, 89(2), 135-150.
- [52] Ishibuchi, H., et Murata, T. (1998). *Multi-Objective Genetic Local Search for Minimizing the Number of Fuzzy Rules for Pattern Classification Problems*, *The 1998 IEEE International Conference on Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence*, (pp.1100-1105).4-9 May 1998, Anchorage, AK,2.
- [53] Ishibuchi, H., Nozaki, K., et Tanaka, H. (1992). *Distributed representation of fuzzy rules and its application to pattern classification*, *Fuzzy Sets and System*. 52(1),p. 21-32.
- [54] Ishibuchi, H., et Nakashima, T. (2001). *Effect of rule weights in fuzzy rule-based classification systems*. *IEEE Transactions on Fuzzy Systems*, 9(4), 506-515.

- [55] Ishibuchi, H., Nakashima, T., et Nii, M. (2004). *Classification and modeling with linguistic information granules : Advanced approaches to linguistic Data Mining*. Springer Science & Business Media.
- [56] Ishibuchi, H., et Yamamoto, T. (2005). *Rule weight specification in fuzzy rule-based classification systems*. IEEE Transactions on Fuzzy Systems, 13(4), p. 428-435.
- [57] Jamsandekar, S. S., et Mudholkar, R. R. (2014). *Fuzzy Classification System by Self Generated Membership Function Using Clustering Technique*, International Journal of Information Technology, 6(1),p. 697-704.
- [58] Jayachandran, A., et Dhanasekaran, R. (2017). *Multi class brain tumor classification of MRI images using hybrid structure descriptor and fuzzy logic based RBF kernel SVM*. Iranian Journal of Fuzzy Systems, 14(3), 41-54.
- [59] Kahali,S. Adhikari,S.K. et Sing,J.K. (2017). *A two-stage fuzzy multi-objective framework for segmentation of 3D MRI brain image data*, Applied Soft Computing, 60, p.312-327.
- [60] Kamaloo, E., et Abadeh, M. S. (2010, July). An artificial immune system for extracting fuzzy rules in credit scoring. In IEEE Congress on Evolutionary Computation (pp. 1-8). IEEE.
- [61] Kashanipour, A., Milani, N. S., Kashanipour, A. R., et Eghrary, H. H. (2008). *Robust Color Classification Using Fuzzy Rule Based Particle Swarm Optimization*, IEEE, Congress on Image and Signal Processing (CISP 08), Sanya, China, p. 110-114.
- [62] Kaufmann, A. (1975). *Introduction to the theory of fuzzy subsets* (Vol. 2). Academic Pr.

- [63] Kennedy, J., et Eberhart, R.C. (1995). *Particle swarm optimization*, in : Proceedings of IEEE International Conference on Neural Networks (pp. 1942-1948). Perth, Australia.
- [64] Koshiyama, A. S., Vellasco, M. M. B. R., et Tanscheit, R. (2015). *GPFIS-CLASS : A Genetic Fuzzy System based on Genetic Programming for classification problems*, Applied Soft Computing, 37,p. 561-571.
- [65] Lafta, H. A., et Mohammed, Z. A. (2016). *Optimization of Membership Function of Fuzzy Rules Generated using Subtractive Clustering* .International Journal of Current Engineering and Technology, 6(3),p. 821-824.
- [66] Lee, C. H., Wang, C. R., Su, Y. Y., et Lee, S. J. (2017). *Inductive Learning of Fuzzy Rule-Based Classifier with Self-Constructing Clustering*. International Journal of Engineering Research, 6(2), p. 55-60.
- [67] Lei, Z., et Ren-hou, L. (2008). *Designing of classifiers based on immune principles and fuzzy rules*, Information Sciences, 178(7),p. 1836-1847.
- [68] Leung, Y., Zhang, J., et Xu, Z.(2000) . *Clustering by Space-Space Filtering*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12), p. 1396-1410.
- [69] Mahata,N. Kahali,S. Adhikari, S.K. et Sing,J.K. (2018). *Local contextual information and Gaussian function induced fuzzy clustering algorithm for brain MR image segmentation and intensity in homogeneity estimation*, Applied Soft Computing, 68, p. 586-596.
- [70] Mahmoud, M. S. (2018). *Fuzzy control, estimation and diagnosis*. Saudi Arabia : Springer International.

- [71] Mamdani, E. H., et Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1), 1-13.
- [72] Mansoori, E. G., Zolghadri, M. J., et Katebi, S. D. (2008). *SGERD : A steady-state genetic algorithm for extracting fuzzy classification rules from data*. *IEEE Transactions on Fuzzy Systems*, 16(4),p. 1061-1071.
- [73] Mekhmoukh, A., et Mokrani, K. (2015). *Improved Fuzzy C-Means based Particle Swarm Optimization (PSO) initialization and outlier rejection with level set methods for MR brain image segmentation*, *Computer Method and Programs in Biomedicine*, 122,p. 266-281.
- [74] Mendes, R. R., de Voznika, F. B., Freitas, A. A., et Nievola, J. C. (2001, September). *Discovering fuzzy classification rules with genetic programming and co-evolution*. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 314-325). Springer, Berlin, Heidelberg.
- [75] Mittal, M., et Kaur, G. (2013) *Mixels Resolution by hybridization approach (BBO & GA)*, *International Journal of Innovative Research in Science, Engineering and Technology*, 2(10).
- [76] Mohammadpour, R. A., Abedi, S. M., Bagheri, S., et Ghaemian, A. (2015). *Fuzzy rule-based classification system for assessing coronary artery disease*. *Computational and mathematical methods in medicine*, 2015.
- [77] Nemissi, M., Seridi, H., et Akdag, H. (2014). *One-against-all and one-against-one based neuro-fuzzy classifiers*. *Journal of Intelligent & Fuzzy Systems*, 26(6),p. 2661-2670.

- [78] Nilashi, M., Ibrahim, O., Ahmadi, H., et Shahmoradi, L. (2017). *A knowledge-based system for breast cancer classification using fuzzy logic method*. Telematics and Informatics, 34(4), 133-144.
- [79] Nozaki, K., Ishibuchi, H., et Tanaka, H. (1997). *A simple but powerful heuristic method for generating fuzzy rules from numerical data*. Fuzzy Sets and Systems, 86,p. 251-270.
- [80] Omran, M. G. (2006). *Particle swarm optimization methods for pattern recognition and image processing* (Doctoral dissertation, University of Pretoria).
- [81] Pereira, R., Fagundesb, A., Melício, R., Mendes, V. M. F., Figueiredo, J., et Quadrado, J.C. (2014). *Fuzzy subtractive clustering technique applied to demand response in a smart grid scope*, Conference on Electronics, Telecommunications and Computers - CETC, Procedia Technology, 17,p. 478 - 486.
- [82] Permana, K. E., et Hashim, S. Z. M. (2010). *Fuzzy Membership Function Generation using Particle Swarm Optimization*, International Journal of Open Problems in Computer Science and Mathematics,3(1), 58-61.
- [83] Prabhat, N. (2011). *A genetic-fuzzy algorithm to discover fuzzy classification rules for mixed attributes datasets*. International Journal of Computer Applications, 34(5), 15-22.
- [84] Pranevicius, H., Kraujalis, T., Budnikas, G.,et Pilkauskas, V. (2014). *Fuzzy Rule Base Generation Using Discretization of Membership Functions and Neural Network*, International Conference on Information and Software Technologies ,p. 160-171.
- [85] Qu, Y., Shen, Q., Mac Parthaláin, N., Shang, C., et Wu, W. (2013). *Fuzzy similarity-based nearest-neighbour classification as alternatives to their fuzzy-*

- rough parallels*. International Journal of Approximate Reasoning, 54(1), 184-195.
- [86] Rania, C., et Deepa, S. N. (2010). PSO with mutation for fuzzy classifier design. *Procedia Computer Science*, 2, 307-313.
- [87] Rouhani, S. H., Sheikholeslami, A., Hosseini, H., et Kazemi, H. (2014). *Application of Fuzzy Subtractive Clustering for Optimal Transient Performance of Automatic Generation Control*, IETE Journal of Research, 59(6), p. 753-760.
- [88] Ross, T. J. (2004). *Fuzzy Logic with Engineering Applications*, John Wiley and Sons.
- [89] Saade, J. J., et Diab, H. B. (2004). *Defuzzification methods and new techniques for fuzzy controllers*, Iranian Journal of Electrical and Computer Engineering, 3(2), p.161-174.
- [90] Salah, H., Nemissi, M., Seridi, H. (2018). *Breast cancer classification using Fuzzy classifier*. Algerian Journal on Cancer Survey, 1(1), 6-6.
- [91] Salah, H., Nemissi, M., Seridi, H., et Akdag, H. (2019). *Subtractive Clustering and Particle Swarm Optimization Based Fuzzy Classifier*. International Journal of Fuzzy System Applications (IJFSA), 8(3), 108-122.
- [92] Saniee, M. S., Habibi, J., et Soroush, E. (2008). *Induction of Fuzzy Classification Systems via Evolutionary ACO-Based Algorithms*, International Journal of Simulation Systems, Science and Technology, 9(3), p. 1-8.
- [93] Sanz, J., Fernández, A., Bustince, H., et Herrera, F. (2010, July). *A genetic algorithm for tuning fuzzy rule-based classification systems with Interval-Valued Fuzzy Sets*. In International Conference on Fuzzy Systems (pp. 1-3). IEEE.

- [94] Sanz, J., Pagola, M., Bustince, H., Brugos, A., Fernández, A., et Herrera, F. (2011, April). *A case study on medical diagnosis of cardiovascular diseases using a genetic algorithm for tuning fuzzy rule-based classification systems with interval-valued fuzzy sets*. In 2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ) (pp. 9-15). IEEE.
- [95] Schaefer, G. (2012, September). GA-based optimisation of fuzzy rule bases for pattern classification. In 11th Symposium on Neural Network Applications in Electrical Engineering (pp. 139-141). IEEE.
- [96] Scherer, R. (2012). *Multiple fuzzy classification systems* (Vol. 288). Springer.
- [97] Shahab, D. et Alice, E.S. (2017). *An integer programming approach for fuzzy rule-based classification systems*, European Journal of Operational Research, 256(3), p. 924-934.
- [98] Sharma, P., et Kaur, M. (2013). *Classification in pattern recognition : A review*. International Journal of Advanced Research in Computer Science and Software Engineering, 3(4).
- [99] Shi, Y. H., et Eberhart, R. C. (1998). *A modified particle swarm optimizer*. IEEE International Conference on Evolutionary Computation, 4-9 May 1998, Anchorage, Alaska, p. 69-73.
- [100] Shi, B., et Li, G. (2006). U.S. Patent No. 7,146,037. Washington, DC : U.S. Patent and Trademark Office.
- [101] Song, B. G., Marks, R. J., Oh, S., Arabshahi, P., Caudell, T. P., et Choi, J. J. (1993, March). *Adaptive membership function fusion and annihilation in fuzzy if-then rules*. In [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems (pp. 961-967). IEEE.

- [102] Stavrakoudis, D. G., et Theocharis, J. B. (2010, March). *Employing Effective Feature Selection in Genetic Fuzzy Rule-Based Classification Systems*. In 2010 4th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS) (pp. 21-26). IEEE
- [103] Takagi, T., et Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1), 116-132.
- [104] Tettamanzi, A. G., Carlesi, M., Pannese, L., et Santalmasi, M. (2007, April). *Business intelligence for strategic marketing : Predictive modelling of customer behaviour using fuzzy logic and evolutionary algorithms*. In Workshops on Applications of Evolutionary Computation (pp. 233-240). Springer, Berlin, Heidelberg.
- [105] Theodoridis, S. et Koutroumbas, K. (2009). *Pattern recognition*, Amsterdam, Elsevier Academic Press
- [106] Trawinski, K., Cordon, O., Sanchez, L., et Quirin, A. (2013). *A genetic fuzzy linguistic combination method for fuzzy rule-based multiclassifiers*. *IEEE Transactions on Fuzzy Systems*, 21(5), 950-965.
- [107] Viattchenin, D. A., Tati, R., et Damaratski, A. (2013). *Designing Gaussian Membership Functions for Fuzzy Classifier Generated by Heuristic Possibilistic Clustering*, *Journal of Information and Organizational Sciences*, 37(2), p. 127-139.
- [108] Yager, R.R., et Filev, D.P. (1994). *Approximate clustering via the mountain method*, *Man and Cybernetics*, IEEE Transactions on Systems, 24, p. 1279-1284.
- [109] Yang, X. S., et Deb, S. (2009, December). *Cuckoo search via Lévy flights*. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). IEEE.

-
- [110] Yuan, Y., et Zhuang H. (1996). *A genetic algorithm for generating fuzzy classification rules*, Fuzzy Sets and Systems, 84(25), p. 1-19.
- [111] WEE, C. T. (2010). *Advances in fuzzy rule-based system for pattern classification*.
- [112] Werro, N. (2015). *Fuzzy classification of online customers*, (Vol. 148). Heidelberg : Springer.
- [113] Zadeh, L.A., (1965). *Fuzzy Sets*, Inf. Control, 8, p. 338-353 .
- [114] Zadeh, L.M., et Eftekhari, M. (2013). *Designing Fuzzy imbalanced classifier based on the subtractive clustering and Genetic Programming*, 13th Iranian Conference on Fuzzy Systems (IFSC), 27-29 Aug. 2013,p. 1-6.

Bibliographie
