

UNIVERSITE SAAD DAHLAB DE BLIDA

Faculté des Sciences de l'ingénieur
Département d'Informatique

MEMOIRE DE MAGISTER

Spécialité : Systèmes d'information et de connaissances

SYSTEME MULTI-AGENTS POUR LA CLASSIFICATION DYNAMIQUE FLOUE DE DONNEES MULTIMODALES ET EVOLUTIVES

Par

ABBAS Mohamed Amir

Devant le jury composé de

N. Benblidia	Maître de Conférences A, U. de Blida	Présidente
N. Bachari	Professeur, U.S.T.H.B, Alger	Examineur
S. Oukid Khouas	Maître de Conférences A, U. de Blida	Examineur
A. Moussaoui	Maître de Conférences A, U. de Sétif	Promoteur

Blida, Mai 2012

RESUME

Dans cette étude nous présentons une nouvelle approche pour la classification dynamique floue de données non-stationnaires et multimodales, basée sur une architecture multi-agents. Cette approche est adaptative à l'évolution des classes et de données, dans le but d'optimiser l'affectation des entités provenant de sources hétérogènes aux classes, et à renforcer le mécanisme de la classification incrémentale ainsi la détermination des cas de création de nouvelles classes. Nous proposons une approche de classification coopérative conçue avec divers niveaux de traitement de données, elle commence par une phase de collecte et de modélisation des différentes données, suivie par leur classification selon les deux méthodes supervisée et non-supervisée, et elle termine par un ensemble de processus d'interprétations et de fusion des résultats de classifications. Cette approche va permettre aux agents classificateurs de collaborer dans la prise des décisions finales. Elle a été implémentée sur la plateforme JADE, où chaque démarche sera assurée par des agents spécialisés collaborant et communiquant entre eux.

Mots-clés : Classification dynamique, Données multimodales et évolutives, Classes évolutives, Coopération, Système Multi-Agents.

ملخص

في هذا البحث سنعرض طريقة جديدة لتطبيق عملية التصنيف الديناميكي الغامض لبيانات غير ثابتة و متعددة الوسائط، مبنية على أساس نظام متعدد الوكيل. من أهم أهداف هذه الطريقة : التكيف مع التغيرات الدينامكية للبيانات و الأقسام (الفئات) من أجل تحسين توزيع الأفراد ذات المصادر الغير متجانسة على أقسامها (فئاتها)، بالإضافة إلى زيادة المقدرة على تحديد حالات إنشاء أقسام و فئات جديدة و بالتالي تعزيز آلية التصنيف الديناميكي. لقد تم إقتراح اتباع منهج التعاون في تصميم عملية التصنيف و ذلك بتقسيمها على عدة مراحل لمعالجة البيانات. في البداية نقوم بعملية جمع و نمذجة مختلف البيانات، تليها مباشرة مرحلة التصنيف و التي يتم فيها إستخدام كلا الطريقتين "بالإشراف" و "بدون إشراف"، و في النهاية تقوم مجموعة من وكلاء مختصة بعملية الإستنتاج و التفسير لنتائج تصنيف المرحلة السابقة. هذا التصميم جاء ليساعد وكلاء النظام في تسهيل عملية إتخاذ القرارات النهائية للمستخدمين. لقد تم تنفيذ و إختبار هذه الطريقة المقترحة بواسطة البرنامج JADE، حيث تم التمكن من تطبيق كل مرحلة على حدة و في نفس الوقت من طرف وكلاء مختصين يعملون و يتواصلون مع بعض.

الكلمات المفتاحية : التصنيف الديناميكي، بيانات غير ثابتة و متعددة الوسائط، أقسام متغيرة، منهج التعاون، نظام متعدد الوكيل.

ABSTRACT

In this study we present a new approach for fuzzy dynamic classification of multimodal and evolutionary data, based on a multi-agent architecture. This approach is adaptive with classes and data evolutions in order to optimize the allocation of entities from disparate sources into classes, and to strengthen the mechanism for incremental classification and to determinate a cases of creating new classes. We propose a cooperative approach of classification designed with various levels of data processing, it begins by collecting and modeling the different data, followed by classification using the two methods supervised and unsupervised, and it ends with a set process of interpretation and fusion of classification results. This approach will allow to the classifiers agents to collaborate in making final decisions. It was implemented on the platform JADE, where every step will be handled by specialized agents working together and communicating between them.

Keywords: Dynamic classification, multimodal and evolutionary data, evolutionary classes, Cooperation, Multi-Agent System.

REMERCIEMENTS

Je remercie tout d'abord الله notre créateur **le tout miséricordieux** pour m'avoir donnée le courage et la santé pour accomplir ce travail.

Ce travail n'aurait pas pu aboutir à des résultats sans l'aide et les encouragements de plusieurs personnes que je remercie.

Mes vifs remerciements accompagnés de toute ma gratitude vont ensuite à mon directeur de thèse monsieur MOUSSAOUI ABDELOUAHAB, maître de conférence à l'université de Sétif, pour ses conseils judicieux, sa grande disponibilité et pour m'avoir suivie et orientée.

Je remercie la présidente de jury madame BENBLIDIA NADJIA, maître de conférences à l'université de Blida qui nous a fait l'honneur de présider le jury.

Je remercie aussi Prof. BACHARI NOUR EL ISLAM, professeur à l'université de Bab Ezzouar pour l'intérêt qu'il a bien voulu porter à ce travail en acceptant d'être examinateur.

Je remercie également madame OUKID KHOUAS SALIHA, maître de conférences à l'université de Blida pour m'avoir fait l'honneur d'évaluer mon travail.

Enfin, que tous ceux qui nous ont aidés et encouragés de près ou de loin dans la concrétisation de ce projet, trouvent ici ma gratitude et mes sincères remerciements.

DEDICACES

Je dédie ce modeste travail à,

Mes chers parents,

Mes chers beaux-parents,

Ma chère épouse,

Mes frères et sœurs,

Mes beaux-frères et belles sœurs,

Toute la famille ABBAS, la famille KHABET et la famille BOUKRISSE,

Ma deuxième famille GLOBAL-AXIS

Tous mes amis,

Tous les musulmans,

Et toute l'humanité.

TABLE DES MATIERES

RESUME.....	
REMERCIEMENTS.....	
TABLE DES MATIERES.....	
LISTE DES FIGURES.....	
LISTE DES TABLEAUX.....	
INTRODUCTION GENERALE.....	15

CHAPITRE 1 :

LA CLASSIFICATION DYNAMIQUE DES DONNEES EVOLUTIVES

1. Introduction.....	18
2. Objectifs de la classification dynamique	20
3. Les données évolutives.....	21
4. Problématique de la classification dynamique.....	22
4.1.Cas d'un apprentissage non-supervisée.....	22
4.2.Cas d'un apprentissage séquentiel.....	23
4.3.Cas d'un apprentissage adaptatif.....	23
5. Phénomènes de la classification dynamique des données évolutives.....	24
5.1. Apparition des nouvelles classes	24
5.1.1. Nouvelles classes dans un espace de classification « vide »	25
5.1.2. Changement brusque des données.....	25
5.1.3. Evolution rapide des données.....	25
5.2. Evolution de la partition dynamique.....	26
5.2.1. Modifications locales des classes.....	27
5.2.2. Evolutions avec glissements des classes.....	27
5.3. Fusion des classes.....	28
5.4. Scission des classes.....	29
5.5. Elimination des classes.....	30
6. Conclusion.....	31

CHAPITRE 2 :

METHODES DE CLASSIFICATION DYNAMIQUE

1. Introduction.....	32
2. Modélisation de la problématique.....	33

3. Méthodes supervisées	34
3.1. Classification Bayésiennes	35
3.1.1. Apprentissage Bayésien.....	36
3.1.2. Classificateur Bayésien Naïf [NaïveBayes]	36
3.1.3. Champs de Markov.....	38
3.1.3.1. Lien entre Champs de Markov & Champs de Gibbs.....	39
3.1.4. Algorithme des k plus proches voisins.....	40
3.2. Arbres de décision.....	42
3.2.1. Principe des arbres de décision.....	43
3.2.2. Construction de l'arbre de décision.....	43
3.2.3. Apprentissage des arbres de décision.....	45
3.2.4. Principaux algorithmes des arbres de décision.....	46
3.2.4.1. L'algorithme ID3 (Inductive Decision Tree)	46
3.2.4.2. L'algorithme C4.5.....	46
3.2.4.3. L'algorithme de CART (Classification And Regression Tree)	46
3.3. Réseaux de neurones.....	47
3.3.1. Réseaux de neurones artificiels.....	48
3.3.1.1. Le neurone formel.....	48
3.3.1.2. Architecture des réseaux de neurones artificiels.....	48
3.3.1.3. Apprentissage des réseaux de neurones artificiels.....	50
3.3.2. Réseaux des neurones aléatoires (Random Neural Networks, RNN)	50
4. Méthodes non supervisées	51
4.1. Algorithmes des C -moyennes ("Hard C -Means" HCM)	51
4.2. Classification non supervisée floue.....	53
4.2.1. Théorie des ensembles flous.....	53
4.2.2. Algorithme C -moyennes floues ("Fuzzy C -Means" FCM)	53
4.2.3. Algorithmes de C -moyennes possibilistes ("Possibilistic C -means" PCM) .	55
4.2.4. Algorithme des K -plus proches voisins flou.....	56
5. Conclusion	57

CHAPITRE 3 : FUSION DE DONNEES

1. Introduction	58
2. Théories possibles de fusion des données	59
2.1. La théorie des probabilités (théorie de l'incertain)	59
2.2. La théorie des croyances.....	60
2.3. La théorie des possibilités.....	62
2.3.1. Le modèle flou.....	62
2.3.2. Fonctions des mesures de la théorie possibiliste.....	64
3. La Fusion de données	65
3.1. Problématique de la fusion des données.....	66
3.2. Le processus de la fusion des données.....	67
3.3. Classification des opérateurs de fusion.....	68
3.3.1. Opérateurs à comportement constant et indépendant du contexte.....	68
3.3.2. Opérateurs à comportement variable et indépendant du contexte.....	68
3.3.3. Opérateurs dépendants du contexte.....	68

3.3.4. Quelques propriétés.....	68
3.4. Fusion probabiliste.....	69
3.4.1. Combinaison de la fusion probabiliste Bayésienne.....	69
3.4.2. Règle de décision.....	69
3.5. Fusion des fonctions des croyances.....	70
3.5.1. Combinaison de la fusion d'évidence.....	70
3.5.2. Règle de décision.....	71
3.6. Fusion possibiliste.....	71
3.6.1. Combinaison de la fusion possibiliste.....	71
3.6.2. Règle de décision.....	72
4. Conclusion.....	72

CHAPITRE 4 : LES SYSTEMES MULTI-AGENTS

1. Introduction.....	73
2. Concept d'agent.....	74
2.1. Types d'agents.....	76
2.1.1. Agents réactifs.....	76
2.1.2. Agents cognitifs.....	77
2.1.3. Agents hybrides.....	79
2.1.4. Architecture BDI.....	80
3. Système Multi-Agents.....	81
3.1. Intérêts.....	82
3.2. L'organisation.....	82
3.3. L'environnement.....	83
3.4. Formes d'interaction.....	83
3.4.1. Coopération.....	83
3.4.2. Modes de contrôle.....	85
3.4.3. Coordination.....	85
3.4.4. Communication.....	86
4. Le développement des systèmes multi-agents.....	86
4.1. Méthodologies de conception des systèmes multi-agents.....	86
4.1.1. La méthodologie AAI (Australian Artificial Intelligence Institute).....	87
4.1.2. La méthodologie GAIA	87
4.1.3. La méthodologie Cassiopeia	88
4.1.4. Les méthodologies AUML & MAS-ML.....	88
4.1.5. La méthodologie MAS-CommonKADS	88
4.2. Les plates-formes.....	89
4.2.1. AgentBuilder.....	89
4.2.2. Madkit.....	90
4.2.3. Jade.....	90
4.2.4. Zeus.....	91
5. Conclusion.....	92

CHAPITRE 5 : SYSTEME MULTI-AGENTS POUR LA CLASSIFICATION DYNAMIQUE FLOUE DE DONNEES MULTIMODALES ET EVOLUTIVES

1. Introduction	93
2. Hypothèses	94
3. Problématique	94
4. Les grandes lignes de l’approche proposée	95
4.1. Objectif du système.....	95
4.2. Modélisation des données.....	95
4.3. Echantillonnage.....	95
4.4. Classification.....	96
4.5. Interprétation et décision.....	96
5. Identification des agents	96
5.1. L’agent superviseur (<i>Agent_S</i>)	96
5.2. L’agent de collecte et de modélisation (<i>Agent_M</i>)	97
5.3. L’agent d’échantillonnage (<i>Agent_E</i>)	97
5.4. L’agent de classification : Supervisée (<i>Agent_CS</i>) / Non-supervisée (<i>Agent_CNS</i>).....	97
5.5. L’agent d’interprétation (<i>Agent_I</i>)	98
5.6. L’agent de fusion des interprétations (<i>Agent_F</i>)	98
6. Architecture générale de la méthode proposée	98
7. Schéma de fonctionnement de la méthode proposée	99
7.1. Etape de collecte et modélisation des données.....	101
7.2. Etape d’échantillonnage.....	101
7.3. Etape de classification.....	103
7.4. Etape d’interprétation.....	103
7.5. Etape de fusion et de décision.....	104
8. Conclusion	106

CHAPITRE 6 : IMPLEMENTATION & RESULTATS

1. Introduction	107
2. Présentation de la plate forme utilisée	107
2.1. Architecture de la plate-forme JADE.....	108
2.2. Outils de débogage de JADE.....	109
2.3. Evaluation de JADE.....	110
3. Choix de la base de données	110
3.1. Présentation et description des données	111
3.1.1. Description des données	111
3.1.2. Création de la base de données.....	113
3.2. Caractéristique descriptives des données.....	115
3.2.1. Statistique descriptif des données étudiées	115
3.2.2. Répartition de l’insolation (heure/j) par région et par saison.....	117

3.2.3. Coefficient de variation de l'insolation mensuelle par région	117
3.3. Critères du choix de la base de données.....	118
4. Mise en œuvre.....	119
4.1. Architecture fonctionnelle.....	119
4.2. Métriques retenues pour l'évaluation.....	121
4.2.1. Temps.....	122
4.2.2. Charge de communication.....	122
4.2.3. Précision	123
5. Expérimentation.....	123
5.1. Application et Résultats	125
5.2. Évaluation	127
5.2.1. Analyse suivant la métrique Temps.....	127
5.2.2. Analyse suivant la métrique Charge de communication.....	130
5.2.3. Analyse suivant la métrique Précision.....	133
6. Conclusion.....	134
CONCLUSION GENERALE ET PERSPECTIVES.....	135
ANNEXE.....	137
REFERENCES	144

LISTE DES FIGURES

Figure 1.1 :	Modélisation de l'état du processus et l'évolution de ses modes de fonctionnement	19
Figure 1.2 :	Evolution de la végétation	22
Figure 1.3 :	Apparition de nouvelles classes « (a) espace de classification vide (b) après un changement brusque (c) après une évolution rapide de données»	
Figure 1.4 :	Evolution de la partition dynamique par Modifications locales de classes	27
Figure 1.5 :	Evolution de la partition dynamique par glissements de classes	28
Figure 1.6 :	Fusion de classes	29
Figure 1.7 :	Scission de classes	30
Figure 1.8 :	Elimination de classes	31
Figure 2.1 :	Structure d'un réseau Bayésien naïf	37
Figure 2.2 :	Exemple de classification par la méthode k-pp	42
Figure 2.3 :	Exemple d'arbre de décision	45
Figure 2.4 :	Neurone formel	49
Figure 2.5 :	Architectures d'un réseau de neurones	50
Figure 3.1 :	valeur de vérité pour les théories des ensembles flous et non-flous	62
Figure 3.2:	Exemple d'intersection et d'union dans les sous-ensembles flous	64
Figure 4.1 :	Représentation imagée d'un agent en interaction avec son environnement et les autres agents	76
Figure 4.2 :	Interaction d'un agent réactif avec son environnement et les autres agents	77
Figure 4.3 :	Interaction d'un agent cognitif avec son environnement et les autres agents	78
Figure 4.4 :	Architecture BDI	81
Figure 5.1 :	Architecture générale de la méthode proposée	99
Figure 5.2 :	Schéma de fonctionnement global de la méthode proposée	100
Figure 5.3 :	Schéma de fonctionnement de l'étape de collecte et modélisation des données	102
Figure 5.4 :	Schéma de fonctionnement de l'étape d'échantillonnage	102
Figure 5.5 :	Schéma de fonctionnement de l'étape de classification	104
Figure 5.6 :	Schéma de fonctionnement de l'étape d'interprétation	105
Figure 5.7 :	Schéma de fonctionnement de l'étape fusion et de décision	105
Figure 6.1 :	Architecture de la plate forme JADE. [http://jade.tilab.com]	109
Figure 6.2 :	Les stations du réseau météorologique et radiométrique algérien [Mefti et al,	111
Figure 6.3 :	Nombre de données total et manquantes entre 1950 et 1979.	114
Figure 6.4 :	Nombre de données totales et manquantes entre 1992 et 2002.	114
Figure 6.5 :	Architecture fonctionnelle du système MASDyC.	120
Figure 6.6 :	Barre de paramétrage utilisateur.	120
Figure 6.7 :	Boîtes de dialogue pour les agents.	120
Figure 6.8 :	Evolution de l'insolation journalière brute au cours de l'année 2002 dans la station de Constantine.	121
Figure 6.9 :	Evolution de la fraction d'insolation journalière au cours de l'année 2002	124

	dans la station de Constantine.	
Figure 6.10 :	Evolution des durées d'exécution des agents suite à une série de Tests.	124
Figure 6.11 :	Evolution des durées d'exécution des agents de la phase pré-classement suite à une série de Tests.	128
Figure 6.12 :	Evolution des durées d'exécution des agents de la phase classification suite à une série de Tests.	128
Figure 6.13 :	Evolution des durées d'exécution des agents de la phase post-classification suite à une série de Tests.	129
Figure 6.14 :	Interface de l'Agent_Sniffer.	130
Figure 6.15 :	Visualisation du contenu d'un message ACL.	131
Figure 6.16 :	Charge de communication entre les agents pour un cycle d'exécution complet.	132

LISTE DES TABLEAUX

Tableau 4.1:	Comparaison entre agents cognitifs et agents réactifs	79
Tableau 4.2:	Situations de coopération des agents	84
Tableau 4.3:	Modèles de coopération des agents	84
Tableau 6.1:	Résultats sur la qualité de la plateforme JADE1.2	110
Tableau 6.2:	Coordonnées géographiques des stations météorologiques	113
Tableau 6.3:	Statistique descriptives des stations.	116
Tableau 6.4:	Répartition de l'insolation (par région et par saison) sur le territoire national	117
Tableau 6.5:	Coefficient de variation de l'insolation mensuelle par région.	117
Tableau 6.6:	Caractéristique de la base de données utilisée.	118
Tableau. 6.7:	Principales classes d'agents utilisées dans notre implémentation	119
Tableau. 6.8:	Résultats de la classification globale des stations	126
Tableau. 6.9:	Classes des zones climatiques résultantes après la classification	127
Tableau. 6.10:	Moyenne globale des durées d'exécution par agent	129
Tableau. 6.11:	Résultats de la classification référence	133
Tableau. 6.12:	Taux des précisions estimées sur l'ensemble des classes	133

INTRODUCTION GENERALE

L'extraction de connaissances à partir de données provenant de sources hétérogènes nécessite des techniques plus intelligentes classées dans une nouvelle branche de la recherche scientifique nommée **Apprentissage Automatique**. Le principe de l'apprentissage automatique est de concevoir des systèmes autonomes capables d'extraire des connaissances utiles à partir de données disponibles servant à la prise de décision. Cependant, la fiabilité des sources de données, la nature des connaissances elles-mêmes ainsi que le choix des techniques à utiliser pour extraire ces connaissances sont les principaux problèmes liés à l'hétérogénéité des sources de données.

Très peu de systèmes d'apprentissage sont dotés de cette faculté d'adaptation à l'évolution et la multi-modalité de données. **La Classification Dynamique** est un nouvel axe de l'apprentissage automatique qui s'intéresse justement à ce type de problèmes. La majorité des algorithmes de classification classique ne prennent pas en considération le caractère évolutif des classes et la multi-modalité des données, ce qui conduit à de mauvais résultats de classifications comme dans beaucoup de domaine (*biologie, diagnostic médicale, biotechnologie, reconnaissance de visages, etc.*) où les connaissances évoluent dans le temps et où la véracité des connaissances varie en fonction des capteurs qui peuvent être à la fois complémentaires et parfois contradictoires (*exemple : classification de la végétation qui varie en fonction de la saison*).

La collaboration des classificateurs est une des meilleures solutions pour la classification dynamique des données, où le processus d'apprentissage ainsi que la classification sont partagés entre plusieurs classificateurs qui coopèrent d'une façon intelligente et complètement autonome. Les systèmes multi-agents représentent est une solution très intéressante dans le déploiement d'une telle solution, où chaque classificateur peut être représenté par un agent autonome intelligent coopérant avec d'autres agents dans un même environnement.

L'objectif du travail qui nous a été confié est de proposer une architecture multi-agents pour la classification dynamique floue de données, tout en exploitant les différentes techniques et algorithmes de classification dynamique utiles pour l'extraction de connaissances à partir de données évolutifs (*non-stationnaires*) et multimodales.

Pareillement, afin de résoudre le problème de la multi-modalité des données à classifiées, une étude des techniques de fusion de données a été envisagée, selon les principaux théories de base (*probabiliste, croyance et possibiliste*).

L'approche proposée est bâtie essentiellement sur une solution multi-agents avec une architecture hiérarchique, regroupant plusieurs familles d'agents. Dans la même famille, nous avons fait appel à un ensemble d'agents ayant le même rôle et objectif à atteindre avec une façon de traitement spécifique propre à chaque agent. Au lieu de réaliser une grande phase de traitement de données par un seul agent en appliquant une seule méthode de classification dynamique, le processus final de prise de décision est confié à plusieurs agents coopérant d'une manière complètement autonome et parallèle. Les résultats finaux seront fédérés par un agent superviseur via un processus de fusion de décision

Pour cet objectif, notre travail est organisé de la manière suivante :

Primo, intégré les principaux méthodes de classification dynamique (*floue et non-floue*) de données, afin d'exploiter l'ensemble des performances et privilèges de ses méthodes.

Secundo, proposé une architecture globale avec l'ensemble des agents collaborateurs aux niveaux des différents phases de traitement de données. Cette architecture doit assurer le suivi de l'évolution des classes et de données, et de prendre en charge, par la même occasion, le phénomène de la multi-modalité des données sources.

Tertio, développé un prototype de simulation pour illustrer la manière de la coordination et de la communication des agents classificateurs dans un seul système de classification dynamique floue.

Le manuscrit est organisé en six chapitres:

Le premier chapitre : *Classification dynamique de données évolutives*- Consacré à la définition de quelques notions importantes de la classification dynamique de données évolutives, avec une présentation des différentes caractéristiques des techniques de

classification supervisés, non-supervisés, incrémentales et adaptatives. Le phénomène de l'évolution de données fait également office de ce même premier chapitre.

Le deuxième chapitre : *Méthodes de classification dynamique*- Est une introduction aux méthodes de classification dynamique de données Floues et Non-Floues, avec une représentation des différentes approches et algorithmes supervisés et non-supervisés.

Le troisième chapitre : *Fusion de Données*- Présente le concept de fusion de données, sa définition, ses théories de bases, les différentes méthodologies de la fusion pour la combinaison, ainsi que les règles de décision propres à chaque théorie.

Le quatrième chapitre : *Systèmes Multi-Agents*- Est une introduction aux systèmes multi-agents où les notions de base de la description des différentes méthodes et techniques conceptuelles aux différentes plates-formes existantes, aux différents outils de développement les plus intéressantes sont décrit dans ce chapitre.

Le cinquième chapitre : *Approche SMA pour la classification dynamique floue de données multimodales et évolutives*- Présente notre contribution, nous commençons par présenter brièvement la problématique et les principaux objectifs de notre travail, avant de définir les hypothèses de base de notre approche, avec une spécification du problème et le fonctionnement de notre système proposé (*ensemble des diagrammes, l'architecture générale du système*). En fin, nous détaillons le rôle et le fonctionnement de chaque agent ainsi que leurs différentes interactions dans le processus global de prise de décision.

Le sixième chapitre : *Mise en œuvre et implémentation*- Décrit, l'implémentation et la mise en œuvre de l'approche proposée. Afin d'évaluer les performances de l'architecture conçue, nous allons réaliser une série de simulations sur des données multimodales, floues et évolutives de type **METEO**. Les résultats sont très probants, ce qui prouve l'efficacité de l'architecture proposée.

Le mémoire se termine par une conclusion générale et des perspectives très objectives.

INTRODUCTION GENERALE

L'extraction de connaissances à partir de données provenant de sources hétérogènes nécessite des techniques plus intelligentes classées dans une nouvelle branche de la recherche scientifique nommée **Apprentissage Automatique**. Le principe de l'apprentissage automatique est de concevoir des systèmes autonomes capables d'extraire des connaissances utiles à partir de données disponibles servant à la prise de décision. Cependant, la fiabilité des sources de données, la nature des connaissances elles-mêmes ainsi que le choix des techniques à utiliser pour extraire ces connaissances sont les principaux problèmes liés à l'hétérogénéité des sources de données.

Très peu de systèmes d'apprentissage sont dotés de cette faculté d'adaptation à l'évolution et la multi-modalité de données. **La Classification Dynamique** est un nouvel axe de l'apprentissage automatique qui s'intéresse justement à ce type de problèmes. La majorité des algorithmes de classification classique ne prennent pas en considération le caractère évolutif des classes et la multi-modalité des données, ce qui conduit à de mauvais résultats de classifications comme dans beaucoup de domaine (*biologie, diagnostic médicale, biotechnologie, reconnaissance de visages, etc.*) où les connaissances évolues dans le temps et où la véracité des connaissances varie en fonction des capteurs qui peuvent être à la fois complémentaires et parfois contradictoires (*exemple : classification de la végétation qui varie en fonction de la saison*).

La collaboration des classificateurs est une des meilleures solutions pour la classification dynamique des données, où le processus d'apprentissage ainsi que la classification sont partagés entre plusieurs classificateurs qui coopèrent d'une façon intelligente et complètement autonome. Les systèmes multi-agents représentent est une solution très intéressante dans le déploiement d'une telle solution, où chaque classificateur peut être représenté par un agent autonome intelligent coopérant avec d'autres agents dans un même environnement.

L'objectif du travail qui nous a été confié est de proposer une architecture multi-agents pour la classification dynamique floue de données, tout en exploitant les différentes techniques et algorithmes de classification dynamique utiles pour l'extraction de connaissances à partir de données évolutifs (*non-stationnaires*) et multimodales.

Pareillement, afin de résoudre le problème de la multi-modalité des données à classifiées, une étude des techniques de fusion de données a été envisagée, selon les principales théories de base (*probabiliste, croyance et possibiliste*).

L'approche proposée est bâtie essentiellement sur une solution multi-agents avec une architecture hiérarchique, regroupant plusieurs familles d'agents. Dans la même famille, nous avons fait appel à un ensemble d'agents ayant le même rôle et objectif à atteindre avec une façon de traitement spécifique propre à chaque agent. Au lieu de réaliser une grande phase de traitement de données par un seul agent en appliquant une seule méthode de classification dynamique, le processus final de prise de décision est confié à plusieurs agents coopérant d'une manière complètement autonome et parallèle. Les résultats finaux seront fédérés par un agent superviseur via un processus de fusion de décision

Pour cet objectif, notre travail est organisé de la manière suivante :

Primo, intégré les principales méthodes de classification dynamique (*floue et non-floue*) de données, afin d'exploiter l'ensemble des performances et privilèges de ses méthodes.

Secundo, proposé une architecture globale avec l'ensemble des agents collaborateurs aux niveaux des différentes phases de traitement de données. Cette architecture doit assurer le suivi de l'évolution des classes et de données, et de prendre en charge, par la même occasion, le phénomène de la multi-modalité des données sources.

Tertio, développé un prototype de simulation pour illustrer la manière de la coordination et de la communication des agents classificateurs dans un seul système de classification dynamique floue.

Le manuscrit est organisé en six chapitres:

Le premier chapitre : *Classification dynamique de données évolutives*- Consacré à la définition de quelques notions importantes de la classification dynamique de données évolutives, avec une présentation des différentes caractéristiques des techniques de

classification supervisés, non-supervisés, incrémentales et adaptatives. Le phénomène de l'évolution de données fait également office de ce même premier chapitre.

Le deuxième chapitre : *Méthodes de classification dynamique*- Est une introduction aux méthodes de classification dynamique de données Floues et Non-Floues, avec une représentation des différentes approches et algorithmes supervisés et non-supervisés.

Le troisième chapitre : *Fusion de Données*- Présente le concept de fusion de données, sa définition, ses théories de bases, les différentes méthodologies de la fusion pour la combinaison, ainsi que les règles de décision propres à chaque théorie.

Le quatrième chapitre : *Systèmes Multi-Agents*- Est une introduction aux systèmes multi-agents où les notions de base de la description des différentes méthodes et techniques conceptuelles aux différentes plates-formes existantes, aux différents outils de développement les plus intéressantes sont décrit dans ce chapitre.

Le cinquième chapitre : *Approche SMA pour la classification dynamique floue de données multimodales et évolutives*- Présente notre contribution, nous commençons par présenter brièvement la problématique et les principaux objectifs de notre travail, avant de définir les hypothèses de base de notre approche, avec une spécification du problème et le fonctionnement de notre système proposé (*ensemble des diagrammes, l'architecture générale du système*). En fin, nous détaillons le rôle et le fonctionnement de chaque agent ainsi que leurs différentes interactions dans le processus global de prise de décision.

Le sixième chapitre : *Mise en œuvre et implémentation*- Décrit, l'implémentation et la mise en œuvre de l'approche proposée. Afin d'évaluer les performances de l'architecture conçue, nous allons réaliser une série de simulations sur des données multimodales, floues et évolutives de type **METEO**. Les résultats sont très probants, ce qui prouve l'efficacité de l'architecture proposée.

Le mémoire se termine par une conclusion générale et des perspectives très objectives.

CHAPITRE 1

LA CLASSIFICATION DYNAMIQUE DE DONNEES EVOLUTIVES

1. Introduction

L'apparition des entrepôts de données (*DATAWERHOUSE*) dans le monde informatique a changé le comportement et la culture des utilisateurs, à titre de décideurs, envers les systèmes d'information. Au départ, il s'agissait d'éditer tous les flux d'informations et de données circulants dans un processus de gestion (commercial, médical, industriel...), avec des états de sorties et des rapports statiques difficiles à utiliser dans un processus décisionnel. Ensuite, l'ère des entrepôts de données est apparue, où leur chargement se faisait qu'avec des données importantes bien nettoyées, agrégées et extraites à partir de différentes bases de données opérationnelles hétérogènes. L'architecture des entrepôts de données a permis d'éditer facilement des rapports dynamiques très efficaces et collaborant dans le processus d'aide à la décision.

Les chercheurs sont partis encore plus loin, où ils ont essayé d'exploiter ces entrepôts de données non pas par la récupération directe de données stockées, mais plutôt par l'extraction de connaissances à partir de ces données entreposées, avec des techniques plus intelligentes classées dans une nouvelle branche de la recherche scientifique nommée **Apprentissage Automatique**. Le principe de l'apprentissage automatique est de concevoir des systèmes autonomes capables d'extraire les connaissances utiles à partir des données disponibles servant à la prise de décision.

Un système d'apprentissage automatique a des capacités plus importantes que de la simple tâche de récolte et de stockage des connaissances, il dispose de la possibilité d'apprendre automatiquement de nouvelles connaissances non connues *a priori*, ainsi son adaptation lorsque ces connaissances déjà apprises évoluent dans le temps. Or, très peu de systèmes d'apprentissage peuvent faire face à cette évolution des connaissances.

Le présent chapitre présente les notions importantes de la **Classification Dynamique**, où nous avons suivi la démarche du travail initié par [1], vu son étude détaillée et très riche sur ce domaine dans sa thèse de doctorat. Ce dernier qui qualifie la **Classification Dynamique** comme étant un nouvel axe de l'apprentissage automatique qui va s'intéresser aux problèmes de modélisation adaptative et classification de données évolutives (*non-stationnaires*).

Les algorithmes de la classification dynamique représentent les connaissances, utiles à extraire à partir des données, par des classes capables d'évoluer dans le temps, ces classes contiennent l'ensemble de connaissances extraites à partir des vecteurs de données qui représentent l'état du processus suivi.

La figure suivante représente la modélisation d'un processus ainsi l'évolution de ses modes de fonctionnement :

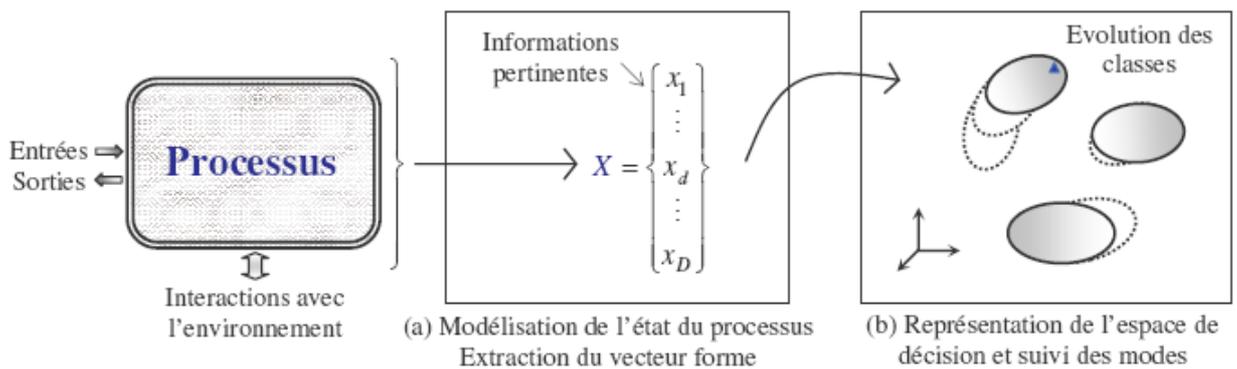


Figure 1.1 : Modélisation de l'état du processus et l'évolution de ses modes de fonctionnement [1].

Le suivi de l'évolution des modes de fonctionnement d'un processus dans un environnement interactif avec des entrées et des sorties est réalisé sur deux étapes. La première, est une étape de modélisation de l'état du processus par son vecteur forme à chaque instant t , pour cela deux méthodes sont définies :

- Extraction d'un modèle physique à partir des entrées et des sorties du processus.
- Extraction des informations pertinentes à partir des signaux des capteurs.

La deuxième étape consiste à représenter l'espace de décision avec une adaptation des classes existantes (*évolution des classes*), ou création de nouvelles classes à partir des nouvelles connaissances recueillies à partir des données du vecteur forme. Dans cette espace de décision l'état du processus sera modélisé à chaque instant t .

Pour plus de précision, citons l'exemple de la supervision du trafic réseau où nous devons détecter les différentes intrusions et pannes du réseau informatique. Dans ce cas, les classes sont les modes de fonctionnement du réseau. Les intrusions ou les défaillances (*pannes ou lenteur de la bande passante*) se manifestent dans l'espace de décision par des changements brusques du trafic réseau ou par des évolutions d'un ou plusieurs modes de fonctionnement. Se disposer d'un outil capable d'apprendre automatiquement les connaissances utiles, extraites du fonctionnement du réseau et son évolution, est très intéressant et collaborant dans la détection des intrusions et les anomalies du trafic réseau.

2. Objectifs de la classification dynamique

Les algorithmes utilisés dans une classification dynamique sont développés sur la base d'un des **objectifs** suivants, selon la disponibilité de données :

a- Apprentissage supervisé: le but de l'algorithme de classification est de construire des classes en connaissant *à priori* les étiquettes d'appartenance des données aux classes, et cela à partir d'une base d'apprentissage déjà construite par des experts.

b- Apprentissage non-supervisé (*Unsupervised Learning*): l'objectif est de faire un apprentissage complètement autonome sans l'aide d'un superviseur, où les étiquettes de données ne sont pas connues *a priori*. Pour cela l'algorithme de classification a la double tâche, il doit être capable de définir l'appartenance des données aux classes et construire la partition des classes en utilisant généralement un critère de ressemblance.

c- Apprentissage séquentiel (*en ligne ; on-line Learning*): ou plutôt Apprentissage incrémental (*traitement séquentiel d'information*), ici la base d'apprentissage n'est pas disponible *a priori* ou incomplète [2]. L'objectif de l'algorithme est de pouvoir apprendre séquentiellement des connaissances supplémentaires au fur et à mesure que de nouvelles données sont acquises, sans obligation de retourner aux données de la base d'apprentissage initiale. Ainsi de pouvoir contribuer les anciennes et les nouvelles connaissances pour la définition du modèle de classification.

d- Apprentissage adaptatif (*adaptive Learning*) : cet objectif sera sollicité dans le cas où nous avons un processus très dynamique qui peut évoluer entre plusieurs modes de fonctionnements. L'algorithme de classification doit être capable de détecter les états inconnus, ainsi de les représentés par de nouvelles classes dans le modèle de classification. L'utilisation des règles d'adaptation pour la prise en compte des nouveaux changements, affectation des nouvelles données dans les classes, est très indispensable [8]. Cette technique est très gourmande en temps de calcul et ne tire aucun avantage des connaissances extraites au cours des apprentissages précédents [1].

3. Les Données évolutives

On vient de présenter les principaux objectifs des algorithmes de la classification dynamique : apprentissage supervisé, non-supervisé, incrémental et adaptatif, pour chaque objectif existe d'innombrables méthodes de classification. La plupart des travaux de recherches portent de plus en plus beaucoup d'intérêt aux méthodes et techniques de la classification incrémentale [1]. Ce type de classification est fréquent lorsqu'il s'agit d'une manipulation de données non-stationnaires, où le modèle de classification évolue dans le temps par l'adaptation des modifications locales au niveau des classes (*création, fusion, scission et élimination de classes*). La non-stationnarité de données ou plutôt les données évolutives sont devenues un souci majeur dans la plupart des études de recherches qui portent sur la classification dans différents domaines.

Nous pouvons spécifier les « données évolutives (*non-stationnaire*) » par la définition suivante [1]. **Définition 1.1** : « *les données non-stationnaires sont des données issues d'un processus dont le comportement est susceptible d'évoluer dans le temps. La non-stationnarité entraîne une évolution des caractéristiques des modèles de connaissances (classes) apprises de ces données. En classification, nous parlons de classes évolutives.* »

Selon la définition ci-dessus, une introduction de la classification dynamique n'est possible que dans le cas où nous avons une classification de données non-stationnaires résultantes d'un processus évolutif dans le temps, ces données évolutives engendrent des évolutions des classes qui modélisent l'espace de classification.

Dans les travaux de Luc Descroix [18], concernant l'analyse de l'évolution de la végétation par la classification barycentrique au nord du Mexique, les scènes suivantes ont été photographiées par le satellite LandSat MSS (*Décembre 1972 & 1992*), où nous pouvons constater l'évolution de données.

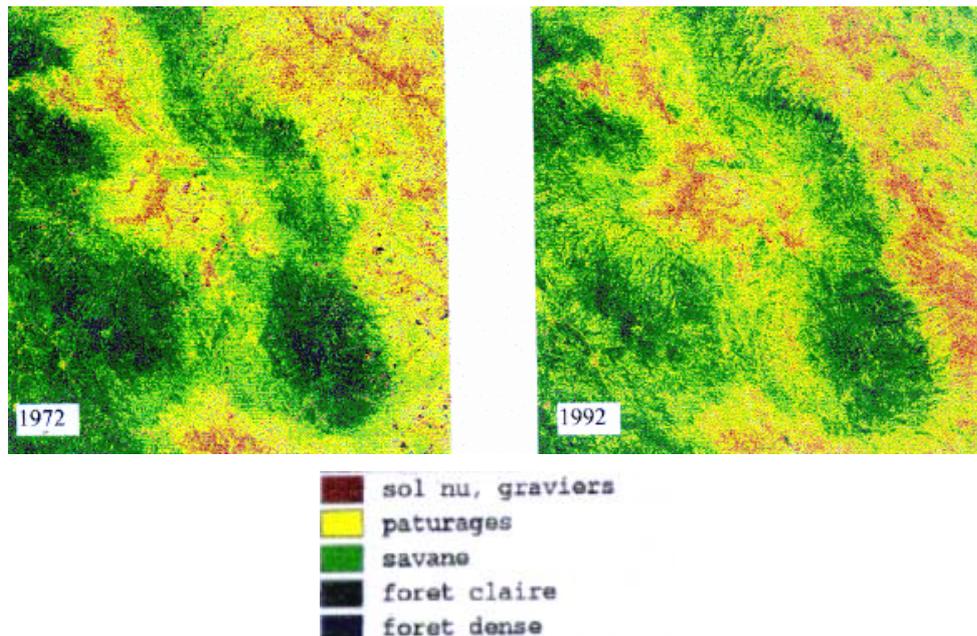


Figure 1.2 : Evolution de la végétation [18].

4. Problématique de la classification dynamique

Généralement, les principaux problèmes liés à la classification dynamique se scindent en deux. En premier lieu, nous avons la problématique relative aux méthodes d'apprentissage suivies : supervisé, non-supervisée, incrémental ou adaptatif. En second lieu, nous avons la problématique de réaliser une classification dans un environnement de données non-stationnaires qui engendrent l'évolution des classes (*disparition, apparition, fusion et scission*).

Par la suite, nous allons énumérer les différentes problématiques de la classification dynamique selon la méthode d'apprentissage suivie. Notons qu'un modèle de classification est un ensemble de classes ou regroupements de données homogènes [1] :

4.1. Cas d'un apprentissage non-supervisée

- Aucune information sur les classes d'appartenance de données n'est disponible *a priori*. Ainsi, le nombre des classes à former n'est pas connu *a priori*, par conséquent le nombre de composantes du modèle à ajuster est inconnu [13].

- La définition d'un critère de ressemblance est nécessaire pour l'affectation des données aux classes. La détermination de ce critère est basée sur la nature du problème à traiter « *dans le cadre de la classification des données spatiales, le critère de ressemblance vise à affecter les données géographiquement proches (ou similaires) à la même classe. Par exemple, l'utilisation d'une distance euclidienne conduit à des classes de modèle hyper-sphériques* » [1].

4.2. Cas d'un apprentissage séquentiel

- La base d'apprentissage n'est pas disponible *a priori* [2], nous devons apprendre des connaissances supplémentaires à partir des nouvelles données, sans la nécessité d'accéder aux données d'origines de la base. Donc, en préservant les anciennes connaissances nous devons avoir la possibilité d'actualiser le modèle de connaissance (*introduire de nouvelles classes*) à partir des nouvelles données acquises.
- L'enjeu des techniques d'apprentissage séquentiel est d'utiliser les connaissances extraites des données déjà acquises et sans attendre que la base d'apprentissage soit complétée par des nouvelles données dans le future.

4.3. Cas d'un apprentissage adaptatif

- Nécessiter de la mise à jour du modèle de classification à chaque acquisition des nouvelles données, soit par des modifications locales des classes (*fusion, scission et déplacement*), ou par des évolutions du modèle de classification (*création et élimination de classes*). Pour cela, nous devons mettre en place des règles d'adaptation afin d'apprendre les changements du modèle de classification après chaque incorporation des données.
- Difficulté dans la détection des états inconnus du mode de fonctionnement du processus. Entre autre, nous devons pouvoir apprendre ces nouveaux états et les représenter par une apparition des nouvelles classes dans le modèle de connaissance [8].
- L'évolution des classes engendre un changement de leur espace de représentation, la disposition des règles d'apprentissage continu est donc nécessaire pour le suivi de ces évolutions.

5. Phénomènes de la classification dynamique de données évolutives

Le changement dynamique des classes peut être représenté par plusieurs mécanismes et phénomènes en environnement non stationnaire, cinq actions ont été dégagées dans [1], notamment avec les deux définitions suivantes :

Définition 1.2 : « *Une classe est définie comme un ensemble (ou regroupement homogène) de données similaires. La similarité est évaluée d'un point de vue de la proximité géométrique des données. Lorsque les données d'une classe évoluent dans le temps, cette classe est dite évolutive. Son modèle doit donc être adapté* ».

Définition 1.3 : « *Une partition dynamique représente l'ensemble des regroupements homogènes des données non-stationnaires. Elle est associée à un modèle de connaissances (i.e. classes) susceptible d'évoluer dans le temps en fonction de l'incorporation de nouvelles données* ».

Par la suite nous allons détailler chaque mécanisme dynamique avec des exemples d'illustration, en expliquant les trois propriétés suivantes:

- L'interaction des nouvelles données avec la partition dynamique.
- Les risques de la méthode de classification dynamique.
- Les mécanismes entraînés par l'insertion des données dans la partition.

5.1. Apparition des nouvelles classes

L'apparition des nouvelles classes est occasionnée par l'acquisition des nouvelles données similaires, soit regroupées dans une même région de l'espace, ou bien situées dans des différentes régions non occupées par les classes existantes.

D'abord, nous devons déterminer les données suffisamment proches pour former chaque classe, et cela en utilisant un critère de similarité, ensuite, calculer le modèle des nouvelles classes créées par la mise en œuvre d'une procédure d'initialisation respectant la distribution des données. Ce mécanisme de création met à jour la partition dynamique par l'incorporation des nouvelles classes.

L'apparition des nouveautés dans un espace de classification est alors définie par l'une des trois formes illustrées dans le schéma ci-dessous (Figure 1.3).

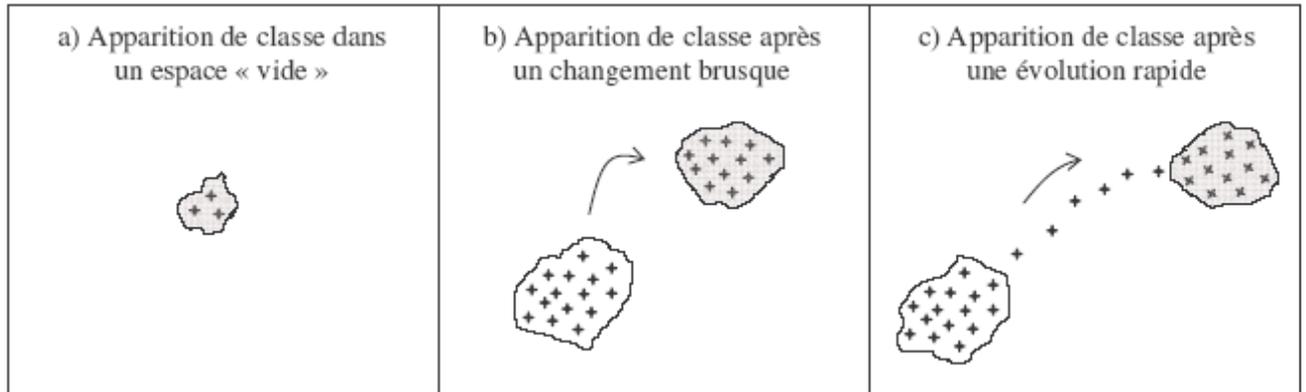


Figure 1.3 : Apparition des nouvelles classes « (a) espace de classification vide (b) après un changement brusque (c) après une évolution rapide de données». [1]

5.1.1. Nouvelles classes dans un espace de classification « vide »

Ne peut être arrivé qu'à l'initialisation de l'apprentissage, où les premières données arrivent dans un espace de classification « vide » (*il ne contient aucune classe*). Dès l'acquisition des premières données nous avons deux cas de figures possibles: le cas où ces données sont similaires, nous allons les intégrer dans une *seul classe*, sinon, pour le cas contraire nous allons créer *plusieurs classes séparées*.

5.1.2. Changement brusque des données

Dans ce cas, l'espace de classification est déjà initié et il contient une ou plusieurs classes. Il s'agit d'une apparition brusque des nouvelles classes suite à l'incorporation des nouvelles données *similaires* et *suffisantes*, regroupées dans un endroit séparé des classes existantes à l'intérieur de la partition dynamique.

5.1.3. Evolution rapide des données

Dans cette situation, l'apparition des nouvelles classes va être précédée par une évolution (*transition*) rapide des données transitoires entre les classes, l'évolution *plus au moins rapide* des données entraîne une dérivation de la classe de départ vers une classe de destination.

Cette dernière, peut être déjà définie (elle existe dans la partition dynamique), mais le cas qui nous intéresse c'est qu'elle soit créée comme une nouvelle classe à la suite d'une stabilisation des données suffisantes dans une région isolée de la partition dynamique.

Avec l'absence d'une supervision permanente sur la partition dynamique, plusieurs risques peuvent se présenter conséquemment à l'apparition des nouvelles classes. Citons comme exemples, l'attribution des données des classes différentes à une seule classe, ou le contraire, l'affectation des données similaires d'une même classe à des classes disjointes.

Les effets du bruit et des perturbations externes génèrent fréquemment des erreurs de classification, et créent des *nouvelles classes parasites* dégradant la qualité de la modélisation. Cependant, ces classes parasites doivent être éliminées sinon elles seront confondues avec des classes fiables représentatives des données présentes d'une façon périodique dans une même région de l'espace.

5.2. Evolution de la partition dynamique

C'est un phénomène largement étudié dans les travaux de recherches qui s'intéressent à la classification dynamique [8][40][17]. Les auteurs de ces travaux confirment que l'incorporation des nouvelles données *non-stationnaires* entraîne l'évolution de la partition dynamique (*évolution des classes représentatives existantes*) soit par des modifications locales des classes, ou par une modification globale de la partition. L'évolution des classes est envisageable pour tout système fonctionnant dans un environnement évolutif, par ailleurs le nombre des classes ainsi leurs contours ne sont jamais stables ils peuvent être changé à chaque acquisition des nouvelles données.

Voici deux questions jugées importantes dans [1], elles doivent être posées avant que l'adaptation ou l'évolution d'une classe soit faite au sein d'une partition dynamique :

- De quelle classe une nouvelle donnée est-elle suffisamment proche pour contribuer à la redéfinition de son modèle ?
 - *L'utilisation d'un critère de similarité est souhaitée pour déterminer l'appartenance des données aux classes.*
- De quelle manière l'information portée par cette nouvelle donnée contribue-t-elle à adapter le modèle de la classe ?
 - *La mise en œuvre de règle d'adaptation récursive peut aider à répondre à cette question.*

Compte tenu de la présentation ci-dessus, l'évolution d'une partition dynamique est possible selon deux manières différentes (*localement ou globalement*). Dans ce qui suit nous allons détailler ces deux méthodes séparément avec des schémas d'illustration :

5.2.1. Modifications locales des classes

L'incorporation des nouvelles données non-stationnaire au sein d'une classe peut générer des modifications locales dans cette dernière (*grossissements, rétrécissements et rotations*). Les modifications sont visibles localement au milieu de l'espace de définition de la classe avec une nouvelle restructuration de son contour. Précisément la classe est représentée par un nouveau schéma dans le même espace de sa définition (*exemple médicale: expansion d'une tumeur*).

« La plupart des méthodes de modélisation adaptative développées avec des règles d'apprentissage incrémental (sans oubli d'information), se limitent à la prise en compte des modifications locales. Au travers de ces règles, l'adaptation de modèles de classes se fait donc localement » [1].

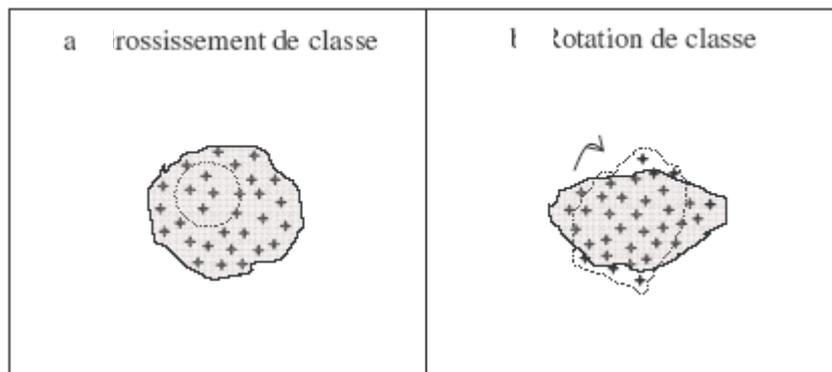


Figure 1.4 : Evolution de la partition dynamique par Modifications locales des classes [1].

5.2.2. Evolutions avec glissements des classes

Lorsque les nouvelles données d'une même classe s'éloignent progressivement de la région occupée par les anciennes données, le phénomène d'évolution avec glissements des classes peut se générer, à travers des simples déplacements des classes, ou une combinaison des déplacements avec des restructurations des contours de ces classes. Dans cette situation, un changement de *la région de définition* des classes est observable suite à ces évolutions.

Ces glissements des classes rendent l'application de classification de données non stationnaires hautement vigilante vis-à-vis des évolutions des classes représentatives présentes dans la partition dynamique. Pour cela, l'utilisation des règles de mise à jour récursives sur ces classes (*déjà existantes*) est très collaborant. Le principe de mise à jour est simplement défini par l'intégration des *informations récentes* apportées par les nouvelles données dans la classe d'une part, et l'élimination des *informations inutiles* contenues dans les données les plus anciennes de cette même classe de l'autre part.

Nous pouvons résumer les principales difficultés confrontées suite à l'utilisation des règles de mise à jour, suivant le modèle de classification choisi :

- Imperfection sur la modélisation des classes causée par le bruit.
- Atténuation de la qualité de modélisation dynamique dans le cas d'un traitement incorrect des interactions entre les classes (*fusion, scission*) pendant leurs évolutions.
- Confus dans la distinction entre le phénomène de dérivé des données et le mécanisme d'évolution de classe.

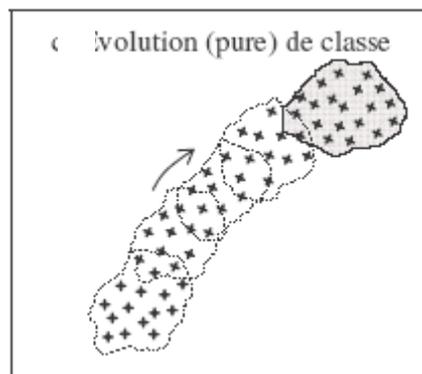


Figure 1.5 : Evolution de la partition dynamique par glissements des classes [1].

5.3. Fusion des classes

Le phénomène de fusion des classes est souvent survenu dans la classification dynamique suite à des interactions entre deux ou plusieurs classes de la même partition dynamique, mais elles sont plutôt définies sur des espaces différents. Lors de l'incorporation des nouvelles données porteuses d'informations mutuelles, deux ou plusieurs classes initialement disjointes peuvent se rejoindre et partager ces nouvelles données. La méthode de classification dynamique modélise cette action par le mécanisme de fusion des classes, en conséquence nous allons avoir une seule classe « résultante ».

Dans ce cas, le *critère de ressemblance* peut assurer la détection de ces informations mutuelles. Or, la mise en œuvre des *règles de fusion* aide à déterminer le modèle résultant de la fusion, sans effectuer un réapprentissage sur toutes les données. Parfois, l'utilisation des règles de fusion peut être influencée par le bruit dans les données, en déclenchant une fusion des classes non désirée après la détection des informations mutuelles frauduleuses.

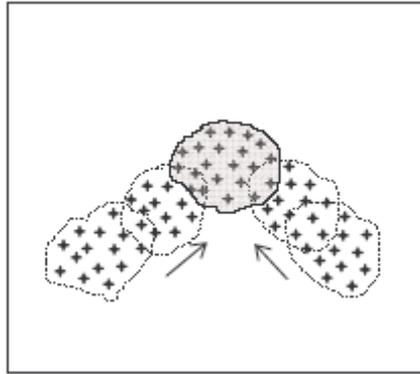


Figure 1.6 : Fusion des classes [1].

5.4. Scission des classes

Contrairement au mécanisme de la fusion, le phénomène de scission des classes se présente lorsqu'une classe se divise en deux classes au cours de l'acquisition de nouvelles données non-stationnaires. De façon plus claire, supposons qu'un ensemble de données parasites (*bruitées*) se mixent avec des données certaines décrivant l'évolution d'un mode de fonctionnement du processus, donc la classe initiale qui doit contenir ces deux types de données est dans l'obligation d'être divisée en deux nouvelles classes : une classe représentative obsolète (*avec les données certaines*) et une classe parasite (*avec les données bruitées*). Elles sont nouvellement définies séparément dans la même partition dynamique.

Dans de telles situations, nous devons faire attention et éviter de construire des classes avec des données non similaires, ce qui dégrade la modélisation de l'espace de connaissance. En effet, la classification dynamique utilise un *critère de détection de scission* de classe afin d'appliquer la scission.

Ensuite, les *règles de scission* de classe seront utilisées pour déterminer les modèles de classes résultantes. Or, la mise en œuvre du critère de détection de scission, représente une complexité principale pour une application adéquate de ce phénomène, car le critère choisi doit répondre à la question suivante posée dans [1]: « *Les données contenues dans chaque*

classe sont elles similaires ou non ? ». L'auteur précise dans sa réponse qu'il est difficile de mettre en œuvre un critère de similarité répondant de façon évidente et fiable à cette question.

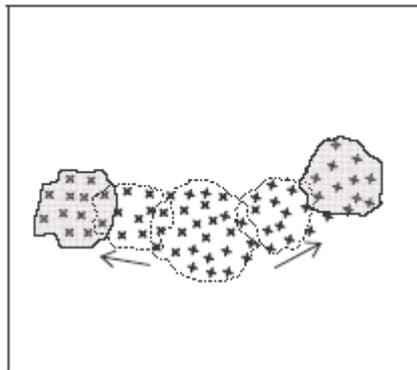


Figure 1.7 : Scission des lasses [1].

5.5. Elimination des classes

Dans un espace de classification dynamique où la base d'apprentissage est incomplète, quelques connaissances apprises dans un passé plus ou moins lointain ne deviennent plus valables. Or, La modélisation des modes de fonctionnement d'un processus est faite d'une façon continue toute au long de sa période d'exécution, les classes représentant les connaissances obsolètes et les classes parasites, créées à cause du bruit ou des perturbations externes, doivent être éliminées au fur et à mesure à chaque nouvelle modélisation de l'espace de classification après acquisition des nouvelles données.

La détection des classes parasites est généralement basée sur un *critère de cardinalité*, ainsi que la détection des classes obsolètes dépend forcément du contexte de l'application.

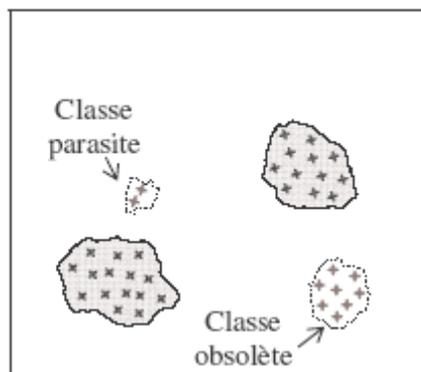


Figure 1.8 : Elimination de classes [1].

6. Conclusion

La classification dynamique est considérée comme étant un nouvel axe de recherche prometteur pour les procédures d'apprentissage automatique et d'extraction des connaissances, dans un environnement susceptible d'évoluer dans le temps. Ce type de classification touche plusieurs domaines : surveillance vidéo « suivi de mouvements de la cible », audit du réseau informatique « détection des intrusions », classification saisonnière de la végétation « analyseur de teneur en chlorophylle », etc.

Dans cette partie, nous avons rapporté les différents phénomènes dynamiques présents dans le contexte de la classification dynamique de données évolutives, ainsi les problématiques et les difficultés causées par la non-stationnarité des données en environnement dynamique.

Dans le chapitre suivant, nous allons présenter les différentes méthodes et algorithmes, les plus abordés dans la littérature, visant à résoudre cette problématique de classification dynamique, ainsi une définition des techniques utilisées pour le traitement de la notion « floue et non-floue » conjointement selon la possibilité de chaque algorithme.

CHAPITRE 2

METHODES DE CLASSIFICATION DYNAMIQUE

1. Introduction

Une méthode de classification classique a pour objectif d'associer chaque élément de l'ensemble des *individus stationnaires* $X = \{X_1, \dots, X_n\}$, à l'une des k *classes statiques* définies dans l'ensemble $C = \{C_1, \dots, C_k\}$, selon un critère de ressemblance. Un individu X_i est associé à la classe C_j si seulement s'il est de même nature des individus déjà regrouper dans la classe C_j . Ce qui explique une stationnarité permanente du modèle de classification au cours du temps. L'introduction de la propriété « évolution (*non-stationnarité*) » dans cette méthode, où X devient un ensemble *d'individus non-stationnaires* et C un ensemble de *classes évolutives*, nous permet d'introduire la méthode de classification dynamique.

Concevoir un algorithme de classification dynamique est un travail assez délicat vu la difficulté de manipuler des données non-stationnaires, afin de les affectées dans des classes évolutives au moyen d'une base d'apprentissage parfois incomplète (*l'étiquetage des données est inconnu a priori*). Dans la littérature, nous trouvons que l'algorithme ISODATA «*Iterative Self-Organizing Data Analysis Technics A*» introduit par Ball, Geoffrey et Hall, David en 1965 [43], est l'un des premières réalisations en classification dynamique (*classification non supervisée*), l'algorithme avait comme objectif de classer un ensemble d'individus (*objets*) caractérisés par des variables quantitatives d'une façon itérative. Entre autre, il a bien introduit les différents phénomènes d'évolution des classes (*scission, fusion et élimination*). Mais, la principale difficulté rencontrée par ISODATA est qu'il faut avoir *a priori* une base de connaissances sur les données afin de fixer leurs paramètres de classification.

L'avancement des méthodes et modèles mathématiques et statistiques a servi comme un appui pour le développement des systèmes informatiques appliqués dans des environnements dynamiques et évolutifs. De nos jours, la classification dynamique est très sollicitée et appliquer dans divers domaines (*Robotique, Contrôle des processus, Apprentissage Artificielle, Traitement d'image..*).

Dans la partie précédente nous avons montré les objectifs et difficultés liées aux méthodes de la classification dynamique. La détermination d'une approche adéquate à cette problématique est notre objectif principal dans ce manuscrit. L'étude préalable de l'ensemble de techniques et algorithmes est présentée dans cette partie, où nous avons défini l'ensemble des méthodes de classification dynamique.

La non stationnarité des données implique l'évolution temporelle des classes qui se caractérise par divers phénomènes dynamiques comme l'apparition des classes, l'évolution des classes, etc. selon Bezdek, on peut identifier ces problèmes en deux variantes: la classification dite « *supervisée* » et la classification dite « *non supervisée* ». Ce chapitre est divisé en deux parties, en premier lieu, nous présentons l'ensemble des **méthodes supervisées**, en second lieu, nous passerons à la présentation **des méthodes non supervisées**, et nous terminons par une étude des méthodes de classification floues.

2. Modélisation de la problématique

Dans la présentation des méthodes, nous utilisons la modélisation de [57]. Un problème de classification est représenté par une chaîne de transformation mettant en relation trois espaces différents :

- **L'espace de mesure** : cet espace est réalisé par une méthode de perception pour représenter le monde réel et physique.
- **L'espace de représentation**: à partir de l'espace de mesure, les informations sont utilisées dans la représentation des objets. Soit Π une population d'individus, d'objets ou d'exemples: alors chaque exemple est décrit par un ensemble d'attributs notés $x_1, \dots, x_i, \dots, x_p$ qui prennent leurs valeurs respectivement dans $R_1, \dots, R_i, \dots, R_p$, donc l'espace de représentation R est le produit cartésien $R_1 \times \dots \times R_i \times \dots \times R_p$. La représentation de l'exemple π dans R est déterminée par la fonction χ , où chaque valeur de l'attribut x_i de l'exemple X est décrite par $x_i(X)$.

$$\chi: \Pi \rightarrow R = R_1 \times \dots \times R_i \times \dots \times R_p$$

$$X \rightarrow \chi(X) = (x_1(X), \dots, x_i(X), \dots, x_p(X))$$

- **L'espace de décision ou d'interprétation**: cet espace regroupe l'ensemble des classes et des étiquettes qui seront associées à chaque élément de l'espace de représentation R . Soit Y une variable de représentation des classes, elle prend ses valeurs dans un espace de décision ou d'interprétation C (*ensemble discret de cardinalité fini c*). Chaque individu X de la population Π est associé à une classe représentée par Y :

$$Y: \Pi \rightarrow C = \{y_1, y_2, \dots, y_c\}$$

$$X \rightarrow Y(X)$$

Le but final de chaque méthode de classification est de déterminer une procédure de prévision P afin de classer chaque exemple X de la population Π , représenté par la fonction χ à l'intérieur de l'espace de représentation R , dans une classe y de l'espace d'interprétation C .

$$P: R \rightarrow C$$

$$\chi(X) \rightarrow y$$

« Dans la pratique, il s'agit de construire P , c'est-à-dire des opérateurs, des méthodes qui, à partir de $X \in \Pi$ permettent de décider automatiquement à quel $y \in C$ l'élément X est affecté » [57].

3. Méthodes supervisées

Chaque méthode de classification dynamique doit passer obligatoirement par deux phases [10]. La première est une phase d'apprentissage sur des échantillons représentatifs, durant laquelle nous définissons les règles de classification et les caractéristiques de chaque classe de l'espace d'interprétation, qui permettront de prédire la classe d'appartenance d'un objet en fonction de sa description. Pour ensuite passer à la seconde phase de décision et d'affectation de chaque objet à sa classe d'appartenance, en fonction des règles de classification découvertes.

Pour parler des méthodes de classification supervisée nous devons utiliser un espace de représentation construis par l'une des techniques d'apprentissage supervisé. Avant d'affecter les objets de l'espace de représentation aux classes, ces dernières doivent être connues *a priori*. L'apprentissage supervisé repose sur une base de données contenant des exemples de cas déjà étiquetés [59].

Cette base de données est un ensemble de couples entrées sorties (X_i, y_i) aléatoires. Alors, le but de l'apprentissage supervisé est d'apprendre à prédire pour toute nouvelle entrée X la sortie y à *valeur discrète*.

Le principe de la classification supervisée est de regrouper dans des classes connues un ensemble des données (*exemples ou échantillons*) *étiquetées* avec le nom de leur classe d'appartenance. La problématique de la supervision est de pouvoir analyser les *ressemblances* et/ou les *dissemblances* de données afin de définir les caractéristiques de chaque classe, et être capable de classer une forme *inconnue (sans étiquette)* parmi les classes connues *a priori* [3].

La catégorisation des méthodes de classification n'est pas une tâche facile [3]. Cependant, nous allons nous intéresser dans cette section à la présentation des méthodes de classification les plus utilisées dans les travaux de recherches.

3.1. Classification Bayésienne

Les méthodes Bayésiennes sont très puissantes dans la résolution des problèmes de classification dynamique grâce à leurs techniques d'apprentissage automatique déployées. Elles sont généralement utilisées en intelligence artificiel afin de modéliser des connaissances complexes et incertaines avec un raisonnement probabiliste basé sur les deux théories suivantes :

- *Théorie des probabilités* : quantification de l'incertitude des relations de causalité entre les variables du domaine.
- *Théorie des graphes* : représentation de la connaissance sous forme d'un graphe causal.

L'objectif d'une classification Bayésienne est de calculer pour chaque nouvel individu X tous les probabilités d'appartenance conditionnelles aux classes y_j ($j \in \{1, \dots, c\}$), pour ensuite choisir *la classe la plus probable*. La décision de la classe est donnée par la fonction φ suivante :

$$\varphi: R \rightarrow C$$

$$X \rightarrow y_j / \forall y \in C, P(y_j/X) \geq P(y/X)$$

Où : $P(y_j/X)$ est la probabilité conditionnelle que l'individu X est rattaché à la classe y_j .

Cette probabilité *a posteriori* est assez simple mais difficile à mettre en pratique (*généralement les probabilités a posteriori sont inconnues, donc elles doivent être estimées*), elle est donnée par la loi de Bayes [29] (son auteur Thomas Bayes, publié à titre posthume en 1764) qui permet de calculer la probabilité d'une variable sachant la valeur d'une autre variable :

$$P(y_j/X) = \frac{P(X/y_j) * P(y_j)}{P(X)}$$

- $P(y_j)$: probabilité d'apparition *a priori* de la classe y_j , elle est souvent déduite de l'expérience d'un expert.
- $P(X/y_j)$: probabilité *a posteriori* d'appartenance de X à la classe y_j , elle est estimée à partir d'une modélisation mathématique des classes d'après les mesures effectuées sur l'ensemble des observations.
- $P(X) = \sum_{i=1}^k P(X/y_i) P(y_i)$: densité de probabilité du mélange, qui sert de normalisation de la distribution des probabilités sur les différentes classes.

Sachant les probabilités *a priori* de certaines variables nous pouvons calculer les probabilités *a posteriori* des autres variables. Cette loi donne une caractéristique de symétrie aux probabilités conditionnelles.

3.1.1. Apprentissage Bayésien

Dans une classification Bayésienne l'apprentissage est défini par la construction de ses tables de probabilités conditionnelles, et cela à partir des connaissances apprises par un expert et/ou par un outil d'apprentissage automatique. Deux formes d'apprentissage sont possibles : nous avons l'apprentissage de structure définie par l'initialisation des nœuds et des relations qu'ils entretiennent, à partir des données d'apprentissage ; et l'apprentissage de paramètres qui consiste à la définition des tables de probabilités conditionnelles relatives à chaque nœud. Nous avons une étude bien détaillée de l'apprentissage Bayésien dans le travail de [59].

3.1.2. Classificateur Bayésien Naïf [NaïveBayes]

Le réseau naïf de Bayes a été introduit par Duda et Hart pour la reconnaissance des formes en 1973, c'est la forme d'application la plus simple des réseaux Bayésiens. Ces derniers n'ont pas été bien exploités jusqu'après avoir démontré leurs grande efficacité par plusieurs travaux de recherches récents.

La simplicité de construction d'un classificateur Bayésien *naïf* est issue de sa simple structure graphique représentative avec une forme fixe et unique construite *a priori* par un expert du domaine d'application. Alors, la base d'apprentissage est bien définie par cette structure graphique, également pour les liens d'indépendances conditionnelles entre les attributs d'individus et les classes. Le graphe *causal* représentatif d'un classifieur Bayésien *naïf* ne peut inclure que deux types de nœuds : un nœud racine (*parent*) correspondant à une classe, et des nœuds d'attributs x_i caractérisant un individu X . Par conséquent les seules relations possibles du graphe sont de *classe* vers *attribut*.

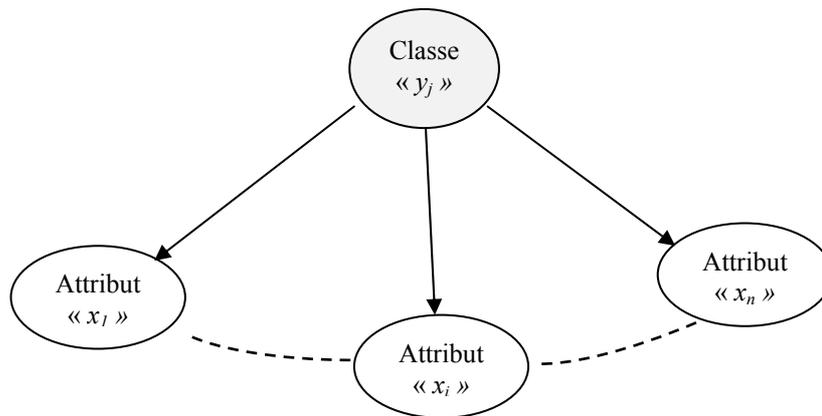


Figure 2.1 : Structure d'un réseau Bayésien *naïf*.

L'objectif est de chercher la classe d'appartenance y_j la plus probable pour chaque individu X défini par ses attributs ($x_1 = a_1, x_2 = a_2, \dots, x_n = a_n$). En d'autre terme, déterminer y_j tel que $P(y_j/X) \geq P(y_k/X) \forall k \in \{1..c\}$, en utilisant la loi de Bayes comme suit :

$$\begin{aligned}
 P(y_i/X) &= \frac{P(X/y_i) * P(y_i)}{P(X)} \\
 &= \frac{P(x_1=x_1/y_i) * \dots * P(x_i=x_i/y_i) * \dots * P(x_n=x_n/y_i) * P(y_i)}{P(X)}
 \end{aligned}$$

De nos jours, le classificateur Bayésien *naïf* est très concurrentiel à d'autres classificateurs dans le domaine de la classification dynamique. Il devient de plus en plus performant, surtout, lorsque les attributs de l'individu sont effectivement indépendants, or ces performances deviennent plus modestes devant des attributs impertinents [56].

3.1.3. Champs de Markov

Les champs de Markov (*introduits par le mathématicien russe Andrei Andreevich Markov 1856-1922*) sont des processus stochastiques simple en applications, riches en propriétés, fondés sur des bases mathématiques solides. Ils sont largement utilisés en segmentation d'images, reconnaissance des formes, reconnaissance de la parole et la reconnaissance de l'écriture [3][43][45].

Dans ce processus stochastiques chaque individu du modèle de classification est modélisé par un vecteur aléatoire indexée X_s appartenant à l'espace de représentation du processus (section_2). X_s est associée à l'état s défini dans un espace *discret* des états noté $S = \{s_i, 1 \leq i \leq n\}$, cette espace contient tous les états possibles du processus. Etant donné Y comme un processus d'étiquetage aléatoire sur l'ensemble S , alors Y est un **champ de Markov** relativement au système de voisinage ϑ si et seulement si : « la probabilité conditionnelle locale de Y en un état n est fonction que de la configuration du voisinage de l'état considéré » [45].

Nous allons définir un espace de voisinage ϑ au sein de l'espace S comme suit [33]:

$$\text{Soit } s, t \in S \text{ et } \vartheta_s, \vartheta_t \subset S, \text{ alors : } \vartheta_s = \{t\} \text{ tels que } \begin{cases} s \in \vartheta_s \\ t \in \vartheta_s \Rightarrow s \in \vartheta_t \end{cases}$$

Alors, la méthode de classification markovienne a pour but d'estimer *les champs des étiquettes* $Y = \{Y_s, s \in S\}$ en fonction de son voisinage, à partir des réalisations du vecteur aléatoire $X = \{X_s, s \in S\}$, appelé *champ des observables*. La probabilité *a posteriori* d'un tel processus est données par la loi de Bayes suivante :

$$P(Y_s = y_s / X_s = x_s) = \frac{P(X_s = x_s / Y_s = y_s) * P(Y_s = y_s)}{P(X_s = x_s)}$$

On appliquant la propriété Markovienne nous avons :

$$P(Y_s = y_s) > 0, \forall s \in S,$$

$$P(Y_s = y_s / Y_t = y_t, t \neq s) = P(Y_s = y_s / Y_t = y_t, t \in \vartheta_s)$$

$$P(X_s = x_s) > 0, \forall s \in S,$$

$$P(X_s = x_s / X_t = x_t, t \neq s) = P(X_s = x_s / X_t = x_t, t \in \vartheta_s)$$

Cela signifie que la probabilité en un état s conditionnellement au reste du champ, est égale à la probabilité en cet état connaissant uniquement ses voisins. Ainsi, l'importance d'une classification Markovienne est de pouvoir spécifier *localement* une modélisation des *propriétés globales*.

3.1.3.1. Lien entre Champs de Markov & Champs de Gibbs

Le lien entre les champs de Markov et les champs de Gibbs est défini par le théorème de Hammersley-Clifford [45]. Ce théorème permet de faire le lien entre la propriété de localité des champs de Markov et la propriété de globalité des champs de Gibbs.

La mesure de Gibbs est définie par la probabilité suivante:

$$P(X = x) = \frac{1}{Z} \exp(-U(x)),$$

- Où U est une fonction d'énergie définie par une somme de potentiels sur les cliques C :

$$U(x) = \sum_{c \in \mathcal{C}} U_c(x)$$

(\mathcal{C} : est le système des cliques associé au système de voisinage \mathcal{G} de U).

- Z est une constante de normalisation sur l'ensemble des réalisations de Ω de la fonction d'énergie, définie par :

$$Z = \sum_{x \in \Omega} \exp(-U(x))$$

En résumé, X est un champ de Markov par rapport au système de voisinage \mathcal{G} si et seulement si sa distribution $P(X)$ est une distribution de Gibbs:

$$P(X = x) = \frac{1}{Z} \exp(-U(x)) = \frac{1}{Z} \exp\left(-\sum_{c \in \mathcal{C}} U_c(x)\right)$$

- c : est une clique - un site isolé (*singleton*) ou un ensemble de sites, tel que tous les couples de sites extraits de cette clique soient voisins.
- \mathcal{C} : est l'ensemble des cliques liées au système de voisinage considéré.
- $U_c(x)$: est une fonction de potentiel.

Dans une configuration donnée, chaque clique c possède une énergie locale U_c , la somme des énergies des cliques donne l'énergie globale, l'objectif est de trouver le modèle optimale qui minimise cette énergie [45]. La classification est donc un problème de minimisation des termes d'énergie, qui peut être facilement résolu par des algorithmes déterministes comme le recuit simulé ou l'algorithme des Modes Conditionnels Itérés (ICM) détaillés dans le travail de [33], où l'auteur a bien exposé plusieurs modèles des champs de Markov (*le modèle d'Ising, le modèle de Potts et le modèle Markovien gaussien*).

3.1.4. Algorithme des k plus proches voisins

Introduit par Cover et Hart, la méthode des k plus proches voisins (k-ppv) (*k-Nearest-Neighbors -kNN*) est largement utilisée en classification d'une manière générale et en segmentation d'images en particulier. C'est une des méthodes non paramétriques les plus simples, où elle consiste à regrouper les individus X en fonction de leur voisinage [30]. D'abord, nous examinons les k plus proche voisins du nouvel individu X à classer dans l'ensemble d'apprentissage Ω , puis nous affectons X à la classe la plus représentée parmi ses k plus proches voisins.

La règle de décision classique de la méthode des k plus proches voisins est simple, elle se base essentiellement sur les deux éléments suivants:

- **la distance** : détermine la forme du voisinage de chaque vecteur X pour mesurer le plus proche voisin. Une métrique est nécessaire pour déterminer ces distances, et le choix d'une métrique est très important car elle influe beaucoup sur la qualité des prédictions, en plus, il n'existe pas de définition préalable sur ce qui constitue une bonne métrique, le choix est toujours arbitraire. « *Généralement, nous pouvons associer une forme en sphère avec la distance euclidienne, un cube avec la distance de Manhattan et enfin une ellipsoïde avec la distance de Mahalanobis* » [8].
- **La valeur de k** : spécifique à chaque utilisation de l'algorithme *kpp*, il détermine le nombre de cas existants pour prédire un nouveau cas. Généralement, k est arrêté lorsqu'une valeur supérieure ne modifie pas la classification.

La probabilité conditionnelle d'appartenance de X à une classe y_j est estimée par la fréquence de présence de la classe y_j parmi les k plus proches voisins de X . Elle s'obtient comme pour la discrimination par boules [57]: le principe est de tracer autour du point X , une boule de rayon donné dans R^p . On compte ensuite le nombre k_j d'exemples de l'ensemble d'apprentissage de la classe y_j dans cette boule.

Donc, la règle de décision de Bayes sera utilisée pour estimer l'appartenance d'un nouveau individu X à la classe y_j , selon la probabilité conditionnelle $P(y_j/X)$ définie par :

$$P(y_j/X) = \frac{k_j}{\sum_{j=0}^c k_j}$$

L'algorithme de classification par la méthode des k plus proches voisins est décrit ci-dessous :

Algorithme des k plus proches voisins pour classifier le vecteur X

Début

1. Choisir « k » le nombre des voisins (entier entre 1 et n , où n est le nombre maximum des voisins);
2. Calculer les distances $d(X, X_i)$, $i = 1 \dots n$ (où X_i est un des n vecteurs de l'espace de représentation);
3. Retenir les k vecteurs pour lesquels les distances sont les plus petites;
4. Compter le nombre de fois k_1, \dots, k_m que ces k vecteurs apparaissent dans chacune des m classes;
5. Attribuer X à la classe la plus représentée dans son entourage;

Fin.

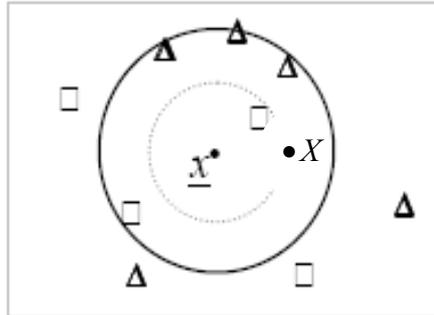


Figure 2.2 : Exemple de classification par la méthode k -ppv [8].

D'après l'exemple de la figure 2.2, la classe d'appartenance de l'individu X dépend du choix de k :

- Si $k = 1$, X sera classé "□",
- Si $k = 5$, X sera classé "Δ".

3.2. Arbres de Décision

C'est une représentation graphique d'une procédure de classification ou d'aide à la décision très efficace avec un formalisme compréhensible par tout utilisateur qui peut la traduire sous forme de règles. Les arbres de décision ont connus plus de succès dans le domaine du diagnostic médical (*analyse ECG*) [52], système expert et la reconnaissance de formes [12], classification des objets spatiales (*données multi dimensionnelles*) [16], ce qui les a rendu l'un des techniques de classification et d'apprentissage automatique les plus efficaces.

La classification avec les arbres de décision est basée sur la décomposition d'un processus de prise de décision en une séquence de décisions élémentaires plus simples à implémenter. Pour classer un nouveau individu, le classificateur va parcourir un chemin dans l'arbre, depuis la racine jusqu'à une feuille d'arrivée, en passant par un nœud de l'arbre l'individu doit satisfaire un test de décision afin de choisir sur quel chemin il va continuer son parcours.

Dans ce qui suit, nous allons représenter les différentes approches de classification et d'apprentissage basées sur les arbres de décision, précédé par un rappel succinct sur leur concept de base.

3.2.1. Principe des arbres de décision

Un arbre de décision est composé par des :

- Nœuds : (**Attributs**) correspondes aux propriétés de l'individu à classer.
- Feuilles : correspondes aux classes.
- Branches : correspondes aux valeurs possibles de l'attribut père.

Chaque individu à classer est constitué d'une liste d'attributs prédictifs. Le processus de classification consiste à trouver la prédiction de ces attributs afin de donnée le meilleur classement de l'individu [16]. Dès lors, les arbres de décision tentent de répondre à la question de base suivante : « *quels sont les meilleurs critères, à découvrir dans les données d'apprentissage, qui pourraient classifier le plus efficacement possible de nouvelles données?* »

Les partitionnements des données d'apprentissage en sous-ensembles de plus en plus petits, selon des critères de séparations, décrivent bien le concept de base des arbres de décision. Ce partitionnement regroupe les données les plus homogènes dans le sens qu'ils appartiennent à la même classe. La procédure de classification peut utilisée un critère de séparation sous forme de règles de décision « *Si condition_i Alors classe_j* ».

3.2.2. Construction de l'arbre de décision

L'arbre de décision est construit par la division récursive de la population d'apprentissage selon un critère de subdivision afin d'obtenir des sous populations plus homogènes. D'abord, nous devons définir la liste des attributs et des classes pour modéliser le processus de classification [16] :

- $\{X_1, X_2, \dots, X_n\}$: ensemble des attributs (*nœuds internes de l'arbre*) ;
- n : est le nombre d'attributs ;
- $\{C_1, C_2, \dots, C_c\}$: ensemble des classes (*feuilles de l'arbre*)
- c : est le nombre de classes ;
- $|C_j|$; $j=1,2,\dots,k$: nombre des exemples qui appartiennent à la classe C_j ;
- T : ensemble des exemples ;
- $|T|$: nombre total des exemples.

Dans un graphe d'arbre de décision on peut définir trois types de nœuds :

- Un seul nœud racine.
- Un ou plusieurs nœuds internes (*nœuds de décision*).
- Un ou plusieurs nœuds terminaux (*Classes*).

La structure de l'arbre de décision est définie par les tests de décision dans chaque nœud d'attribut X_i . Le choix des attributs de décision peut être basé sur un critère d'entropie et le critère ratio du gain. Une **Entropie** quantifie l'information nécessaire pour identifier la classe C_j dans l'ensemble T :

$$Info(T) = - \sum_{j=1}^k P_j \log_2(P_j)$$

Où : P_j est la probabilité de la classe C_j dans l'ensemble T , définie par : $P(C_j) = \frac{|C_j|}{|T|}$

Pour diviser l'ensemble d'exemples T , nous utilisons la formule de l'**Entropie conditionnelle** pour calculer l'information gagnante par l'attribut X_i :

$$\begin{aligned} Info(T/X) &= \sum_{i=1}^n P(T_i) \cdot Info(T_i) = - \sum_{i=1}^n \left(\frac{|T_i|}{|T|} \sum_{j=1}^k P_{ij} \log_2(P_{ij}) \right) \\ &= - \sum_{i=1}^n \left(\frac{|T_i|}{|T|} \sum_{j=1}^k \frac{|C_j|}{|T_i|} \log_2 \left(\frac{|C_j|}{|T_i|} \right) \right) \end{aligned}$$

Le **gain d'information** (l'information mutuelle) est définie par:

$$Gain(X,T) = Info(T) - Info(T/X)$$

A partir des gains d'information calculés pour chaque attribut, nous allons choisir l'attribut X_i qui le maximise (*i.e. qui minimise l'entropie*) pour qu'il soit utilisé comme un test. Cet attribut va partitionner l'ensemble T en sous ensembles sur plusieurs branches selon ses valeurs possibles. Pour chaque sous ensemble nous allons refaire le même processus de répartition jusqu'à un critère d'arrêt, où nous allons créer une feuille qui correspond à la classe la plus probable.

Le **Ratio du gain** (*Gain d'Entropie normalisée*) est un autre critère qui va corriger les points faibles du gain d'information. Dans ce cas, le choix d'un attribut test est basé sur la maximisation de ce ratio par la formule :

$$GainRatio(X, T) = \frac{Gain(X, T)}{SplitInfo(X, T)}$$

Où :

$$SplitInfo(X, T) = - \sum_{i=1}^n \frac{|T_i|}{|T|} \log_2 \left(\frac{|T_i|}{|T|} \right)$$

Exemple : nous avons extrait l'exemple suivant de [59].

Nous voulons classifier des objets graphiques dans l'une des classes : Objets2D, Objets3D. Chaque objet est décrit par deux attributs : Couleur \in {jaune, vert, bleu, rouge} ; Forme \in {rond, carré, triangle, losange, étoile}.

L'arbre de décision est construit comme suit :

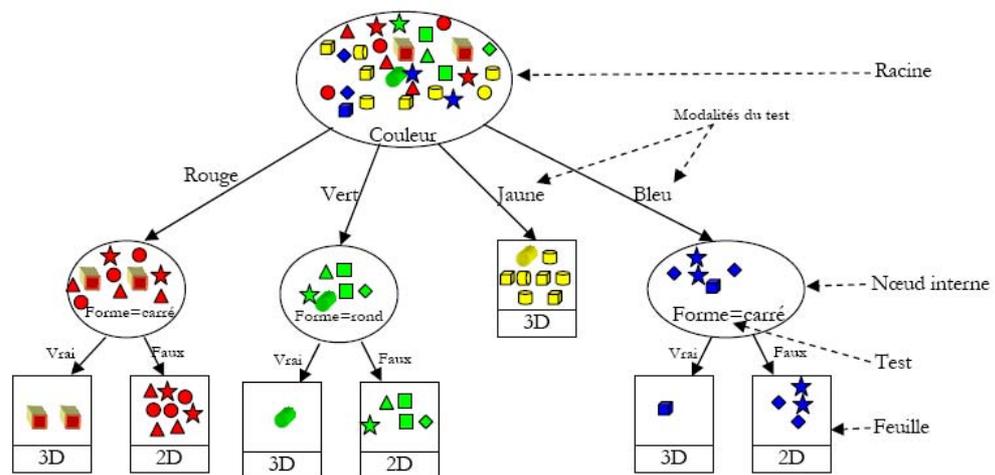


Figure 2.3 : Exemple d'arbre de décision [59].

3.2.3. Apprentissage des arbres de décision

Un arbre de décision est caractérisé par un apprentissage supervisé, où les classes d'appartenance sont connues *a priori*. L'idée principal de l'apprentissage, est de trouver le plus efficacement possible une série de répartitions à partir de l'ensemble des exemples, selon des tests sur les valeurs des attributs, jusqu'à l'obtention des sous-ensembles contenant uniquement (ou presque) des objets d'une même classe.

Une fois le modèle de répartition de l'ensemble est construit, nous allons extraire un ensemble de règles de classement pour classer des nouveaux individus dont la classe est inconnue [31].

3.2.4. Principaux algorithmes des arbres de décision

Dans la littérature, plusieurs méthodes de classification utilisant la modélisation par arbres de décision existent, selon la manière d'utilisation du critère de séparation. Nous allons présenter dans cette section les algorithmes les plus utilisés dans la pratique. La méthode ID3, ensuite l'algorithme C4.5 qui est le plus connu et enfin le modèle CART.

3.2.4.1. L'algorithme ID3 (Inductive Decision Tree)

Introduit par Quinlan en 1983, l'objectif de l'algorithme ID3 est de générer tous les arbres de décision possibles à partir de l'ensemble d'apprentissage afin de choisir l'arbre le plus simple et le plus correct pour le processus de classification. Le mécanisme de génération des arbres est fait d'une façon itérative jusqu'à l'obtention d'un arbre de décision fiable [31]. L'algorithme commence par une construction initial de l'arbre de décision à partir d'un sous ensemble d'apprentissage choisi aléatoirement de l'ensemble globale, et le choix de l'attribut test associé à un nœud se fait à l'aide de la fonction **Gain** ou **GainRatio** basé sur la fonction entropie. Ensuite, le processus de classification va utiliser cet arbre initial pour classer les nouveaux individus. Tant que l'arbre nous donne de bons résultats de classification, il sera maintenu. Or, s'il y aura une mauvaise classification, l'algorithme sera obligé de reconstruire un nouvel arbre avec l'ensemble des individus mal classés.

3.2.4.2. L'algorithme C4.5

C'est un résultat d'une amélioration de l'algorithme ID3 en 1993. Cette amélioration est constatée dans la possibilité de résoudre le problème du traitement des attributs continus, et le problème des valeurs manquantes ainsi que l'élagage d'un arbre de décision.

L'algorithme C4.5 est exécuté sur deux étapes : en premier lieu, nous avons la phase de construction de l'arbre de décision optimal jusqu'à sa taille maximale par une division récursive de l'ensemble d'apprentissage. L'algorithme utilise la fonction de **GainRatio** comme critère de séparation afin de favoriser les tests ayant un maximum de modalités. En second lieu, une phase d'élagage de l'arbre est appliquée. C4.5 va élaguer les branches ayant le plus de contribution dans l'augmentation du taux d'erreur de classification.

3.2.4.3. L'algorithme de CART (Classification And Regression Tree)

Développé par Breiman et al. en 1984, cet algorithme a une modalité de pouvoir supporter tous les types d'attributs (*binaires, continus à valeurs réelles, discrets ou qualitatifs, à valeurs dans un ensemble fini de modalités*).

CART est basé sur un arbre de décision binaire, où tous les tests étiquetant les nœuds de décision sont binaires. Pour le critère de subdivision l'algorithme utilise l'indice de *Gini*. Le processus d'élagage de l'arbre est fait par une estimation de l'erreur réelle en se basant sur un ensemble de tests différents avec les données d'apprentissage. Dès lors, l'utilisation de la validation croisée comme estimation de l'erreur pour l'élagage est favorable, car ce dernier se fait sur des échantillons de plus petite taille.

3.3. Réseaux de Neurones

Un réseau de neurones, introduit par les travaux Hebb (1949) et Rosenblatt (1958), est défini par un réseau d'unités élémentaires interconnectées entre eux, chaque élément du réseau est dénommé par *nœud*. Ces nœuds sont regroupés en plusieurs ensembles et chaque ensemble représente une *couche* du réseau. Il est convenu qu'un réseau de neurone est composé, au minimum, par deux couches (une couche *d'entrée* et une couche de *sortie*), comme il peut contenir en supplément un ensemble de couches *cachées*.

Le maillage du réseau est défini par les interconnexions complètes ou partielles entre les nœuds. Une connexion entrante d'un nœud est un lien qui converge vers ce nœud, parallèlement, une connexion sortante c'est un lien qui diverge du nœud. Un poids est associé à chaque connexion entre deux nœuds, afin d'estimer la force d'influence du nœud initial sur le nœud final. Le regroupement des poids de connexions détermine un vecteur de *poids synaptique*.

De nos jours, Les réseaux de neurones sont connus par leur grande capacité d'apprentissage et de classification automatique dans un environnement dynamique. Ils sont largement présents dans plusieurs disciplines de recherche qui portent sur l'apprentissage et la classification dynamique, telle que la reconnaissance de formes, segmentation d'image ou la bioinformatique.

Dans [1], l'auteur a introduit deux grands groupes de structure des réseaux de neurones selon la méthode d'apprentissage utilisé: pour un apprentissage supervisé nous avons les réseaux à *architectures statiques*, où le nombre de neurones de chaque couche est fixe (*inchangeable*) et connu *a priori*. Ces réseaux sont connus par le modèle PCM (*Perceptron Multi-Couche*), dont l'apprentissage est basé sur des règles de rétro-propagation.

Concernant l'apprentissage non supervisé, nous avons les réseaux à *architectures évolutives*, or ces réseaux peuvent aussi fonctionner en apprentissage supervisé. Dans ce cas, le nombre de neurones et de connexions est variable, ce qui définit un réseau plutôt dynamique capable d'évoluer dans le temps soit d'une façon constructive ou destructive.

Par la suite, nous allons d'abord présenter les grands aspects des réseaux de neurones : fonctionnement et apprentissage, pour ensuite introduire les deux modèles des réseaux de neurones à architecture artificielle et aléatoire.

3.3.1. Réseaux de Neurones Artificiels

Inspiré de leurs travaux de recherches portant sur les neurones biologiques, Warren McCulloch et Walter Pitts ont proposé un modèle de base des réseaux de neurones artificiels en 1943[9]. La classification par réseaux de neurones artificiels consiste à soumettre un individu à classer dans la couche d'entrée du réseau, afin de récupérer sa classe d'appartenance la plus probable à partir de la couche de sortie. L'initialisation d'un RNA est faite par le choix du type de réseau, type de neurones (*unités de calcul simple, nœuds de processeurs*), le nombre de couches et le choix de la méthode d'apprentissage.

3.3.1.1. Le Neurone Formel

Le neurone formel (*perceptron*) est l'élément de base d'un réseau de neurones artificiel. C'est un processeur élémentaire qui permet de:

- Calculer son état d'excitation interne σ en fonction de ses entrées avec leur poids :
 - $x = (x_1, \dots, x_n)$: est le vecteur d'entrée du perceptron « $x \in \mathbb{R}^n$ (Réel) ou $x \in \mathbb{B}^n$ (Booléen) »,
 - w_{ij} : est le *poids* d'entrée du neurone i vers le neurone j ,
 - $\sigma_j = \sum_{i=1}^n w_{ij}x_i$: est l'*état d'excitation* du neurone j ,

La Règle d'excitation : Si $\sigma_j \geq 0$, le neurone est à l'état excité ; Si $\sigma_j < 0$, il n'est pas excité .

Souvent, nous ajoutons une valeur fixée à 1 au vecteur d'entrée:

$$\sigma_j = w_{0j}x_0 + \sum_{i=1}^n w_{ij}x_i$$

Nous avons : $\sigma > 0 \Leftrightarrow \sum_{i=1}^n w_{ij}x_i > -w_{0j}x_0$

Donc le neurone sera excité si la combinaison linéaire de ses entrées dépasse le seuil $-w_{0j}$.

- Emettre un résultat $y=f(\sigma)$ en sortie, (f : fonction d'activation).

Les fonctions d'activation les plus classiques sont :

- Fonction Echelon : $f(\sigma) = \begin{cases} 1 & \text{si } \sigma \geq 0 \\ 0 & \text{si } \sigma < 0 \end{cases}$
- Fonction Signe : $f(\sigma) = \begin{cases} 1 & \text{si } \sigma \geq 0 \\ -1 & \text{si } \sigma < 0 \end{cases}$
- Fonction Sigmoidé : $f(\sigma) = \frac{1}{1+e^{-\sigma}}$

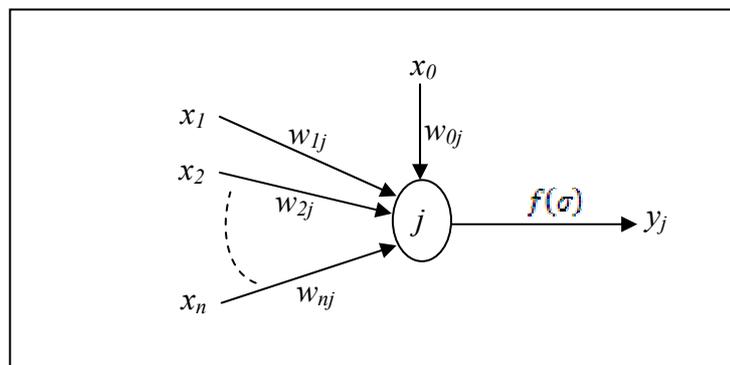


Figure 2.4 : Neurone formel.

3.3.1.2. Architecture des réseaux de neurones artificiels

C'est un ensemble de neurones formels interconnectés entre eux, chaque neurone peut avoir l'une des formes suivantes : une forme visible qui correspond aux neurones avec des valeurs de sortie connues (observables) ; ou la forme caché pour les neurones ayant un état interne inconnu. L'architecture d'un réseau de neurone va dépendre de la manière d'interconnection entre ses neurones visibles et neurones cachés.

Deux grandes classes de réseaux sont définies :

- *Réseaux à propagation directe* : c'est le modèle des réseaux PMC *perceptron multicouches* (Multi Layer Perceptron, MLP). Les connexions entre les neurones des différentes couches partent dans un seul sens. Nous avons d'abord une couche d'entrée avec seulement des entrées du réseau (*pas de neurone*), ensuite les couches cachées contenant des neurones cachés, et en dernier la couche de sortie englobant les neurones visibles.

- *Réseaux récurrents* : exemple réseau de Hopfield ou la machine de Boltzmann. Dans ce type de réseaux les neurones (*cachés ou visibles*) sont connectés dans n'importe quel sens d'une façon cyclique.

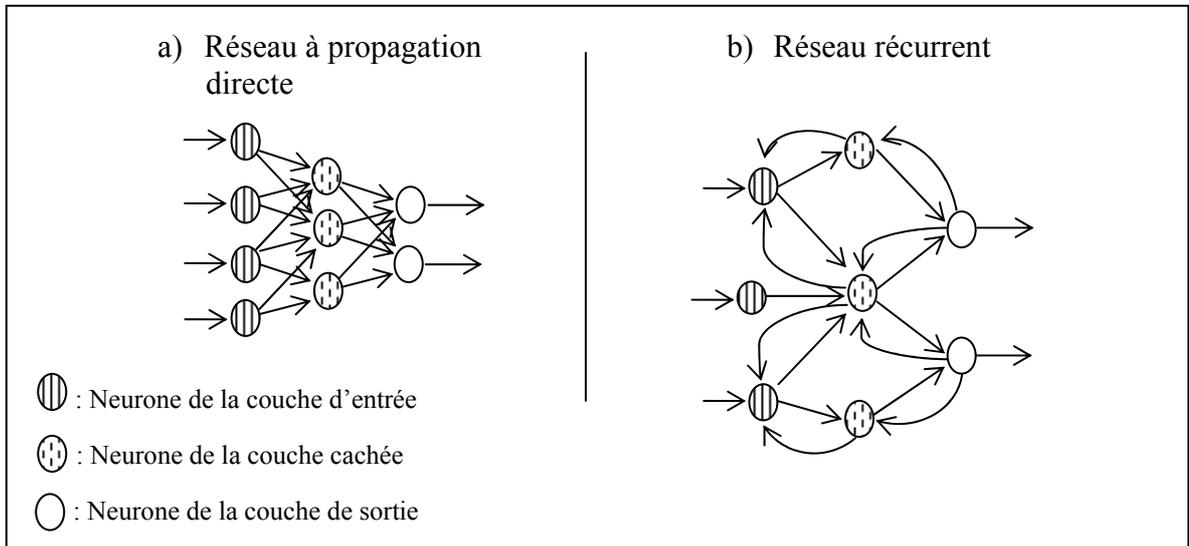


Figure 2.5 : Architectures d'un réseau de neurones.

3.3.1.3. Apprentissage des réseaux de neurones artificiels

La base d'apprentissage d'un réseau de neurone va contenir l'ensemble des exemples d'étiquetage (x, v) , où x représente le vecteur d'entrée réceptionné au niveau de la couche d'entrée, et v c'est le vecteur de sortie désiré sortant de la couche de sortie. L'objectif de l'apprentissage soit supervisé ou incrémental [32], est d'estimer les poids de connexion du réseau afin de rapprocher la sortie $y = f(x)$ du réseau de la sortie désiré v pour chaque exemple d'apprentissage.

En mots, minimiser l'erreur suivante :

$$E = \frac{1}{2} \sum_{i=1}^d (v_i - y_i)^2$$

3.3.2. Réseaux de Neurones Aléatoires (Random Neural Networks, RNN)

Défini comme un autre modèle des réseaux de neurones, les réseaux de neurones aléatoires sont fondés sur le modèle biologique, où les neurones élémentaires du réseau s'activent selon des signaux, *excitateurs* ou *inhibiteurs*, à des fréquences moyennes données au cours du temps [60]. Un signal exciteur augmente le potentiel du neurone qui le perçoit, contrairement au signal inhibiteur qui diminue ce potentiel.

Le RNA est constitué par un ensemble de neurones élémentaires aléatoires. Le neurone aléatoire est caractérisé par sa valeur potentiel à chaque instant t , cette valeur est augmentée (+ unité) quand le neurone reçoit un signal excitateur. Pareillement, elle est diminuée (- unité) quand le neurone aperçoit un signal inhibiteur. De même, lorsque le neurone transmet un signal (*excitateur* ou *inhibiteur*), après son excitation, vers d'autres neurones sa valeur de potentiel est diminuée.

En comparant les réseaux de neurones aléatoires avec les réseaux de neurones classiques, nous allons constater qu'ils peuvent appliquer les mêmes types d'architectures (i.e. *architecture à propagation directe* et *architecture récurrente*). Sauf en apprentissage, les RNA gardent le même objectif d'optimisation de la fonction d'activation $f(x)$ pour avoir le vecteur de sortie désiré, mais ils utilisent des poids de connections divisés en deux matrices : $W+$ (*matrice des poids de connections excitatrices*) ; et $W-$ (*matrice des poids de connections inhibitrices*). L'apprentissage dans les réseaux de neurones aléatoires peut être fait suivant plusieurs algorithmes, les plus connus sont *L'algorithme de rétropropagation du gradient de l'erreur*, et *l'algorithme d'apprentissage par renforcement* [60].

4. Méthodes non supervisées

Dans les sections suivantes nous allons présenter un ensemble d'algorithmes de classification non supervisée les plus abordés dans la littérature. Cependant, nous avons une base d'apprentissage qui contient seulement des individus ou des objets non étiquetés, nous ne connaissons pas leurs classes d'appartenances *a priori*.

Les méthodes de classification non supervisée (*méthodes de Clustering*) vont diviser cette base d'apprentissage en différents groupes plus au moins homogènes nommés classes (*clusters*). La division de l'espace d'apprentissage est basée souvent sur un critère de similarité entre les objets [10].

Par la suite, nous définissons le principe de l'algorithme *C-moyennes* qui va être la base de tous les autres algorithmes de la classification non supervisée, pour passer ensuite à la présentation des algorithmes de classification flous.

4.1. Algorithmes des C-moyennes ("Hard C-Means" HCM)

La méthode HCM a pour objectif de définir un partitionnement de l'ensemble de données $X=(X_1, \dots, X_N)$ en C classes. Une donnée X_j est représentée par un vecteur d'attributs $\{x_{j1}, x_{j2}, \dots, x_{jn}\}$ dans un espace d'attribut de dimension n .

Une classe W_i est caractérisée par un centre P_i , qui va permettre de calculer le degré d'appartenance u_{ij} de l'objet X_j à cette classe. En finalité, chaque donnée X_j sera attribuée à une et une seule classe W_i parmi les C classes proposées.

Pour cela, à partir des centres des classes, l'algorithme va minimiser la fonction nommée *WGSS (Within Group Sum of Squared errors)* [3] suivante :

$$J(P, U, X) = \sum_{i=1}^C \sum_{j=1}^N u_{ij} d^2(X_j, P_i)$$

- $P = (P_1, \dots, P_C)$: l'ensemble des centres de classes, avec $P_i = \frac{\sum_{j=1}^N u_{ij} X_j}{\sum_{j=1}^N u_{ij}}$;
- $d^2(X_j, P_i) = \sum_{k=1}^n (x_{jk} - P_{ik})^2$: distance euclidienne entre la donnée X_j et le centre de la classe W_i ,
- $U = [u_{ij}]$: la partition recherchée de l'ensemble X , c'est la matrice des degrés d'appartenance du modèle de classification. Nous avons :

$$u_{ij} = \begin{cases} 1 & \text{si } d^2(X_j, P_i) < d^2(X_j, P_k) \quad \forall k \neq i \\ 0 & \text{sinon} \end{cases}$$

L'algorithme des C -moyennes est illustré comme suit :

Algorithme HCM [3]

Début

1. Initialisation : fixer C et initialiser la partition U_0 ;

2. Répéter

a. Evaluation des regroupements : évaluer les centres P_i ;

b. Adaptation des prototypes : calculer les nouveau degrés $u_{ij}(t)$ et mettre à jour la partition U_t ;

3. Jusqu'à le Test d'arrêt est réussi : $\Delta U < \varepsilon$;

Fin

Remarque : Pour les sections suivantes, nous allons garder la même terminologie définie dans le HCM.

4.2. Classification non supervisée floue

Le C -moyennes partitionne l'ensemble des données en un nombre C (*fixé au départ*) de classes *disjointes*. Le but est d'avoir une partition où chaque élément de l'espace de représentation est associé à une et une seule classe. Néanmoins, le débordement des classes entre elles pose un grand problème dans la prise de décision sur l'appartenance des données déjà associées à ces classes croisées. Afin de surpasser cette difficulté, l'introduction du concept *flou* nous permet de traiter l'incertitude d'appartenance des données à plusieurs classes distinctes, selon un certain degré d'appartenance à un sous-ensemble flou. Il existe plusieurs dérivées de cette méthode HCM définies sous le chapeau des algorithmes de classification non supervisée floue.

4.2.1. Théorie des ensembles flous

Introduite en 1965, par Lotfi A. Zadeh, [8][3], le principe de la théorie est la représentation des connaissances *imprécises* et *incertaines*. Cette théorie est basée sur deux autres théories : la théorie des sous-ensembles flous qui correspond au traitement de l'imprécision des données ; la théorie des possibilités qui modélise plutôt l'incertitude des données. Contrairement à la méthode classique (HCM) où les degrés d'appartenance u_i à une classe W_i sont définies dans l'ensemble $\{0,1\}$, une méthode floue a pour objectif d'éviter la transition directe de u_i de la valeur 0 à la valeur 1, en le définissant sur l'intervalle $[0,1]$.

Donc, un sous-ensemble flou L est constitué de données X_j ayant un degré d'appartenance $u_L(X_j) > 0$. Tant que $u_L(X_j)$ tend vers 1, tant que X_j est plus proche de L . La partition du sous-ensemble flou L est défini par la connaissance des couples (X_j, u_L) .

4.2.2. Algorithme C-moyennes floues ("Fuzzy C-Means" FCM)

Développé par Dunn en 1973 et approuvé par Bezdek en 1981, c'est une méthode de classification non supervisée qui permet d'associer une donnée à une ou plusieurs classes en même temps. L'appartenance d'une donnée à une classe est définie par un degré d'appartenance qui représente l'élément de base d'une partition U définie en (section 4.1). L'objectif de FCM est de construire une matrice de C -partition flou $U: \mathbb{C} \times \mathbb{N}$, ayant des éléments numériques dans l'intervalle $[0,1]$, ces éléments représentent les degrés d'appartenance u_{ij} des données X_j aux classes W_i .

L'objectif de l'algorithme FCM est défini par la minimisation de la somme pondérée des carrés des distances entre les données à regrouper et les centres de classes, décrite ci-après :

$$J(P, U, X) = \sum_{i=1}^C \sum_{j=1}^N (u_{ij})^m d^2(X_j, P_i)$$

Où :

- $m \in [1, \infty[$: est un indice de flou « *fuzzy index* » qui détermine le degré de flou de la partition obtenu. Dans [45], le choix de l'indice m est souvent compris entre 1.1 et 3, autrement, le théorème de *Tuker* suggère de prendre $m \geq N / (N-2)$, N étant le nombre de données. Tant que m tend vers ∞ la partition est de plus en plus floue.
- Dans cet algorithme la distance euclidienne entre la donnée et le centre de classe est définie par :

$$d^2(X_j, P_i) = (X_j - P_i)^T A (X_j - P_i)$$

Avec : $P_i = \frac{\sum_{j=1}^N (u_{ij})^m X_j}{\sum_{j=1}^N (u_{ij})^m}$; et $A : n \times n$ est une matrice définie positive.

Les degrés d'appartenances sont calculés par la manière suivante :

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{d^2(X_j, P_i)}{d^2(X_j, P_k)} \right)^{\frac{1}{m-1}}}$$

L'algorithme des C -moyennes floues est illustré comme suit :

Algorithme FCM [3]

Début

1. Initialisation: fixer C & m , initialiser la matrice floue U_0 ;
2. **Répéter**
 - 2.1. Adaptation des prototypes : évaluer les C centres P_i ;
 - 2.2. Evaluation des regroupements : calculer les nouveaux degrés u_{ij} et mettre à jour U ;
3. **Jusqu'à** le Test d'arrêt est réussi : $\Delta U < \varepsilon$;

Fin

4.2.3. Algorithmes de C-moyennes possibilistes ("Possibilistic C-means" PCM)

Cet algorithme est une nouvelle modélisation possibiliste de l'algorithme C-moyennes introduite par Krishnapuram et Keller [45] afin d'améliorer les performances de classification en présence du *bruit*. Il est basé sur la théorie de possibilité pour définir la partition U des données X sur l'ensemble des classes W représentées par leur centres P. En effet, les degrés d'appartenances u utilisés en FCM sont traduits en PCM autant que degrés de vérité relatifs, décrivant l'appartenance d'une donnée à chacune des classes possibles. La nouvelle formulation de la fonction objective à minimiser est décrite ci-après:

$$J(P, U, X) = \sum_{i=1}^C \sum_{j=1}^N (u_{ij})^m d^2(X_j, P_i) + \sum_{i=1}^C \eta_i \sum_{j=1}^N (1 - u_{ij})^m$$

Où :

- η_i : est le carré de la distance entre P_i (centre de la classe W_i) et l'ensemble des données X_j ayant leur degrés d'appartenance (à la classe W_i) $u_{ij} = 0.5$; C'est un paramètre positif utilisé pour évaluation des degrés d'appartenance possibles aux classes. Il peut être défini de plusieurs manières, soit estimer à l'initialisation de l'algorithme ou recalculer pendant l'itération de l'algorithme. Dans [3] il est défini par la formule suivante:

$$\eta_i = \frac{\sum_{j=1}^N (u_{ij})^m d^2(X_j, P_i)}{\sum_{j=1}^N (u_{ij})^m}$$

- Les centres de classes P_i sont calculés de la même manière dans FCM ;
- Contrairement au FCM, où la somme des degrés d'appartenance doit être égale à 1, en PCM, la somme des possibilités d'appartenance de données aux classes n'est pas forcément égale à 1.

La mise à jour des valeurs de possibilité d'appartenance est faite de la manière suivante :

$$u_{ij} = \frac{1}{1 + \left(\frac{d^2(X_j, P_i)}{\eta_i} \right)^{\frac{1}{m-1}}}$$

- Dans [45], le bon choix de l'indice m est conseillé par Krishnapuram et Keller à ce qu'il soit ~ 1.5 , contrairement à 2 dans le FCM ;

Le premier terme de la nouvelle fonction PCM correspond au critère du FCM. Le deuxième terme représente la nouveauté apporté par le PCM dont sa minimisation va contribuer pour avoir des degrés d'appartenance de plus en plus élevés.

4.2.3. Algorithme des K -plus proches voisins flou

C'est une autre variante des algorithmes de classification floue introduite dans [8], qui parte sur le même principe de travailler avec des données *incertaines* ou *imprécises*. Certainement, ce n'est pas le même type de méthodes traitées dans cette partie (*Méthodes non supervisées*), mais c'est plutôt une méthode supervisée. Nous avons voulu la présenter, vu qu'elle introduit la notion du **flou** dans les règles de décisions de sa version classique nommée : algorithme des k plus proches voisins (*section 3.1.5*).

Après le choix du paramètre k (*nombre de voisins*), l'initialisation de l'algorithme passe par la détermination des degrés d'appartenance $u_i(X) \in [0,1]$ pour chaque classe C_i de la base d'apprentissage, tel que $\sum_{i=1}^c u_i(X_j) = 1$; avec $j \in \{1,2, \dots, N\}$. Ces degrés peuvent être fixés par des experts ou calculés. Nous avons la proposition de Keller [8] qui calcule un degré d'appartenance à une classe C_i par la formule suivante :

$$u_i(X) = \frac{\sum_{j=1}^N u_{ij} \left(\frac{1}{d(X, X_j)^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^N \left(\frac{1}{d(X, X_j)^{\frac{2}{m-1}}} \right)}$$

Où :

- $d(X, X_j)$: c'est la distance entre X et son $j^{\text{ième}}$ plus proche voisin ;
- m : c'est un paramètre d'ajustement du degré d'appartenance ;
- u_{ij} : c'est le degré d'appartenance du $j^{\text{ième}}$ plus proche voisin de X à la classe C_i ;

L'objectif de la méthode k -ppv floue, est d'associer chaque nouvelle observation X à la classe dont le degré d'appartenance est maximal. En mots, X est classée dans C_i ssi :

$$u_i(X) = \max_{i=1, \dots, c} (u_i(X))$$

5. Conclusion

Nous pouvons conclure cette partie d'étude qui est censé d'être la plus fondamentale, dans la globalité de notre travail, pour dire qu'il n'existe pas une méthode de classification dynamique optimal. L'évaluation des performances des méthodes de classification n'est pas une tâche facile, chaque méthode se trouve plus avantagée que l'autre selon les objectifs attendus de son application dans divers domaines. La qualité de la classification peut être jugée selon les deux critères suivants: la grande ***divergence*** entre tous les classes de la partition ; et la forte ***homogénéité*** des données dans chaque classe. Afin de tirer avantages de chaque une des méthodes, la coopération des classificateurs est une solution très envisageable, qui a donnée de bonnes résultats dans plusieurs travaux avec différents approches [29][44].

Le présent chapitre a fait l'objet d'une étude sur les méthodes de classification dynamique les plus abordées dans la littérature, qui s'étendent des années 40 jusqu'à nos jours. En premier lieu, nous avons défini une modélisation pour la problématique de classification sur trois espaces (*mesure, représentation et interprétation*). Ensuite, l'étude des modèles de classification a été divisée en deux parties, celle qui concerne les méthodes supervisées, suivi par les méthodes non supervisées. En dernier lieu, nous avons présenté les méthodes de classification floues.

Nous venons de présenter quelques méthodes ayant des capacités à modéliser les processus d'évolutions en environnement dynamique. Dans le chapitre suivant de ce mémoire, nous allons présenter les principes de fusion de données. Introduction des théories de bases avec le concept des sous-ensembles flous, ainsi les méthodes de fusion des données fondées sur chaque théorie.

CHAPITRE 3

FUSION DE DONNEES

1. Introduction

Le concept de fusion des données a été toujours présent pour servir à l'être humaine comme un moyen pour faciliter son processus d'analyse, dans le but de perfectionner sa prise de décision en cours de sa vie quotidienne. La méthode de fusion des données est généralement utilisée dans les systèmes d'extraction de connaissance, systèmes décisionnels et classification dynamique, dans le but de faire face à la problématique d'usage de données multimodales issues de divers sources ou capteurs externes avec des formes et types d'informations (*attributs*) différents.

C'est une méthode élémentaire dans le métier de chaque expert qui veut diagnostiquer une anomalie ou un processus dynamique nécessitant une analyse de plusieurs données multimodales (*textes, images, sons, vidéos, etc.*). Chez un médecin, une fusion de données est faite automatiquement à partir d'un bilan médical comportant plusieurs analyses sanguines et de radiographie [47]. Dans [46], nous avons apprécié le travail de l'auteur concernant les systèmes biométriques multimodales, qui sert à identifier une personne selon un ensemble de signatures biométriques, capturées via plusieurs capteurs biométriques (*Empreinte digitale, Iris, forme du visage, signature, ...etc.*), ce système est dans le besoin fondamentale d'utiliser une méthode de fusion de données, dans la mesure d'avoir une seule donnée formelle finale à partir de différentes informations spécifiques initiales.

Initié par Zadeh, Shafer et Dempster dans les années 1960 [39], les modèles mathématiques de la fusion ont connus plus d'amélioration dans les années 80, où ils étaient plus utilisés dans différents domaines technologiques de pointe (*technologie spatiale, la robotique, domaine médical, militaire...etc.*).

Dans cette partie, nous allons présenter les trois théories de bases pour la fusion de données permettant la manipulation des données et des connaissances *incertaines* et/ou *imprécises*, afin de définir par la suite, les différentes méthodes et techniques de fusion selon chaque théorie.

2. Théories possibles de fusion de données

La fusion de données a pour objectif d'associer un élément x à une classe C_i à partir d'informations f_j issues de différentes sources. Ces informations sont imprécises ou incomplètes, leur modélisation fait appel aux différentes théories de base qui peuvent fournir un cadre mathématique plus adéquat à la représentation symbolique de l'information *incertaine* et/ou *imprécise*. Dans la littérature, la fusion de données peut être envisagée dans trois cadres théoriques : la théorie des probabilités, la théorie des possibilités et la théorie des croyances. Le choix de l'une d'entre elles est souvent conditionné par la facilité de sa mise en œuvre. Nous représentons par la suite, d'une manière succincte, chacune de ces trois théories.

2.1. La théorie des probabilités (théorie de l'incertain)

L'interprétation du hasard dans les jeux, fut être le premier contexte où la notion de probabilité a été introduite (*choisir une valeur parmi un ensemble de valeurs possibles*) par Pascal (1623-1662) [20]. Il est toujours possible de modéliser une fusion de données *incertaines* par une modélisation probabilistes. La méthode de fusion Bayésienne (*la distribution de probabilité conditionnelle*) est la plus connue, où elle illustre bien le lien entre les hypothèses $\{H_1, \dots, H_N\}$ estimées et l'information des données I_j .

En d'autre terme, Il s'agit de déterminer la probabilité *a posteriori* qu'un événement H_i parmi les N hypothèses connaissant l'information I_j , en appliquant la règle de Bayes sur les probabilités conditionnelles (*section 2.3.1*), comme suit :

$$P(H_i/I_j) = \frac{P(I_j/H_i)P(H_i)}{\sum_{k=1}^N P(H_k)P(I_j/H_k)}$$

Où :

- $P(H_i)$: c'est la probabilité *a priori* que l'hypothèse H_i soit réalisée, elle est évalué à partir d'une base d'apprentissage initialisée par un expert;
- $P(I_j/H_i)$: la probabilité d'observation de l'information I_j lorsque l'hypothèse H_i est réalisée, elle est déterminée par des modèles statistiques ou estimée à partir des méthodes d'apprentissage.

2.2. La théorie des croyances

Les origines de la théorie des croyances ont été fondées par Art Dempster [48] dans ses travaux de recherches qui portent sur les probabilités inférieures et supérieures. Mais historiquement, elle est introduite par Glenn Shafer en 1976 [20] qui a proposé une nouvelle interprétation des probabilités inférieures (*dans les travaux de Dempster*) à des degrés de croyance. En démontrant l'intérêt des fonctions de croyance pour la modélisation de connaissances *incertaines* par Shafer, la théorie des croyances fut être favorisée dans le domaine de fusion de données.

La théorie des croyances est fondée sur trois fonctions de bases pour la modélisation de données *incertaines* [7]. Cette théorie, qui consiste à quantifier la crédibilité attribuée aux faits observés, utilise ses fonctions de bases afin de représenter l'incertitude et l'imprécision d'une information, contrairement à la théorie des probabilités où la probabilité qu'une occurrence d'événement appartienne à une seule classe [51].

Dans la définition des trois fonctions qui suivent, nous proposons la modélisation suivante: D est un ensemble de n hypothèses d_i sur les quelles nous pouvons prendre une décision, où 2^D représente un ensemble de 2^n sous-ensembles A_j de D .

- La fonction de **Masse** m (*attribuée à chaque événement*) :

Une fonction $m : 2^D \rightarrow [0,1]$ est une fonction de masse si seulement si [66]:

$$\begin{aligned} m(\emptyset) &= 0 \\ \sum_{j=1}^{2^n} m(A_j) &= 1 \end{aligned}$$

La fonction de masse autorise un raisonnement sur des sous-ensembles A_j de l'ensemble des décisions possibles D . Un rapprochement de la notion de masse élémentaire à celle de probabilité est fait, lorsque les sous-ensembles A_j dont la masse est non nulle se réduisent aux singletons d_i . La théorie des croyances nous permet d'évaluer conjointement des ensembles d'hypothèses quelconques (exemple : évaluation des ensembles $d_i \cup d_j$ tq $\forall i,j \in [1,n]$). La fonction de masse $m(D)$ représente le **degré d'incertitude ou d'ignorance totale**, pareillement pour la masse $m(A_j)$ qui désigne la vraisemblance attribuable à l'un des éléments du sous-ensemble A_j .

Les deux fonctions de croyances qui vont suivre seront des nouvelles présentations de la fonction de masse.

- La fonction de **Crédibilité** Cr :

Une fonction $Cr : 2^D \rightarrow [0,1]$ est définie par :

$$Cr(\emptyset) = 0$$

$$Cr(D) = 1$$

$$Cr(A_k) = \sum_{A_j \subset A_k} m(A_j)$$

La fonction de crédibilité $Cr(A_k)$ est une mesure de confiance [20] accordée à A_k , elle prend en compte toutes les hypothèses qui **impliquent** A_k , alors si $Cr(A_k)$ est non nulle nous croyons à A_k .

L'utilisation et la définition des fonctions de masse élémentaire et des fonctions de crédibilité sont indépendantes, et chaque jeu de **masse** sur 2^D est associé à un jeu de **crédibilité** sur le même ensemble. Ceci détermine une bijection entre l'ensemble des fonctions de masse élémentaire et l'ensemble des fonctions de crédibilité.

- La fonction de **Plausibilité** Pl :

La fonction $Pl : 2^D \rightarrow [0,1]$ est définie par :

$$Pl(A_k) = \sum_{A_j \cap A_k = \emptyset} m(A_j)$$

En contrepartie de la fonction de crédibilité, la fonction de plausibilité $Pl(A_k)$ prend en compte toutes les hypothèses qui **ne contredisent pas** A_k [20]. Les deux fonctions Cr et Pl sont dites duales car une définition d'une fonction entre elles est identique à la définition de l'autre. Pour un sous-ensemble d'hypothèses A_k et son opposé $\overline{A_k}$ nous avons [8]:

$$Pl(A_k) = 1 - Cr(\overline{A_k})$$

Donc, plus la croyance est augmentée pour une hypothèse \rightarrow plus l'inverse de cette hypothèse deviens moins plausible. De même, la connaissance d'une fonction parmi (m , Cr , Pl) sur 2^D , permet la déduction des deux autres fonctions. Naturellement, une mesure de vraisemblance est une traduction de la crédibilité, et sa maximisation est une traduction de la plausibilité.

Dans [8], l'intervalle $[Cr(A_k), Pl(A_k)]$, encadre la probabilité mal connue $P(A_k)$ pour chaque A_k :

$$Cr(A_k) \leq P(A_k) \leq Pl(A_k)$$

2.3. La théorie des Possibilités

Améliorée par Dubois et Prade en France, après son introduction par Zadeh en 1978 [65], C'est une dérivée de la théorie des modèles flous [7] pour la représentation et le traitement de données *imprécis* et/ou *incertain*. Dans [51], elle est utilisée pour une manipulation des imprécisions et des incertitudes d'information pour un même formalisme.

Le passage par une définition du modèle flou est une étape incontournable dans une approche possibiliste, afin de présenter par la suite les fonctions de mesure utilisées dans cette théorie (fonction de *possibilité* et fonction de *nécessité*).

2.3.1. Le modèle flou

La théorie des ensembles flous a été introduite par Zadeh en 1965 [64]. Elle offre une grande possibilité de représentation explicite des informations *imprécises* [7]. La logique floue permet de définir l'affectation d'un élément x de l'ensemble X à un ou plusieurs sous-ensembles flous disjoints, avec des degrés d'appartenance différents relatifs à chaque sous-ensemble. Contrairement à la logique non floue, où l'élément x doit appartenir seulement à un et un seul ensemble classique. Ainsi, la théorie des ensembles flous permet de prendre en compte un grand intervalle de prédicats (*exemple dans la figure suivante avec les prédicats : « Lourd, Moyen, Léger »*) [20].

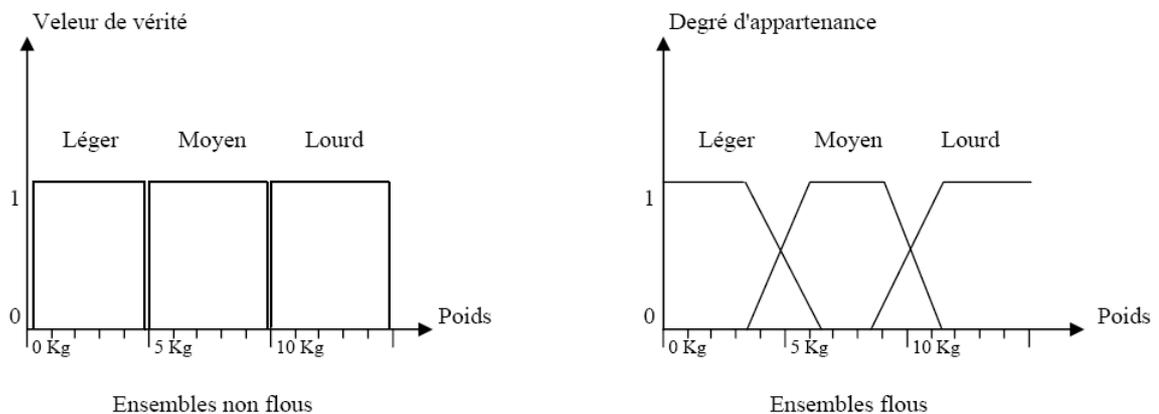


Figure 3.1 : valeur de vérité pour les théories des ensembles flous et non-flous.

Soit A un sous-ensemble flou, x est un élément de X , nous pouvons définir l'ensemble des fonctions sur les sous-ensembles flous comme suit [26]:

La fonction $f_A(x)$ définie le degré d'appartenance de x à A dans l'intervalle $[0,1]$:

$$f_A: X \rightarrow [0, 1]$$

- $f_A(x)$: est le degré d'appartenance de x à A .
- **supp**(A) (*support de A*): ensemble des éléments x ayant un degré d'appartenance non nul,

$$\text{supp}(A) = \{x \in X / f_A(x) \neq 0\}$$

- **h**(A) (*hauteur de A*) : est le plus fort degré d'appartenance de x à A ,

$$h(A) = \sup_{x \in X} f_A(x)$$

- Si $h(A) = 1$, alors A est dit **normalisé**.

- **noy**(A) (*noyau de A*): ensemble des sous éléments appartenant de façon absolue à A ,

$$\text{noy}(A) = \{x \in X / f_A(x) = 1\}.$$

Si nous reprenons l'exemple de la (*figure 3.1*), avec A = sous-ensemble flou « Moyen », alors nous avons les l'interprétation suivantes :

- $\text{supp}(A) = [3.5, 11.5]$;
- $h(A) = 1$ donc A est *normalisé* ;
- $\text{noy}(A) = [6,9]$.

Avant de continuer sur la présentation des différentes opérations des sous-ensembles flous, nous introduisons la terminologie suivante :

- A et B sont deux sous-ensembles flous de X .
- A est égale à B si pour tout élément x de X les deux fonctions d'appartenance f_A et f_B prennent la même valeur: $\forall x \in X ; f_A(x) = f_B(x)$
- A est inclus dans B si pour tout élément x de X appartenant à A doit appartenir également à B , avec un degré d'appartenance au moins aussi grand : $\forall x \in X ; f_A(x) \leq f_B(x)$

La théorie des ensembles flous est caractérisée par les opérations suivantes :

- **L'intersection** : c'est un sous-ensemble flou contenant les éléments de X ayant leurs plus petits deux degrés d'appartenance f_A et f_B :

$$\text{si } C = A \cap B, \forall x \in X ; f_C(x) = \min(f_A(x), f_B(x))$$

- **L'union** : c'est un sous-ensemble flou contenant les éléments de X ayant leurs plus grands deux degrés d'appartenance f_A et f_B :

$$\text{si } D = A \cup B, \forall x \in X; f_D(x) = \max(f_A(x), f_B(x))$$

- **Le complément A^c de A** : c'est un sous-ensemble flou de X défini par la fonction d'appartenance f_{A^c} suivante:

$$\forall x \in X; f_{A^c}(x) = 1 - f_A(x)$$

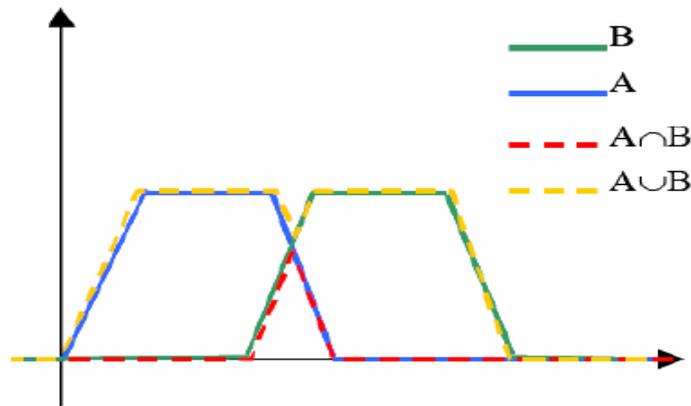


Figure 3.2: Exemple d'intersection et d'union dans les sous-ensembles flous [26]

2.3.2. Fonctions de mesure de la théorie possibiliste

La théorie des possibilités permet la représentation de l'imprécision et l'incertitude en se basant sur deux fonctions qui caractérisent les événements [7]: des mesures de *possibilité* Π ; et des mesures de *nécessité* N .

La possibilité Π définit un degré de préférence représentant la mesure la plus pessimiste ou la plus prudente [20]. Elle fournit une information sur l'occurrence d'un événement A relatif à un ensemble de référence X . Pour un ensemble de parties $P(X)$, la mesure $\Pi : P(X) \rightarrow [0,1]$ vérifie les points suivants :

- $\Pi(\emptyset) = 0$;
- $\Pi(X) = 1$;
- $\forall A_1 \in P(X), A_2 \in P(X), \dots$ nous avons : $\Pi(\bigcup_{i=1,2,\dots} A_i) = \sup_{i=1,2,\dots} \Pi(A_i)$;
- Si $\Pi(A) = 1 \rightarrow A$ est un événement tout à fait possible ;
- Si $\Pi(A) = 0 \rightarrow A$ est un événement impossible ;
- $\forall A \in P(X), \max(\Pi(A), \Pi(A^c)) = 1$ (i.e. pour chaque événement A et son complément A^c au moins l'un des deux est tout à fait possible) ;

La mesure de nécessité N définit le caractère prioritaire d'un événement [20]. Elle estime un degré *d'incertitude* avec lequel la réalisation de l'évènement A est certaine. Pour un ensemble de référence fini X et son ensemble de parties $P(X)$, la mesure $N : P(X) \rightarrow [0,1]$ vérifie les points suivants :

- $N(\emptyset) = 0$;
- $N(X) = 1$;
- $\forall A_1 \in P(X), A_2 \in P(X), \dots$ nous avons : $N(\bigcap_{i=1,2,\dots} A_i) = \inf_{i=1,2,\dots} N(A_i)$;
- Pour un ensemble de référence X , la mesure N peut être obtenue par la mesure Π correspondante, comme suit :

$$\forall A \in P(X) ; N(A) = 1 - \Pi(A^c)$$

Cependant, la maximisation de la *nécessité* de l'évènement A minimise la *possibilité* de son complément A^c , et maximise la certitude de la réalisation de A .

On appelle *distribution de possibilité* π , une fonction définie sur l'ensemble de référence X , où pour chaque évènement A elle associe un coefficient de degré de possibilité à valeurs dans $[0,1]$, qui satisfait la condition suivante :

$$\sup_{x \in X} \pi(x) = 1$$

Les mesures de possibilité et de nécessité sont calculées par cette fonction de *distribution de possibilité* π comme suit [7] :

$$\Pi(A) = \sup\{\pi(x), x \in A\}$$

$$N(A) = \inf\{1 - \pi(x), x \notin A\}$$

3. La Fusion de Données

Dans la littérature nous avons une variété de définitions pour le concept de fusion de données. La majorité des auteurs jugent avoir trouvé une facilité dans la compréhension de ce concept, mais une difficulté dans la formulation d'une définition adéquate. Voici quelques définitions extraites de différents travaux :

Dans [4] : « *La fusion de données homogène ou hétérogène peut donc être vue comme l'utilisation de données fournies par plusieurs sources d'informations pour construire une source de connaissance, plus précise et plus riche que les informations initiales prises indépendamment les unes des autres et ceci dans le but de prendre une décision ou d'effectuer un diagnostic ou de réaliser une meilleur estimation* ».

Dans [5], l'auteur a formulé la définition suivante : «*nous envisageons la fusion de données comme une agrégation d'informations ambiguës, conflictuelles, complémentaires et redondantes, autorisant une interprétation des données plus précise et/ou moins incertaine*».

Dans [66], l'auteur a opté pour cette définition : «*la fusion de données consiste à combiner des informations issues de plusieurs sources afin d'améliorer la prise de décision*».Après avoir cité les deux définitions suivantes :

Définition 1 : «un processus multi-niveaux à facettes multiples ayant pour objet la détection automatique, l'association, la corrélation, l'estimation et la combinaison d'informations de sources singulières et plurielles». [Proposée par le Joint Directors of Laboratories (JDL), du ministère de la défense aux Etats-Unis d'Amérique].

Définition 2 : «la fusion de données constitue un cadre formel dans lequel s'expriment les moyens et techniques permettant l'alliance des données provenant de sources diverses». [Proposée par le groupe européen SEE ; la branche française(IEEE) et la branche européenne(ISPRS)].

3.1. Problématique de la fusion de données

La fusion de données joue un grand rôle dans les systèmes d'extraction de connaissances, **classification multimodale** et les systèmes d'aide à la décision, où l'environnement d'application dispose d'un nombre important d'informations complexes, de nature et de fiabilité différentes provenant de sources hétérogènes (*étiquetage d'un expert ou une base d'apprentissage*). Le challenge de la fusion de données est de pouvoir surpasser les difficultés suivantes énumérées dans [4] :

- **Incertitude et Imprécision** : les données introduites par les différentes sources peuvent être *incomplète, imprécises, floues, incertaines ou inconsistantes (contradictoires)*.

- **Redondance et Complémentarité** : lorsque les différentes sources d'information fournissent des informations de la même nature sur le même phénomène observé, la qualité de précision et d'incertitude d'information est perfectionnée. Pareillement, lorsque ces sources introduisent des informations de différentes natures sur le phénomène observé, nous allons avoir une vue générale complémentaire de ce phénomène.
- **Concordance et Conflit** : les informations sont concordantes lorsqu'elles sont vraies simultanément sans aucune difficulté, sinon, elles sont en conflit (*leurs déclarations ne se concordent pas*).
- **Combinaison et Décision** : c'est l'objectif final de la fusion de données, où les informations concordantes, précises et certaines sont combinées sur la base de plusieurs critères de décision afin de déterminer le résultat de la fusion.

3.2. Le processus de la fusion de données

Chaque processus de fusion doit passer par quatre étapes de traitement, qui seront représentées ci-après [6] :

- **Représentation homogène et recalage des informations pertinentes**

Dans cette étape initiale, nous commençons par la transformation de certaines informations initiales hétérogènes vers des informations homogènes dans un espace de représentation commun de fusion ;

- **Modélisation des connaissances**

Cette seconde étape consiste à modéliser et évaluer la connaissance en provenance de chaque source d'informations, dans le but de pouvoir exploiter cette connaissance (*initialement imparfaite*) par la suite, et avoir plus d'information sur l'état du système ;

- **Fusion**

Dans cette troisième étape l'opération de fusion va être exécutée sous la forme d'une combinaison d'informations (*précédemment : représentées et modélisées*) selon des règles de combinaison relatifs au domaine théorique choisi ;

- **Décision par le choix d'une stratégie**

C'est l'étape finale du processus de la fusion de données. Après la détermination d'un critère de décision selon le cadre théorique choisi et l'objectif à atteindre, nous allons sélectionner l'information la plus vraisemblable parmi toutes les hypothèses possibles ;

3.3. Classification des opérateurs de fusion

Pour la classification des opérateurs de fusion, nous nous sommes inspirés des travaux de [5] et [11] où ils présentaient un problème d'agrégation de deux informations, n_1 et n_2 introduites à partir de deux sources différentes pour le même phénomène à observer. Un opérateur binaire F a été choisi pour l'agrégation des données. Cet opérateur est défini par les différents comportements suivants :

- **Sévère** (de type conjonctif) si : $F(n_1, n_2) \leq \min(n_1, n_2) \rightarrow$ les sources sont en accord ;
- **Prudent** si : $\min(n_1, n_2) \leq F(n_1, n_2) \leq \max(n_1, n_2)$;
- **Indulgent** (de type disjonctif) si : $F(n_1, n_2) \geq \max(n_1, n_2) \rightarrow$ les sources sont en désaccord.

Les différentes classes d'opérateurs sont présentées ci-après selon leur variabilité et leur dépendance au contexte [11]:

3.3.1. Opérateurs à comportement constant et indépendant du contexte

Notée **CCIC**, c'est le type d'opérateur qui se ressemblent dans leur manière d'agrégation de n_1 et n_2 (quelque soient les valeurs de n_1 et n_2). Mais le résultat de la fusion est indépendant du contexte de l'agrégation, d'où F est exclusivement sévère, indulgent ou prudent.

3.3.2. Opérateurs à comportement variable et indépendant du contexte

Notée **CVIC**, dans cette classe, le résultat de la fusion est en fonction des valeurs de n_1 et n_2 , or, les opérateurs ne dépendent pas du contexte de l'agrégation.

3.3.3. Opérateurs dépendants du contexte

Notée **CDC**, l'opérateur F renvoie une valeur qui dépend de n_1 et n_2 , et aussi d'une connaissance *a priori* sur le système de capteurs des deux sources ou sur le phénomène lui-même. F peut être construit avec un comportement sévère par une fonction croissante de l'accord entre les deux opérateurs (respectivement : *indulgent - fonction décroissante*).

3.3.4. Quelques propriétés

Chaque type d'opérateur de fusion doit être associatif et commutatif, pour qu'il soit indépendant de l'ordre de présentation des informations. De même, il doit être continu, afin d'assurer la robustesse de la combinaison pour des couples voisins, et enfin, strictement croissant par rapport au couple (n_1, n_2) .

3.4. Fusion probabiliste

Dans [51], nous avons trouvé deux formulations de fusion probabiliste : fusion dite *Non Bayésienne*, où la seule fonction de fusion possible est **la moyenne pondérée des données consommées** avec une somme de poids, positifs ou nuls, égale à 1. L'objectif de fusion, et de chercher la probabilité qu'un objet observé appartienne à une classe précise ; le deuxième type de fusion est celui de la fusion *Bayésienne*, qui va être détaillée juste après.

3.4.1. Combinaison de la fusion probabiliste Bayésienne

Supposons que nous avons un ensemble d'évènements A de cardinalité c , à partir de l'évènement A_i nous utilisons deux capteurs afin d'extraire respectivement deux observations n_1 et n_2 . La probabilité d'obtention de l'évènement A_i en ayant effectivement mesuré n_1 et n_2 est donnée par la règle de Bayes comme suit :

$$P(A_i/n_1, n_2) = \frac{P(n_1, n_2/A_i) \cdot P(A_i)}{\sum_{k=1}^c P(n_1, n_2/A_k) \cdot P(A_k)}$$

- $P(n_1, n_2/A_i)$: c'est la probabilité que n_1 est observée par le premier capteur et n_2 observée par le second, sachant la vraie valeur de A_i .

Si les mesures n_1 et n_2 sont issues des variables aléatoires indépendantes, la fonction de vraisemblance $P(n_1, n_2/A_i)$ peut être simplifiée en un produit de deux mesures $P(n_1, n_2/A_i) = P(n_1/A_i) \cdot P(n_2/A_i)$, d'où nous aurons la simplification suivante :

$$P(A_i/n_1, n_2) = \frac{P(n_1/A_i) \cdot P(n_2/A_i) \cdot P(A_i)}{\sum_{k=1}^c P(n_1/A_k) \cdot P(n_2/A_k) \cdot P(A_k)}$$

Cette dernière équation définit un opérateur conjonctif et *CCIC* de fusion probabiliste, pour les observations fournies par les deux capteurs. Cet opérateur est de plus associatif, commutatif et continu si les probabilités composant la règle le sont [5].

3.4.2. Règle de décision

Le critère de décision le plus utilisé est celui du maximum *a posteriori* (MAP):

$$P(A_i) = \max(P(A_k), 1 \leq k \leq c)$$

D'autres critères de décision Bayésienne possibles existent (*maximum de vraisemblance, maximum d'entropie, l'espérance mathématique, etc.*).

3.5. Fusion des fonctions de croyances

Le nom *espace de discernement* a été introduit par Shafer, afin de décrire l'ensemble des réalisations possibles d'une épreuve (*univers probabiliste ou cadre de décision*), cette espace regroupe un ensemble d'hypothèse distinctes [20].

Dans ce modèle de fusion, nous considérons un espace de discernement $D = \{d_1, \dots, d_n\}$, où chaque hypothèse d_j est observée par deux capteurs différents. Un capteur $i \in \{1, 2\}$ est associé à un jeu de masse m_i exprimant au mieux l'information apportée par chaque source [5]. De même, la masse m_i peut être assignée à des combinaisons des hypothèses (exemple : $d_j \cup d_k$) dans le cas où la distinction entre ces hypothèses ne peut pas être faite par le capteur i .

3.5.1. Combinaison de la fusion d'évidence

La fusion des informations fournies par les deux capteurs est modélisé par une estimation de la masse m à partir de la combinaison des deux masses m_1 et m_2 . Elle est définie par la règle orthogonale de Dempster [5] :

$$(\forall A \subset D, A \neq \emptyset) m(A) = (m_1 \oplus m_2)(A) = \frac{\sum_{d_1 \cap d_2 = A} m_1(d_1) m_2(d_2)}{K}$$

$$m(\emptyset) = 0, \quad K = 1 - \sum_{d_1 \cap d_2 = \emptyset} m_1(d_1) m_2(d_2)$$

Où :

- K représente une mesure d'accord entre les deux capteurs. Plus cette mesure tend vers 0 plus le conflit entre les sources deviens important.

La masse $m(A)$ telle qu'elle est définie ci-dessus engendre l'utilisation d'un opérateur de fusion *CCIC* avec un comportement conjonctive. Autrement, le comportement disjonctif peut être engendré par d'autres règles, en voici l'une de ces règles:

$$(\forall A \subset D, A \neq \emptyset) m(A) = (m_1 \oplus m_2)(A) = \sum_{d_1 \cap d_2 = A} m_1(d_1) m_2(d_2)$$

3.5.2. Règle de décision

Nous pouvons envisager plusieurs règles de décision pour le choix d'une hypothèse, selon l'un des critères de favorisation cités dans [5]. Nous allons choisir l'hypothèse ayant, soit un :

- Maximum de crédibilité ;
- Maximum de plausibilité ;
- Maximum de probabilité pignistique ;
- Intervalle de confiance de longueur minimal.

3.6. Fusion possibiliste

Dans le cas où les informations fournies par les capteurs des deux sources sont incertaines, une fusion possibiliste sera mieux adaptée, surtout en présence de conflit. Cependant, une distribution de possibilité π_i est associée à chaque source $i \in \{1,2\}$ pour qu'elle exprime son incertitude. Plusieurs règles de fusion possibilistes existent dans la littérature, en exploitant des données externes (*mesure de fiabilités des sources*) ou en combinant les conjonctions et les disjonctions. La majorité de ces règles ont pour but de construire une distribution finale à partir de l'ensemble des distributions d'origine [19].

3.6.1. Combinaison de la fusion possibiliste

Le modèle de fusion possibiliste peut s'adapter avec plusieurs opérateurs de différents comportements. Les opérateurs F_i possibles peuvent avoir un comportement conjonctif ou disjonctif, le classement de ces opérateurs est donné par la relation d'ordre suivante [5] :

$$(F_1 \leq F_2) \Leftrightarrow ((\forall n_1, n_2) F_1(n_1, n_2) \leq F_2(n_1, n_2))$$

Voici une notification des techniques de comportements des opérateurs de fusion possibiliste par classe :

- Dans la classe *CCIC* : les opérateurs peuvent adopter plusieurs comportements : exclusivement sévères (*T*-normes, généralisant les intersections aux ensembles flous), prudents (moyennes, agissant comme un compromis entre les capteurs), ou indulgents (*T*-conormes, généralisant les unions aux ensembles flous).
- Dans la classe *CVIC* : selon les valeurs de mesures des deux capteurs, il est possible d'élaborer des opérateurs décrivant l'ensemble des comportements, du plus conjonctif au plus disjonctif.

- Dans la classe *CDC* : le comportement des opérateurs va dépendre d'une information contextuelle choisie (*exemple : conflit entre les sources, confiance accordée aux capteurs, contexte spatial, etc.*).

En résumé, l'opérateur F doit en effet créer une distribution de possibilité globale à partir de deux distributions de données [19].

3.6.2. Règle de décision

Une fois la combinaison des distributions possibilistes des deux sources est faite, le processus de fusion définit une étape de décision pour choisir à quelle classe C_k l'élément x sera affecté. Evidemment, ce choix est basé sur la règle de maximisation des degrés d'appartenance [51] :

$$d_i \text{ si } \mu_i(x) = \max\{\mu_k(x), 1 \leq k \leq n\}$$

La maximisation du degré d'appartenance revient à maximiser le degré de possibilité ou le degré de nécessité (*choisir le critère le plus sévère*).

4. Conclusion

Les méthodes de fusion de données sont largement étudiées afin de résoudre les difficultés liées à la manipulation de données multimodales, imprécises et incertaines. C'est une problématique qui se manifeste, généralement, dans les travaux de recherche basés essentiellement sur les méthodes de : fouille de données, extraction de connaissances, classification dynamique, analyse automatique pour prise de décision.

Dans ce chapitre, nous avons essayé de structurer une large présentation des différents concepts de la fusion de données. Nous pouvons encadrer le processus de fusion sur trois modèles : le modèle probabiliste, modèle possibiliste et enfin le modèle des fonctions de croyances. Pour chaque modèle nous avons présenté sa théorie de base, à partir de laquelle un ensemble de règles et de méthodes de fusion ainsi que des critères de décision ont été définies.

Cette étude nous a permis de bien exploiter les techniques et les capacités des méthodes de fusion de données existantes dans la pratique. Elle nous sera très utile par la suite pour faire part dans la proposition d'une approche de classification dynamique, précisément pour le traitement de la *multimodalité* des données à classifiées.

CHAPITRE 4

LES SYSTEMES MULTI-AGENTS

1. Introduction

En pratique, un travail collaboratif (*travail en groupe*) est plus favorable pour réussir et d'avoir des meilleurs résultats qu'un travail individuel, où les compétences de chaque individu seront exploitées d'une façon à optimiser le temps d'exécution des tâches d'une part, et d'aboutir à un résultat final plus précis et fiable de l'autre part. Dans la nature, le phénomène de coopération est observé en particulier chez les insectes (*fourmis, abeilles*) avec leurs méthodes de déplacement en juxtaposé, ce qui rend ces simples actions individuelles en un comportement global et intelligent. En informatique, la coopération entre plusieurs unités algorithmiques séparées peut être modélisée ou plutôt appliquée selon plusieurs méthodes, parmi lesquelles nous avons les systèmes multi-agents, dont la notion de coopération est définie par la collaboration de plusieurs agents très simples et autonomes, dans le but de résoudre un problème complexe impossible de la résoudre individuellement (*par un seul agent*).

Suite à la rencontre entre l'Intelligence Artificielle (*IA*) et les Systèmes Distribués, les systèmes multi-agents ont vu le jour (*dans les années 80*) [54]. L'objectif été de tirer avantage des deux disciplines (*IA & SD*), et simplifier le développement et l'implémentation ainsi le contrôle d'exécution des systèmes complexes. Le principe de l'intelligence artificielle classique est défini par un développement des programmes autonomes pour l'exécution des tâches complexes à partir des connaissances des expertes centralisées dans un système unique. Contrairement à un système centralisé, un système distribué repose sur l'idée d'avoir un ensemble d'unités de calculs indépendantes qui apparait à l'utilisateur comme un système unique et cohérent [61].

Donc un système multi-agents est classé au centre d'un rayon d'étude approprié à l'intelligence artificielle distribuée (*IAD*), où le critère d'intelligence n'est pas restreint sur une seule entité, il est plutôt répartie et distribuée à l'intérieur d'un groupe d'entités nommées *Agents*. Chaque agent est caractérisé par une autonomie de planification et d'exécution des actions (*prise de décision*), tout en communiquant avec les autres agents dans un environnement de collaboration bien défini [24].

L'application des systèmes multi-agents a connue une grande expansion, touchant une large sélection des domaines techniques liés à l'Intelligence artificielle, système distribués et le génie logiciel, parmi ces domaines nous avons :

- **L'industrie** : l'automatisation des processus de production & la robotique ;
- **L'information** : systèmes d'aide à la décision, recherche d'information, l'assistance personnelle, classification;
- **Enseignement assisté par ordinateur** : gestion des cours, évaluation ;
- **Médecine** : Interprétation des images médicales & la télémédecine ;
- **Gestion des réseaux** : énergétique, transport (*routier, aérien et ferroviaire*) ;
- **Jeux de défis** : entre machine et homme ;
- etc.

Dans ce quatrième chapitre, notre objectif n'est pas d'établir un état de l'art complet et rigoureux sur les notions des systèmes multi-agents, mais nous allons présenter des définitions et des concepts clés relatifs à ce domaine, qui nous paraît les plus appropriés pour la mise en place de notre système de classification dynamique floue proposé.

2. Concept d'Agent

Le principe d'agent peut être défini par plusieurs caractérisations selon le domaine de son application. Dans notre cas, l'agent est conçu pour être utilisé dans une modélisation d'Intelligence Artificielle Distribuée, où il représente l'élément de base d'un système multi-agents. Parmi les définitions les plus abordées d'un agent nous avons celle de [24] : « *Un agent est défini comme une entité physique ou virtuelle, autonome, capable d'agir dans un environnement, de communiquer avec d'autres agents, de percevoir son environnement et de se reproduire (éventuellement)* ».

L'auteur définit un agent par une entité physique (*exemple : Robot*) ou virtuelle (*exemple : programme informatique*) possédant un comportement individuel, et caractérisée par les propriétés fondamentales suivantes [53]:

- **Autonomie** : un agent est capable de prendre des décisions et d'exécuter des actions d'une façon autonome par rapport à son état et son environnement, sans intervention externe ;
- **Proactivité** : cette propriété décrit l'initiative individuelle d'un agent pour fixer sa propre mission afin d'atteindre ses propres buts ;
- **Sensibilité** : un agent peut distinguer et communiquer avec d'autres agents par un échange d'informations, comme il peut percevoir l'environnement qu'il partage avec ces agents ;
- **Flexibilité** : l'interaction d'un agent est automatique dès qu'un changement brusque est aperçu dans son environnement ;
- **Localité** : l'ensemble d'actions et des interactions d'un agent sont limités ;
- **Adaptation** : Habilité d'un agent de régler ses compétences et connaissances en fonction des agents avec lesquels il interagit et/ou de l'environnement dans lequel il évolue.

De plus, une autre propriété clés caractérisant un agent est citée dans [27], elle concerne le comportement d'un agent dans son environnement. C'est *l'Aptitude sociale*, qui indique la nature d'interaction de l'agent avec les autres agents, soit coopérative ou compétitive pour atteindre un but commun.

En résumé, nous pouvons reprendre la définition d'un agent par une combinaison entre l'ensemble des comportements, réactions et perceptions lui permettant d'interagir avec son environnement, et l'ensemble des ressources individuelles (*compétences, connaissances, mécanismes de gestion de tâches..*) lui permettant de collaborer et atteindre un objectif commun.

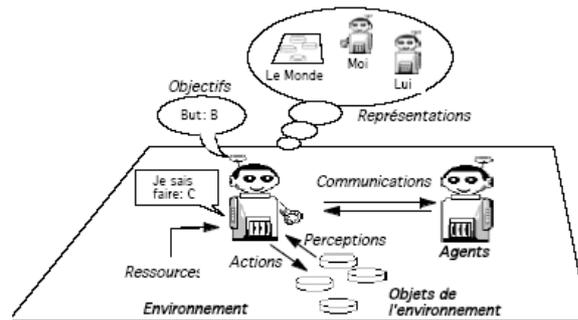


Figure 4.1 : Représentation imagée d'un agent en interaction avec son environnement et les autres agents [24].

2.1. Types d'Agents

Nous pouvons classer les agents en trois classes : Agents réactifs, Agents cognitifs et Agents hybrides. En effet, cette classification d'agents est réalisée suivant leur niveau *d'intelligence*. Par la suite nous présentons brièvement chacune de ces trois types, avec plus de détaille pour les agents cognitifs qui seront utilisés dans le système proposé.

2.1.1. Agents réactifs

Un agent est *réactif* s'il a une capacité de communication réduite, utilisant une simple modélisation de lui-même et des autres agents, ainsi une représentation modeste (*non explicite*) de son environnement. Ce type d'agents est connu par son comportement stimulus/réponse [37], son exécution est liée directement à ses perceptions par une fonction réflexe : stimulation / réponse. Les agents réactifs sont caractérisés par leur rapidité d'interaction pour des problèmes simples, n'exigeant pas un grand niveau de raisonnement, ils réagissent dès qu'ils perçoivent un changement de leur environnement ou ils reçoivent des messages provenant des autres agents.

Ils n'ont pas besoin ni de formuler leurs objectifs, ni de fixer des processus de planification [27], leurs décisions sont fondées sur un nombre restreints d'informations avec des simples règles de décision de type *situation* \rightarrow *action*.

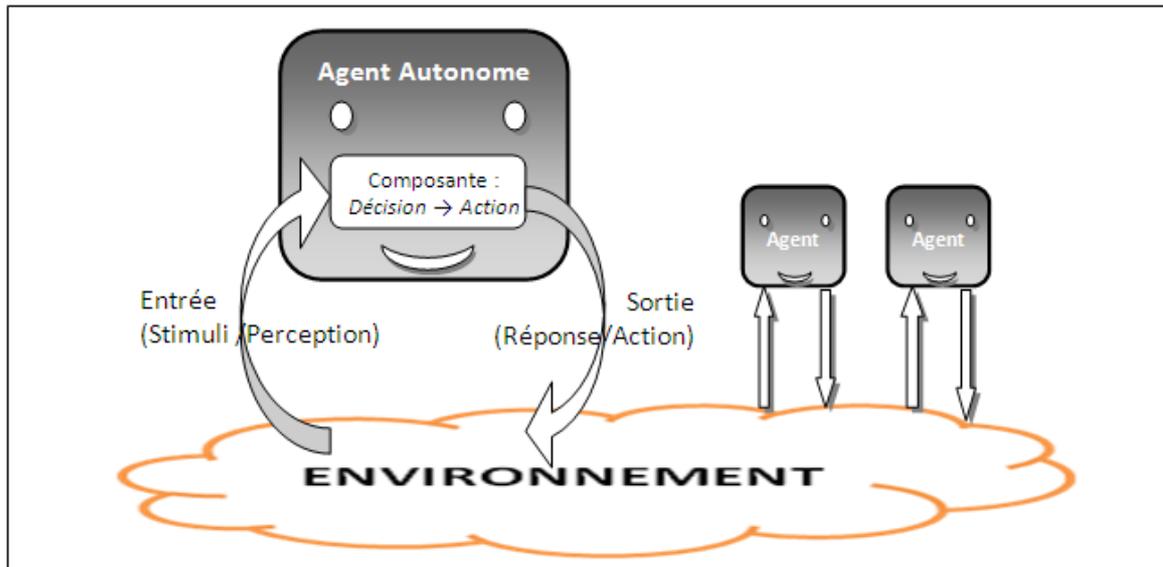


Figure 4.2 : Interaction d'un Agent réactif avec son environnement et les autres agents.

2.1.2. Agents cognitifs

Un agent cognitif est plus performant qu'un agent réactif, où il possède une grande capacité de mémoire, de raisonnement et de communication. Il est capable d'élaborer des modèles des autres agents et le modèle de son environnement pour mieux comprendre et prévoir les faits et les événements qui peuvent arriver ainsi l'évolution des phénomènes. Eventuellement, ce type d'agent peut spécifier les cognitions interactionnelles d'entrée possibles avant qu'elles soient perçues [24].

Un système d'agents cognitifs est caractérisé par un nombre réduit d'agent. La disposition d'une base de connaissance explicite est obligatoire pour chaque agent, elle est utilisée afin de lui permettre d'accomplir sa mission et de gérer ses interactions avec les autres agents [25]. Le concept de cognition permet de fixer les objectifs à atteindre, et cela par une planification des actions et des services à fournir, sur la base des connaissances et des informations recueillies au fur et à mesure à partir de l'environnement d'interaction de l'agent.

Dans [27], l'auteur a rapporté une liste de fonctionnalités caractérisant un agent cognitif, que nous allons citer ci-après :

- **Expertise du domaine** : un agent cognitif peut disposer de plusieurs compétences utilisées dans sa prise de décision pour la résolution d'une partie du problème ;
- **Perception** : interprétation des connaissances perçues par l'agent suivant ses compétences, ses objectifs, ses engagements, etc. ;

- **Communication** : en communiquant les agents cognitifs peuvent échanger des informations et combiner leurs actions, comme ils peuvent exprimer leurs intentions ;
- **Représentation des autres et de soi** : lors de son interaction avec les autres agents, un agent peut recueillir l'ensemble de connaissances représentant son état (*je : sais faire quoi ? veux faire quoi ? en train de faire quoi ? , quels sont mes plans ?*), et les états des autres (*qui : peut faire quoi ? veut faire quoi ? fait quoi ? , où se trouve les autres ? , quels sont leurs plans ?*) ;
- **Contrôle** : l'agent peut contrôler ses tâches internes (*engagements & buts*), ainsi ses tâches collaboratives (*relation avec les autres agents*) ;
- **Apprentissage** : cette fonctionnalité survient pour chaque nouvelle perception et action émise par l'agent, où ce dernier met à jour sa base de connaissance ;

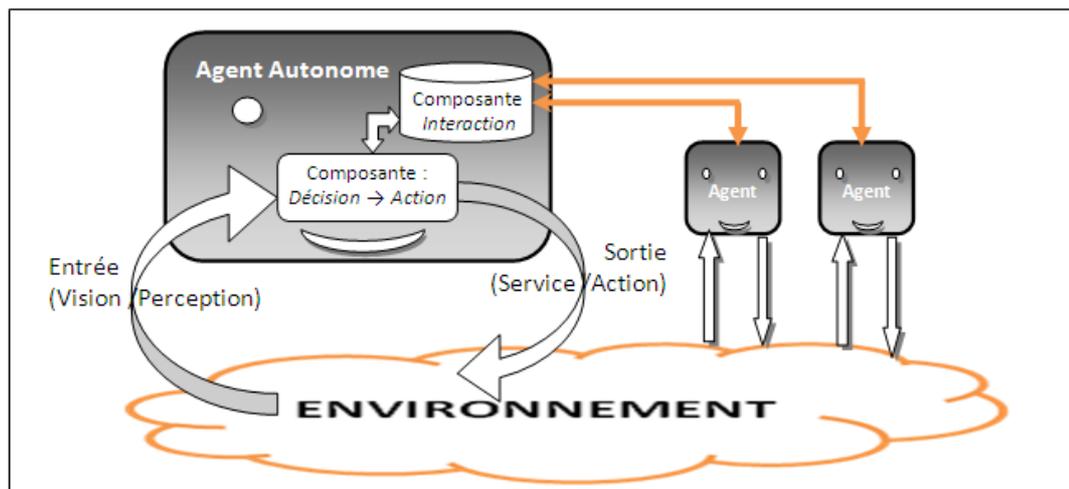


Figure 4.3 : Interaction d'un Agent cognitif avec son environnement et les autres agents.

De même, nous pouvons distinguer différents types d'agents cognitifs, par rapport à leurs degrés d'autonomie, d'adaptation et de coopération qui sont présentés comme suit :

- **Agents intelligents** : définissent les plus grands niveaux d'autonomie (*planification des actions*), coopération (*communication avec les autres agents*) et d'adaptation (*apprentissage de connaissances*) ;
- **Agents collaborant** : caractérisés par un grand niveau d'autonomie et de coopération sans adaptation. Ils peuvent modéliser les comportements d'organisations humaines ou animales ;

- **Agents d'interface (Assistants)** : ils sont fortement adaptatifs, et utilisés pour l'assistance des utilisateurs dans le changement et paramétrage des interfaces manipulées. Ils facilitent l'apprentissage humain selon ses actions capturées par un dispositif : souris, clavier, voix, touches tactiles, etc. ;
- **Agents d'information** : ils sont fortement autonomes dans leurs réactions, où ils peuvent déclencher individuellement des actions lors d'un manque ou apparition de nouvelles informations. Ils sont capables d'agir à des heures fixes (*un calendrier des tâches*) pour la récolte d'informations.

Afin de mieux synthétiser les divergences entre les agents réactifs et les agents cognitifs, une comparaison tabulaire est ainsi présentée contenant une liste des propriétés les plus fondamentales pour définir le concept d'agent.

Caractéristiques	Agents cognitifs	Agents réactifs
Représentation de l'environnement	Explicite	Non explicite
Historique (<i>base d'apprentissage</i>)	Oui	Non
Structure	Complexe	Simple (<i>Stimulus / Réponse</i>)
Granularité	Elevé	Faible
Capacité de raisonnement	Elevé	Faible
Temps de réaction	Lent	Rapide
Nombre d'agents dans le système	Petit	Grand

Tableau 4.1 : Comparaison entre agents cognitifs et agents réactifs.

2.1.3. Agents hybrides

Il est intéressant d'avoir un agent possédant en même temps les deux niveaux de comportements cognitif et réactif, cela va mener vers une combinaison entre l'efficacité et la complexité, ou encore entre l'autonomie et la coopération de l'agent. L'entité résultante est connue sous le nom d'*agent hybride* [50].

Les agents hybrides sont souvent utilisés pour faire face aux faiblesses des agents réactifs et cognitifs. En effet, la réactivité rend le comportement des agents assez simpliste sans intelligence ni de coopération. Or la cognition oblige les agents à raisonner par des mécanismes qui ne sont pas faciles à manipuler et ne sont pas suffisamment réactifs [27]. Dans [50], l'auteur a parfaitement exposé les principales motivations d'application des agents hybrides pour la construction des applications de type « *Temps Réel* ».

L'hybridation peut être définie par plusieurs architectures possibles. Chaque architecture est représentée par une structure de fonctionnalité de l'agent, sous forme de deux ou plusieurs couches hiérarchiques interagissant entre elles. Souvent une architecture hybride est constituée de trois couches qui sont jugées suffisantes. La couche inférieure est considérée purement réactive, elle sert à prendre des décisions en se basant sur des données brutes en provenance des capteurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision qui se situe au niveau des connaissances de l'environnement. La couche supérieure a pour objectif de gérer les aspects sociaux de l'environnement, c'est à dire d'organiser la communication entre l'agent hybride et les autres agents [34].

La représentation de l'approche hybride par une structure en couches présente plusieurs avantages, nous avons :

- Une séparation claire des différentes fonctionnalités de l'agent (*agent modulé*) ;
- Une conception plus compacte d'un agent robuste ;
- Une augmentation des performances de l'agent suite au travail parallèle des couches ;
- Une augmentation de la réactivité de l'agent puisqu'il peut réfléchir tout en surveillant son environnement et en réagissant avec les autres agents ;
- Un allègement de la base de connaissance globale de l'agent par sa division en sous base pour chaque couche.

2.1.4. Architecture BDI

Les chercheurs ont développé l'architecture BDI (*Belief, Desire, Intention*, pour croyance, désir, et intention) sur la base du raisonnement pratique (*la pratique des êtres humains*), en prenant en compte les *états mentaux* de l'agent en question. Cette approche va décrire l'état interne d'un agent par rapport à ses attitudes mentales suivantes [34]:

- **Les croyances** (*Beliefs*): l'ensemble des connaissances recueillies par l'agent pour connaître son environnement ;
- **Les désirs** (*Desires*): l'ensemble des états possibles sur lesquels l'agent peut vouloir s'engager ;
- **Les intentions** (*Intentions*): l'ensemble des états sur lesquels l'agent s'est engagé ou il a engagé des ressources.

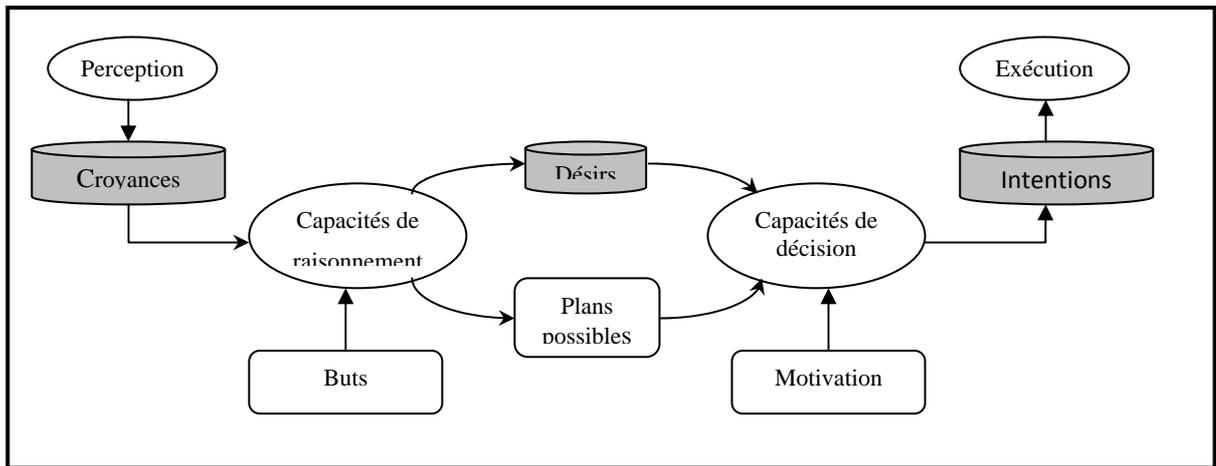


Figure 4.4 : Architecture BDI.

L'agent BDI perçoit les informations et les messages qui l'entourent ou en provenance d'autres agents, afin de mettre à jour ses croyances portant sur son environnement, ainsi : ce que croient les autres agents avec leur actions possibles et leur description. L'agent est muni de buts explicites. C'est ce qui lui garde son autonomie de réaction, puisqu'il n'a pas nécessairement des buts communs avec d'autres agents. Ses capacités de raisonnement lui permettent d'utiliser ses connaissances et ses buts pour déduire des plans d'actions, décrivant des alternatives d'actions que l'agent peut mettre en œuvre pour satisfaire ses buts.

Les désirs représentent les états de l'environnement ou de l'agent lui-même, que celui-ci aimerait avoir atteindre. Elles proviennent du caractère partiel des plans déduits. Les motivations sont l'expression symbolique ou numérique des préférences servant comme critères pour le choix d'un plan possible. Alors, l'habileté de décision de l'agent doit appliquer les critères de performances pour filtrer le plan qui semble le meilleur parmi les plans possibles, ensuite il peut générer ses nouvelles intentions qui sont l'expression du plan choisi.

A la fin, l'agent va exécuter ses intentions sous la forme d'actions, qu'il a décidé de faire pour accomplir ses désirs, dans son environnement et/ou avec les autres agents. L'agent devra suivre ce plan jusqu'au bout, mais il existe certaines limites qui définissent les conditions d'abandon des intentions.

3. Système Multi-Agents

Au cours de notre recherche bibliographique, nous avons trouvé plusieurs définitions des systèmes multi-agents qui s'entendent sur un même principe affirmant : qu'un SMA est un ensemble d'agents autonomes capables de communiquer et d'agir dans un environnement commun pour effectuer plusieurs tâches communes ou individuelles.

Dans [24] : « On appelle système multi agent, un système composé des éléments suivants: un ensemble de tâches à réaliser et un ensemble d'objets associés à l'environnement » ; cette définition veut bien signifier qu'un SMA est un système composé d'un ensemble d'agents, un ensemble d'objets et d'un environnement. L'ensemble de tâches permet aux agents de produire, percevoir, échanger et manipuler les objets de l'environnement.

3.1. Intérêts

Nous pouvons constater diverses importances justifiant l'utilisation des systèmes multi-agents. Ci-après l'énumération des principaux avantages des SMA [34] :

- Adaptation à la modélisation des modèles distribués ;
- Grande précision et une meilleure qualité de résultats pour la résolution des problèmes complexes par une coopération de plusieurs agents ;
- Possibilité de résolution des problèmes complexes et de grande taille par plusieurs agents suite à l'incapacité de les résoudre par un seul agent ;
- Modularité, consiste à scinder la base de connaissance globale en petite taille sur un ensemble d'agents réduisant la complexité de formalisation et de traitement du système de résolution ;
- Efficacité, mesurée par la rapidité de résolution du problème, où les tâches peuvent être exécutées en parallèles par des agents autonomes ;
- Fiabilité, aperçue par la continuité de fonctionnement du système de résolution dans le cas d'un arrêt de fonctionnement d'un agent ;
- Réutilisation, l'expertise d'un agent acquise lors de ses exécutions précédentes peut être réutilisée plusieurs fois sans le besoin de sa ré-implémentation.

3.2. L'organisation

Nous distinguons deux structures sociales possibles pour décrire la manière d'organisation des agents au sein d'un système multi-agents :

- **Systèmes multi-agents à structure horizontale** : tous les agents sont alignés sur le même niveau, aucune supériorité entre les agents n'est envisagée (*maître/esclave*). Ici le contrôle des agents est le résultat des mécanismes de contrôle locaux propres aux agents ;

- **Système multi-agents à structure verticale** : les agents sont positionnés sur plusieurs niveaux, alors chaque niveau va représenter une structure horizontale. Dans la structure verticale, les agents d'un niveau inférieur sont réclamés à résoudre un problème après sa réception par des agents superviseurs, qui leur sont supérieurs dans la hiérarchie et qui peuvent prendre leur contrôle.

3.3. L'environnement

Il représente le monde dans lequel l'ensemble des agents évoluent et partagent des objets et des messages d'informations entre eux. Nous pouvons spécifier deux sortes d'environnement selon [55]: en premier lieu, un environnement global du système multi-agents représentant le volet physique des ressources en agents et leur action sur les objets; en second lieu, nous avons un environnement local d'agent incluant l'environnement global du SMA et l'ensemble des environnements locaux des autres agents.

La présentation d'un environnement peut être décrite selon deux méthodes, soit par un bloc monolithique, dans ce cas nous parlons d'un environnement *centralisé* (*accès de tous les agents à la même structure*). Soit par un ensemble de cellules disposées sous forme d'un réseau formant une sorte d'automate cellulaire, et constituant un environnement dit *distribué* (*accès partiel des agents par cellule ; comme exemple le déplacement des robots sur une surface divisée en carrés*) [24]

3.4. Formes d'interaction

La résolution d'un problème complexe par un système multi-agents est simplifiée par son comportement intelligent représentant le travail collectif des agents. L'interaction est un phénomène important dans un SMA qui permet aux agents de participer à la réalisation d'un but global [27]. Elle est assurée par les deux mécanismes de transfert d'informations : *perception* (entre l'environnement et les agents) et *communication* (entre agents). Eventuellement, nous distinguons trois formes d'interaction entre agents : *coopération*, *coordination* et *communication*.

3.4.1. Coopération

L'engagement d'un ensemble d'agents dans la résolution commune d'un problème commun est un travail de *coopération*. Le niveau des deux paramètres (*compétences & ressources*) pour chaque agent peut déterminer les situations de coopération illustrées dans le tableau suivant [24] :

Compétences	Ressources	Situation de coopération	Exemple de coopération
Suffisantes	Suffisantes	Pas besoin de coopération	-
Suffisantes	Insuffisantes	Encombrement	Utilisation d'un seul instrument par plusieurs agents.
Insuffisantes	Suffisantes	Collaboration simple	Addition de l'ensemble des résultats par étape pour avoir un résultat final (<i>production en chaîne</i>).
Insuffisantes	Insuffisantes	Collaboration coordonnée	Restriction d'exécution de tâches aux agents compétents.

Tableau 4.2 : Situations de coopération des agents.

Dans [38], plusieurs modèles de coopérations sont décrites par l'auteur, pour n'importe quel type d'organisation de la société d'agents dans le système. Ces modèles sont listés dans le tableau suivant :

Modèle	Description
Coopération par commande	un agent superviseur divise un problème en sous-problèmes pour les distribués entre les agents du niveau inférieur.
Coopération par partage : a. Des tâches b. Des résultats	a) Un problème global est décomposé en sous-tâches, ensuite distribuées aux différents agents, les divers résultats obtenus sont intégrés pour former la solution du problème. b) Les données initiales du problème sont affectés aux agents concernés, après une phase de résolution, les résultats partiels sont groupés puis synthétisés.
Coopération par appel d'offre	Diffusion d'une liste des sous-problèmes d'un problème décomposé par un agent, ce dernier choisira par la suite les agents souhaitant avoir une offre de ces sous-problèmes.
Coopération par compétition	L'agent ayant décomposé et diffusé la liste des sous-problèmes d'un problème, va recevoir les résultats de résolution en provenance des autres agents afin de les triés.

Tableau 4.3 : Modèles de coopération des agents.

4.1.1. Modes de contrôle

La coopération sociale entre agents devra être contrôlée pour mieux gérer et coordonner l'autonomie de ces entités, et bénéficier d'une utilisation rationnelle des moyens de communications et des ressources disponibles. Le plan de contrôle est établi selon les deux approches suivantes [27]:

- **Contrôle Centralisé** : un agent superviseur va recueillir les connaissances globales sur le problème à résoudre, dans le but d'assurer le contrôle et l'orientation des processus partiels de décision des autres différents agents ;
- **Contrôle Distribué** : dans ce type de système nous avons des agents dotés d'une grande autonomie de raisonnement, possédant des méta-connaissances sur les différents agents. Le contrôle est décentralisé, chaque agent doit échanger des informations avec les autres agents pour connaître leurs plans et leurs buts.

3.4.2. Coordination

La coordination est une planification des interdépendances des tâches exécutées par les différents agents coopérant pour la résolution d'un problème commun sans conflits entre eux. La construction d'un plan de coordination par un agent va dépendre des plans des autres agents qui : sont créés, entraînent d'être créés ou vont être créés. Les techniques principales de coordination dans les systèmes multi-agents sont citées ci-dessous :

- **Coordination par allocation des tâches** : consiste à décomposer une tâche initiale en une liste de sous-tâches ensuite la partager entre plusieurs agents. L'allocation peut être centralisée ou distribuée ;
- **Coordination basée sur des structures organisationnelles** : l'objectif est de créer une structure organisationnelle contenant un espace d'interaction avec des règles, auxquelles les agents doivent respecter pour limiter leurs conflits et coordonner leur travail ;
- **Coordination basée sur des protocoles d'interaction** : ces protocoles définissent un schéma prédéfini structurant les conversations entre les agents ;
- **Coordination par planification** : consiste à coordonner les plans de plusieurs agents ;
- **La négociation** : c'est un mécanisme très important pour la coordination des agents et la résolution de leurs conflits. Elle intervient pour mettre en place un compromis entre agents lorsqu'il y a un conflit de solutions différentes, pour le même problème, ou d'accès aux ressources.

3.4.3. Communication

Si nous avons des agents coopérant, coordonnant leurs actions et résolvant des problèmes en commun, c'est grâce à leur possibilité de communication. En communiquant, les agents peuvent synchroniser leurs actions et résoudre leurs conflits par négociation. Deux stratégies de communications sont utilisées :

- **Par partage d'informations (Tableau noir) :** ici les agents ne se communiquent pas directement entre eux, mais plutôt à travers une zone de données commune connue par le *Tableau noir* ou *Blackboard*. Dans ce tableau un agent peut déposer des messages, ses plans et des résultats partiels de ses calculs, comme il peut retirer ces mêmes types d'informations déposées par les autres agents ;
- **Par envoi de messages :** dans ce cas, chaque agent peut communiquer directement avec les autres agents. La communication est soit individuelle (*destinée vers un seul agent*) soit regroupée (*diffusée entre les agents*). La structuration des messages envoyés est basée souvent sur les langages suivants: KQML [63] (*Knowledge and Query Manipulation Language*), FIPA ACL [14] (*Agent Communication Language*). Ces deux langages sont plus détaillés dans [27].

4. Le développement des systèmes multi-agents

Cette section est composée de deux parties. Au commencement, nous allons exposer les différentes méthodes de conception des SMA les plus abordées dans la littérature. A la fin, il y aura un enchaînement sur la présentation des plates formes disponibles en pratique recueillant ce genre des systèmes collaboratifs intelligents.

4.1. Méthodologies de conception des systèmes multi-agents

Le rôle d'une méthodologie est la définition d'une séquence ordonnée d'étapes pour la conception et l'implémentation d'une application, avec la possibilité de gérer son cycle de vie global. Les méthodologies orientées agents spécifient les étapes à suivre pour développer un système multi-agents, allant de la phase d'analyse à la phase de conception ainsi que le formalisme utilisé. Ils sont classés en deux catégories : en premier lieu nous avons des extensions des méthodologies orientées objet (AAII [36], GAIA [62], AUML [49] et MAS-ML [21]) ; en second lieu nous avons des méthodologies s'inspirant des systèmes à base de connaissances (CoMoMAS [28] et MAS-CommonKADS [15]).

4.1.1. La méthodologie AAI (Australian Artificial Intelligence Institute)

Développée au sein de l'institut Australien de l'Intelligence Artificielle, l'AAI est basée sur l'utilisation de la technologie BDI-PRS (*Belief-Desire-Intention Architecture & Procedural Reasoning System*). Avec l'architecture BDI chaque classe d'agent possède trois attributs : croyances, désirs et intentions. Cette méthodologie définit deux types de modèle : un *modèle externe* spécifiant le point de vue système, entre autre l'ensemble des agents, les relations entre agents, les classes d'agents et les instances associées, les rapports d'héritage entre les classes d'agents et les instanciations de ces classes pendant l'exécution ; un *modèle interne* représentant l'implémentation des agents.

La conception d'un système multi-agents avec la méthodologie AAI doit passer par les étapes suivantes :

- Identification des rôles importants de l'application ;
- Développement d'une hiérarchie de classes d'agents ;
- Pour chaque rôle :
 - Identification des responsabilités, des services exigés et fournis par le rôle ainsi les buts associés à chaque service ;
 - Détermination des plans utilisables avec leurs conditions de favorisation;
- Déterminer la structure de croyances du système (*information exigée pour chaque couple « plan/but »*).

4.1.2. La méthodologie GAIA [62]

Proposée par Michel Wooldridge, la méthodologie GAIA permet de passer automatiquement des énoncés et demandes du problème (*un cahier de charge*) à la conception assez détaillée et implémentation du système. La phase initiale de cette méthodologie est une analyse des entités *abstraites* (*rôles, permissions, responsabilités et protocoles*) qui sont utilisées par la suite pour la conceptualisation du système avant l'implémentation. La phase finale est une conception des entités *concrètes* (*types d'agents, services et connaissances*) modélisant le système à implémenté.

L'entité agent ou rôle possède quatre attributs : responsabilités, permissions, activités et protocoles. Où les responsabilités déterminent la fonctionnalité de l'entité, les permissions indiquent les droits de l'entité, et enfin les activités sont les actions réalisées par l'entité sans sa coopération avec les autres agents.

4.1.3. La méthodologie Cassiopeia [Drogoul et al, 1998a]

La Cassiopeia a été appliquée pour la conception d'un groupe de « Robots Footballeurs » [23]. Dans cette méthodologie nous distinguons trois phases pour la conception d'un système multi-agents, partant sur le principe d'une spécification des comportements nécessaires des agents pour l'exécution de leurs tâches. Ces trois étapes sont les suivantes :

- Identification des comportements élémentaires des agents ;
- Identification des comportements relationnels entre les agents;
- Spécification de la dynamique d'organisation des agents (*qui est allié avec qui ?*).

4.1.4. Les méthodologies AUML & MAS-ML

Le langage de modélisation unifiée UML est un essai d'unification des différents paradigmes d'analyse et de conception pour la modélisation des systèmes orientés objet. La notation UML désigne un langage et non pas une méthodologie, or son extension vers une modélisation orientée agent introduit les deux nouvelles notations suivantes : AUML (*Agent Unified Modelling Language*) ; et MAS-ML (*Multi-Agent System Modeling Language*).

[21]. **MAS-ML** définit un langage de modélisation orientée agent. Cette méthodologie définit de nouvelles méta-classes UML par des méta-modèles. Elle propose deux nouveaux diagrammes : *d'organisation* et de *rôle*, qui sont des résultats de l'extension des diagrammes de classes et de séquence.

[49]. **AUML** permet la représentation des protocoles d'interactions entre les agents et les rôles attribués pour chaque agent, par deux nouveaux diagrammes : *protocoles* et *rôles*. Cette méthodologie a facilité la conception des systèmes multi-agents pour des utilisateurs déjà familiarisés avec le développement orienté objet, même ils pourront distinguer entre un système d'une société d'agents et une collection d'objets.

4.1.5. La méthodologie MAS-CommonKADS [15]

La méthodologie MAS-CommonKADS constitue une extension de la méthodologie de l'ingénierie des connaissances CommonKADS, dans laquelle est rajoutées des techniques orientées objets et des ingénieries de protocole, afin de pouvoir modéliser les agents et leur interactions. Cette méthodologie consiste à développer les sept modèles suivants :

- *Modèle d'Agents* : définit les caractéristiques importantes de chaque agent ;
- *Modèle de Travail* : décrit les tâches devant être effectuées par les agents ;
- *Modèle d'Expertise* : définit les connaissances utiles aux agents pour atteindre leurs buts;

- *Modèle d'Organisation* : décrit les relations structurelles (*hiérarchie*) entre agents ;
- *Modèle de Coordination* : définit les interactions (*relations dynamiques*) entre agents ;
- *Modèle de Communication* : décrit les interactions entre l'agent humain et l'agent logiciel ;
- *Modèle de Conception* : définit l'architecture globale du système après le regroupement de tous les modèles précédents.

4.2. Les plates-formes

Dans cette partie, nous analysons succinctement les différentes plates-formes des systèmes multi-agents les plus courantes dans la pratique. Une plate-forme multi-agent est un outil d'assistance au développement d'un SMA, elle permet de construire des agents alliés en société. Notre étude concernera les plates-formes suivantes : *Agent Builder*, *Madkit*, *Jade* et *Zeus*. Ces plates-formes étant plus détaillées dans [54].

4.2.1. AgentBuilder

Cette plate-forme est une suite d'outils permettant de construire des agents intelligents. Elle a été développée par *Rectular System Inc*. La création du comportement des agents est basée sur l'architecture BDI, et le langage de communication utilisé est celui de KQML. La méthodologie globale de l'*AgentBuilder* est définie par les étapes suivantes [54]:

- **Analyse** : spécifier les objets du domaine et leurs opérations sous forme d'une ontologie du domaine ;
- **Conception** : décomposer le problème en fonctionnalités d'agents. Ces derniers sont ensuite identifiés avec leurs rôles et leurs caractéristiques ainsi leurs protocoles d'interactions ;
- **Développement** : à travers un ensemble d'outils graphiques l'utilisateur est assisté pour définir les comportements des agents, par des règles de croyances initiales, intentions et capacités ;
- **Déploiement** : L'exécution du système est lancée par un *Run-Time Agent Engine*, qui interprète le code des agents ensuite il les exécute. De même, un outil de débogage est fourni pour le suivi des états mentaux et interactions des agents.

AgentBuilder possède une documentation de support très riche, son inconvénient est qu'il ne nous permet que de construire des agents basés sur l'architecture *BDI*.

4.2.2. Madkit

Madkit (*Multi Agent Development Kit*) représente une plate-forme multi-agents développée en Java par Olivier Getknechi et Michel Ferber, selon le modèle organisationnel Aalaadin (*Agent, Groupe, Rôle*). Dans cette plate-forme, chaque agent a un rôle et peut appartenir à un groupe. Madkit n'a pas une étape d'analyse particulière, elle passe directement par les étapes suivantes :

- **Conception** : conformément au modèle Aalaadin, une définition du modèle organisationnel (*groupes, rôles*), et du modèle d'interaction (*protocoles, messages*), ainsi des entités spécifiques (*tâches, buts, etc.*) est faite ;
- **Développement** : implémentation d'un modèle d'agent choisi, or Madkit ne possède pas des modèles d'agent spécifiques, une sollicitation des librairies est très envisageable ;
- **Déploiement** : à travers une interface graphique l'utilisateur peut créer, modifier et détruire des agents dans la G-Box.

Madkit est un logiciel libre, très utile pour un usage de pédagogie ou de recherche. Son principal point faible est le manque de modèle d'agent. Néanmoins, ils existent de nouvelles librairies sur le net contenant des agents cognitifs intégrables dans cette plate-forme.

4.2.3. Jade (<http://jade.tilab.com/>)

La plate-forme Jade (*Java Agent DEvelopment Framework*) est développée avec le langage Java conformément aux spécifications FIPA au sein du laboratoire TILAB. A travers un ensemble d'outils graphiques, Jade assiste les développeurs dans leurs phases de déploiement et de débogage. C'est une plate-forme qui peut être distribuée sur réseau (*installée sur un ensemble de machines avec des utilisateurs distants*).

L'association entre Jade et les bibliothèques LEAP (*Lightweight Extensible Agent Platform*) a permis de modéliser un agent en un processus autonome et indépendant communiquant avec les autres agents par envoi de messages structurés en FIPA-ACL. Jade est un logiciel libre (*open source*) distribué par Telecom ITALIA, titulaire du droit d'auteur. Cette plate-forme est composée des trois modules suivants :

- **Director Facilitator** : fourniture d'un annuaire de services à la plate-forme ;
- **Agent Communication Channel** : supervision des communications entre les agents ;
- **Agent Management System** : gestion des authentifications et accès des agents du système.

4.2.4. Zeus

Développé en Java par British Telecom conformément aux spécifications FIPA, Zeus est conçu pour le développement rapide des systèmes à base d'agent collaboratifs. Elle regroupe les composantes suivantes :

- Une boîte aux lettres avec gestionnaire de messages pour analyser et envoyer les messages à leurs destinataires (*Agents*) ;
- Un moteur de coordination ;
- Un planificateur des tâches d'agent en fonction des décisions du moteur de coordination, des ressources disponibles et des spécifications des tâches ;
- Un ensemble de bases de données représentant les plans d'agent, les ressources et l'ontologie utilisée ;
- Un contrôleur d'exécution qui gère l'horloge locale de l'agent et les tâches actives.

La méthodologie de Zeus est décomposée en quatre étapes :

- **Analyse** : consiste à modéliser les rôles. Elle se fait à l'aide des diagrammes de classe UML et des patrons, ce qui évite l'introduction de nouveaux formalismes et la rend accessible à une plus grande audience ;
- **Conception** : définit des agents collaboratifs qui remplissent les rôles définis dans la phase précédente afin de satisfaire des buts. Cette étape est basée sur trois cas études : *FruitMarket* (vente), *PC Manufacture* (chaîne de distribution) et *Maze Navigator* (agents à base de règles) ;
- **Développement** : à partir des trois cas d'études de conception et l'*Application RealisationGuide* le développement des agents est fait. Cinq activités sont impliquées : création d'ontologie, création d'agents, configuration de l'agent utilitaire, configuration de l'agent tâche et implémentation des agents ;
- **Déploiement** : pour accomplir cette étape la plate-forme offre une documentation riche avec un *RuntimeGuide*, qui simplifie le déploiement et l'exécution du système via une interface graphique conviviale. Un outil de débogage et de visualisation sont disponibles, permettant le suivi des états de changement de la société d'agents pendant l'exécution du SMA.

Zeus peut être utilisé gratuitement pour des fins académiques ou industrielles. L'inconvénient de cette plate-forme est qu'elle supporte qu'un seul modèle d'agent, limitant ainsi son utilisation pour une large gamme d'SMA.

6. Conclusion

D'un point de vue architecture collaborative, les systèmes multi-agents offrent un cadre intéressant pour résoudre notre problématique et élaborer un système de classification dynamique basé sur la coopération de plusieurs agents classificateurs. La mise en œuvre d'un système multi-agents pour la classification dynamique floue de données multimodales et évolutives, sera principalement basée sur la répartition des tâches entre ces agents classificateurs ainsi que la définition de leurs comportements et interactions entre eux, afin d'aboutir à une affectation optimale des éléments à classifiés aux classes. Pour cela, nous devons définir le modèle des agents classificateurs, leurs plans d'action ainsi que leurs protocoles de communications. A cet effet, tous ces modèles seront générés dans le but d'aboutir à un objectif commun (*assurer une meilleure classification de données multimodales non-stationnaires dans des classes dynamiques floues à travers la collaboration des agents classificateurs*).

Au cours de ce présent chapitre, nous avons présenté les principales notions d'agent intelligent et des systèmes multi-agents. Passant d'abord par une définition du concept agent ainsi qu'une typologie des agents, pour décrire ensuite les diverses formes d'interaction qu'un SMA peut prendre : coopération, coordination et communication. En finalité, nous avons listé les méthodologies et plates-formes les plus abordées dans la littérature, utilisées pour la conception et l'implémentation des systèmes multi-agents.

Dans le chapitre suivant, nous allons décrire une approche multi-agents pour la classification dynamique floue manipulant des données non-stationnaires et multimodales. Cette approche sera basée sur des **agents cognitifs**, afin d'avoir une meilleure décision sur : la *présentation des données échantillonnées* ; et *l'interprétation des résultats* obtenus par ces agents classificateurs.

CHAPITRE 5

SYSTEME MULTI-AGENTS POUR LA CLASSIFICATION DYNAMIQUE FLOUE DE DONNEES MULTIMODALES ET EVOLUTIVES

1. Introduction

Au cours des quatre premiers chapitres nous avons pu réaliser une étude d'état de l'art très objective visant les domaines : de classification dynamique, fusion de donnée et systèmes multi-agents. Elle nous a permis d'identifier les principales méthodes et algorithmes de classifications dynamiques floues présentes dans la littérature, ainsi les principaux niveaux de conception d'un système multi-agents.

A travers ce travail de recherche nous souhaitons adapter ces méthodes de classifications avec des données évolutives et multimodales. L'approche que nous proposons dans ce chapitre s'applique bien à la classification de données stationnaires et non-stationnaires qui varient d'une façon aléatoire (*évolutives*), et provenant de sources homogènes ou hétérogènes (*multimodales*).

Cette approche est dynamique et adaptative à l'évolution des classes et des données. Elle vise à optimiser l'affectation des entités aux classes (*homogénéité au sein d'une même classe*), et à renforcer le mécanisme de la classification incrémentale et la détermination des cas de création des nouvelles classes.

Le mécanisme de classification est adaptatif dans le sens où lorsque la donnée est bien apprenue (*étiquetée*), la classification sera simple dite supervisée (*les entités seront affectées directement à leur classe*). Contrairement, si la donnée est non étiquetée alors la classification sera complexe dite non-supervisée (*possibilité d'affectation dans une nouvelle classe*).

La complexité de notre travail ne se situe pas au niveau du modèle de la classification lui-même, puisque nous allons utiliser les mêmes modèles de classification étudiés précédemment, mais plutôt dans le suivi du nombre et de la dimension des classes.

Dans cette partie conceptuelle, nous allons suivre la méthode de conception **GAIA** [62], décrite dans le chapitre précédent. Nous commençons par définir les entités d'agents avec leurs types (*rôles, responsabilités, connaissances, protocoles*), ensuite nous modéliserons l'interaction entre ces agents en utilisant les diagrammes d'organisation.

2. Hypothèses

Dans cette section nous allons citer quelques hypothèses contribuant à la spécification de quelques caractéristiques des agents classificateurs utilisés :

- 1- Nous supposons que les agents sont synchronisés et connaissent leurs positions (*pour faciliter l'échange d'information*), et que nous disposons d'un protocole de communication entre les agents ;
- 2- Un groupe d'agents est orienté vers un et un seul objectif parmi les objectifs suivants : modélisation, échantillonnage, classification supervisée, classification non supervisée, interprétation des résultats de classification et fusion des interprétations ;
- 3- Au sein du système proposé, il y a un agent superviseur qui gère tous les entités d'agents disponibles. Ce superviseur peut avoir le contrôle des agents du système ainsi de savoir leurs états ;
- 4- Nous supposons que la détection, la modélisation et la caractérisation de données sont faites avec succès par l'agent de modélisation. Notre méthode ne traite pas les modalités, ni les erreurs et les pannes de détection ;
- 5- Enfin, nous supposons que les agents se trouvent initialement à l'état endormi (*à part l'agent superviseur*), et ne commencent leur travail que lorsqu'ils sont sollicités par l'agent superviseur qui déclenche le processus de classification.

3. Problématique

Une fois qu'il y aura une disponibilité de nouvelles données, éventuellement, une évolution de données, le problème consiste à fournir une bonne affectation de ces données dans des classes d'appartenance.

La problématique traitée par notre méthode, vise à chercher la meilleure classe pour chaque donnée et non pas sa classe d'appartenance exacte, et cela en prenant en considération le phénomène de leur non-stationnarité ainsi la diversité de leur types.

De même, la collecte des données manipulées ne fait pas l'objet d'une tâche assurée par notre approche (Hypothèse : 4). Nous ne nous focalisons que sur l'aspect collaboratif et gestion distribuées des communications entre les agents du système.

Reste l'un des défis majeurs de notre méthode, qui se manifeste juste après l'accomplissement du mécanisme de classification. Plusieurs agents interpréteurs (*spécialisés par domaine*) procèdent à l'interprétation des résultats de classification, dans le but de pouvoir fusionner par la suite ces diverses interprétations (*spécialisées*), et avoir une décision finale, répondant aux questions des utilisateurs du système ou les experts du domaine d'application (*propriétaires des données sources*).

4. Les grandes lignes de l'approche proposée

Nous présentons dans cette partie les grandes lignes de notre approche proposée, modélisant un système multi-agents pour une classification dynamique floue de données multimodales et évolutives. Elles sont détaillées ci-après :

4.1. Objectif du système

La classification dynamique floue est une discipline difficile à mettre en œuvre avec une maximisation des performances, où le système doit se doter d'une capacité d'auto-adaptation capable de faire face aux changements brusques de l'environnement d'exécution (*évolution des classes*), ainsi la non-stationnarité des données manipulées. Pour faciliter cette tâche nous sommes inspiré des travaux de recherches récents [29][44][57][1], qui se basent sur la notion de coopérations entre plusieurs méthodes. D'où nous avons orienté notre conception vers un système collaboratif (*à base d'un système multi-agents*) entre plusieurs classificateurs maîtrisant l'ensemble des mécanismes suivants : modélisation de données, échantillonnage, classification et interprétation (Décision). L'objectif est alors d'avoir un ***SMA de classification explicite qui soit objectivement dynamique et floue manipulant des données non-stationnaires et multimodales.***

4.2. Modélisation des données

La méthodologie de modélisation développée s'appuie sur des règles initiées par des experts, afin d'adapter les données hétérogènes et les regrouper dans une base de données globale, facilitant donc leur manipulation et leur échantillonnage avant l'étape de classification. A ce niveau, nous pouvons défier la *multimodalité* de données en appliquant des règles de fusion et de mise à jour sur la base de données globale, et cela, dans le cas d'une évolution ou d'une acquisition de nouvelles données.

4.3. Echantillonnage

Vu la variété des méthodes de classification dynamique (*supervisée et non-supervisée*) disponibles, l'action d'échantillonnage aura pour but de séparer les données apprenties (*avec étiquetage*), et les données non apprenties (*sans étiquetage*) en deux sous bases, à partir de la base de données globale. Par conséquent, nous pouvons traiter par la suite chacune de ces bases séparément, avec des méthodes spécifiques dans le but d'avoir plus de performance.

4.4. Classification

Divisée en deux types, nous avons voulu séparer les méthodes de classification dynamique supervisées et les méthodes non supervisées, afin de tirer avantages de ces deux techniques de fonctionnalités, vers lesquelles les deux sous bases résultantes (*de la phase d'échantillonnage*) seront attribuées. Eventuellement, le phénomène dynamique des classes est traité d'une façon récursive, après chaque étape de classification, si nous détectons une évolution des classes, alors nous allons reprendre le processus de classification.

Le dynamisme de classification est représenté par un taux d'échec des classifications précédentes, sur les *anciennes données déjà classifiées*. Si ce taux est petit alors nous pouvons déduire que les données sont bien affectées et elles sont stationnaires. Sinon, si ce taux est élevé, nous pouvons déduire que les données son non-stationnaires.

4.5. Interprétation et Décision

Le déroulement d'une démarche de classification dynamique conformément à notre approche proposée sera conclu par un dégagement d'une décision finale. Cette dernière est définie par la fusion des différentes interprétations réalisées par plusieurs agents interpréteurs spécialisés, sur la base des résultats des classifications fournies par l'ensemble des agents classificateurs, ayant participé dans le processus de classification. Autrement dit, c'est une *interprétation finale* des résultats de classification de données attendu par les experts.

5. Identification des Agents

Le choix de l'ensemble des agents a été dressé automatiques au cours de la section précédente (*présentation des grandes lignes de l'approche proposée*). Chaque agent doit répondre à des besoins spécifiques définis sur la base des objectifs de l'étape où l'agent intervient. Nous avons opté pour une population d'agents cognitifs suivants : agent superviseur, agent d'échantillonnage, agent de collecte et de modélisation, agents de classification, agents d'interprétation et agent de fusion & Décision.

Tous ces agents cognitifs nécessitent l'utilisation des connaissances sur leur environnement (*les autres agents « plus exactement l'Agent Superviseur », méthodes de classification et de fusion, les sources de données, les règles et protocoles définies par les experts du domaine d'application*).

5.1. L'agent Superviseur (Agent_S)

L'Agent_S joue le rôle du leader et du coordinateur entre tous les autres agents. Il est responsable sur l'envoi des messages de réveil aux agents endormis lorsqu'il ya un travail à faire. De même, il est destinataire des messages d'achèvement envoyés par chaque agent éveillé ayant terminé son travail, et avant son passage à l'état endormi. C'est un agent cognitif, il n'intervient qu'après perception d'un changement dans l'environnement du système.

Dans la phase de classification, l'une des tâches principales de l'agent superviseur est de faire une sélection *adaptative* des classificateurs les mieux adaptés à la nature de données à classifiées (*supervisée ou non-supervisée*). Une fois les résultats de classification seront prêts, l'Agent_S compare l'ensemble des classes résultantes avec l'ensemble des classes initiales ; s'il détecte un changement dans ces classes, il va provoquer une mise à jour sur la base des métas données suivi par une nouvelle réitération de la phase de classification.

5.2. L'agent de collecte et de Modélisation (Agent_M)

La tâche de l'agent de modélisation consiste à préparer les données initiales au processus de classification, et cela par leur adaptation, agrégation et modélisation au sein d'une seule base de données globale. C'est un agent cognitif, en appliquant les règles de modélisation et d'adaptation définies par les experts du domaine, il exécute des requêtes de jointure et d'union sur les tables de données appartenant aux différentes bases de données présentes dans l'environnement du système, afin d'alimenter la base globale.

5.3. L'agent d'Echantillonnage (Agent_E)

L'Agent_E joue le rôle d'une passerelle entre l'étape de préparation de données et l'étape de classification. A partir de la base de la base de données globale, l'agent va la diviser en deux sous bases. Le critère de division est la condition si la donnée est étiquetée ou non (*sa classe d'appartenance est connue ou pas*) ? La réactivité de l'agent d'échantillonnage consiste à parcourir la base de données globale, où pour chaque donnée il va vérifier si la valeur de son attribut « classe » est présente dans la base des métas données (liste des classes

initiales), si c'est le cas, alors la donnée avec ses propriétés sera insérée dans la sous BD étiquetées, dans le cas contraire elle sera insérée dans la sous BD non-étiquetées.

5.4. L'agent de Classification : Supervisée (Ag_CS) / Non-supervisée (Ag_CNS)

Adoptant l'une des méthodes de classification dynamique étudiées précédemment, le rôle d'un agent de classification est d'affecter les données modélisées à leurs meilleures classes d'appartenance. Vu l'importance de la phase de classification dans notre méthode, où ces agents vont intervenir, et vu l'opportunité offerte par les systèmes multi-agents concernant la distribution des tâches entre plusieurs entités intelligentes, nous avons choisi la spécialisation de chaque agent (*méthode supervisée ou non-supervisée*) afin de concentrer le travail de l'agent sur une seule manière de résolution. Cela mène à diviser le problème global en sous problèmes, traités séparément et parallèlement par les différents agents.

5.5. L'agent d'Interprétation (Agent_I)

L'Agent_I permet de valoriser les résultats de classification en leur donnant une interprétation adéquate, sur la base d'un protocole et un langage d'interprétation définie par les spécialistes du domaine. Chaque agent interpréteur est un agent cognitif spécialisé sur l'un des axes du domaine d'application. Leur nombre est défini par les experts.

5.6. L'agent de Fusion des Interprétations (Agent_F)

L'Agent_F présente le résultat ou la décision finale à l'utilisateur, il est responsable de la fusion des interprétations fournies par les agents interpréteurs selon des règles de fusion définies par les connaisseurs du domaine. Il permet à l'utilisateur de visualiser les résultats finaux, et de choisir la bonne interprétation.

6. Architecture générale de la méthode proposée

Notre méthode est le résultat d'une fusion entre cinq phases de traitement de données. Elle est la conjonction de deux processus initiaux : la modélisation et l'échantillonnage de données, suivi par les trois processus celui de la classification, interprétation et enfin la fusion des interprétations et présentation de la décision finale à l'utilisateur.

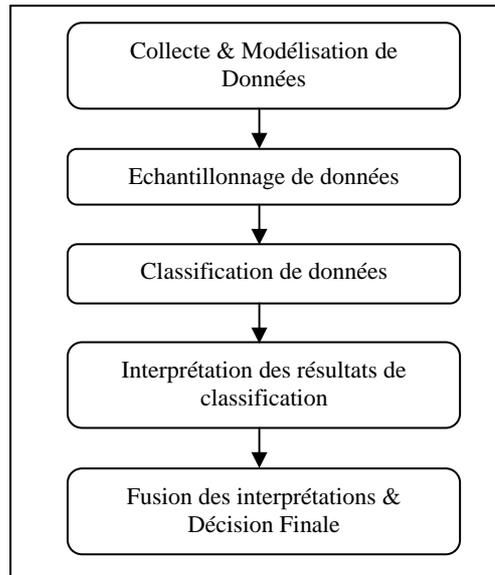


Figure 5.1 : Architecture générale de la méthode proposée.

7. Schéma de fonctionnement de la méthode proposée

Initialement, tous les agents se trouvent à l'état endormi, afin d'optimiser les performances de l'agent superviseur (*qui assure la gestion et le contrôle des agents actifs*) et de minimiser l'utilisation des ressources système (*les unités algorithmiques étant distribuées*), contrairement aux systèmes classiques non-distribués, où tous les modules du système sont compilés est activés en même temps, tout au long de l'exécution globale du système.

Avant de passer à la phase de classification, nous avons deux étapes initiales. L'étape de collecte et de modélisation de données, qui commence au moment de la réception des nouvelles données. L'agent superviseur envoie un message de réveil (**Msg#1**) à l'agent de modélisation lui indiquant qu'une nouvelle source de données est présente dans l'environnement du système. Une fois la modélisation et l'adaptation des nouvelles données est achevée (*selon l'ensemble des règles de modélisation établies par les experts du domaine*), l'agent de modélisation envoie un message d'achèvement à l'agent superviseur, où ce dernier envoie un message de réveil (**Msg#2**) à l'agent d'échantillonnage pour commencer le traitement de la nouvelle base de données globale. L'étape d'échantillonnage a pour but d'affecter les nouvelles données dans l'une des deux sous bases (*supervisée* et *non-supervisée*), dont l'agent superviseur sera informé par leurs mises à jour. Par conséquent, ce dernier pourra choisir les agents classificateurs (**Msg#3** : méthode *supervisée* ; ou **Msg#4** : méthode *non-supervisée*) à réveiller pour commencer le processus de classification.

Les résultats de classification sur les nouvelles données peuvent engendrer l'apparition, l'élimination ou la modification des classes. Un test est appliqué après chaque étape de classification par l'agent superviseur. Dans le cas d'une mise à jour de l'ensemble des classes, une nouvelle phase de classification est exécutée, mais cette fois-ci avec l'ensemble de données globales. Dans le cas contraire (*aucune mise à jour sur l'ensemble des classes*), les agents interpréteurs seront réveillés par l'agent superviseur (**Msg#5**) pour commencer la procédure d'interprétation des résultats de classification (figure 5.2).

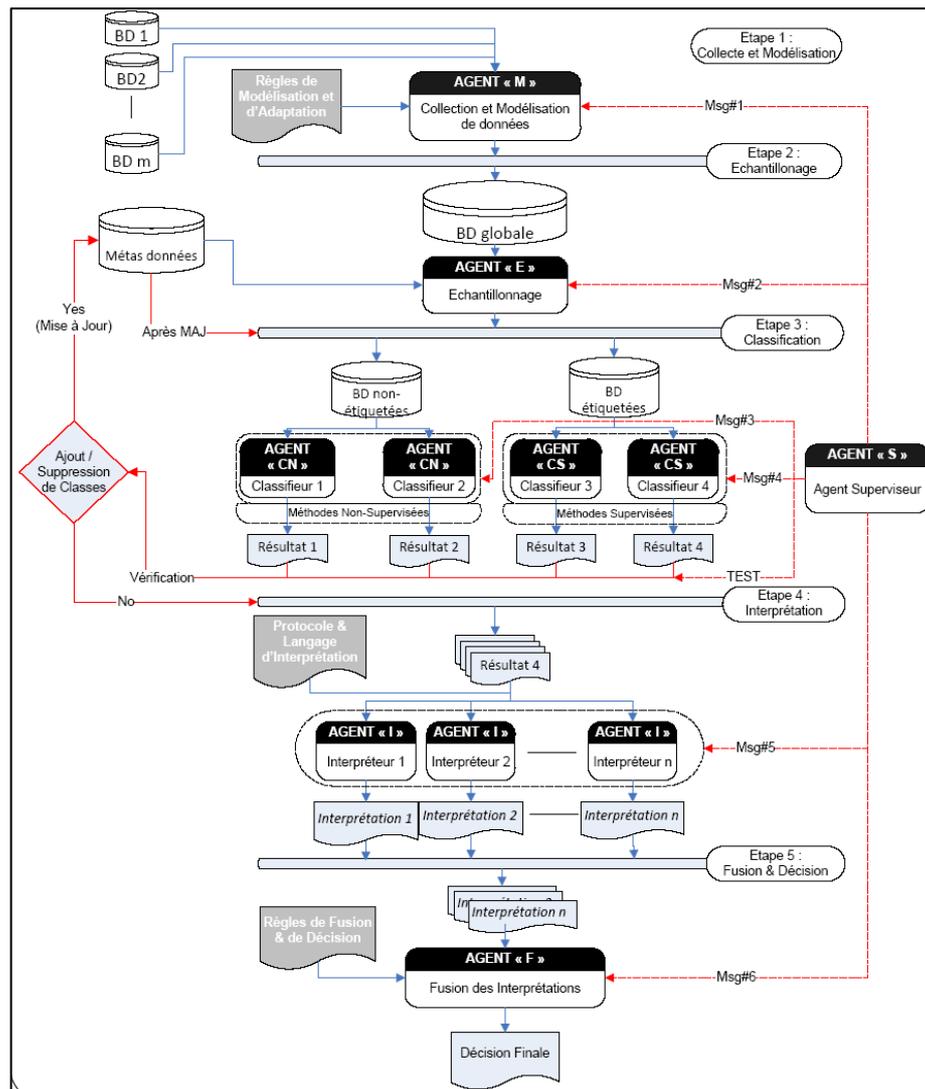


Figure 5.2 : Schéma de fonctionnement global de la méthode proposée.

L'objectif est de fournir des interprétations indépendantes et spécifiques par agent à partir de sa propre base de connaissance (*expériences d'interprétation*), et selon un protocole d'interprétation définis par les experts.

La dernière phase, consiste au déclenchement de la procédure de fusion des résultats d'interprétations une fois quelles seront disponibles. Après la sollicitation de l'agent de fusion & de décision par l'agent superviseur (**Msg#6**). L'utilisateur du système ou le propriétaire des données à classifiées aura comme résultat en sortie une décision finale fournie par l'agent de fusion, conformément aux règles de fusion et de décision prédéfinies par les experts du domaine.

Par la suite nous allons présenter les différentes étapes de fonctionnement de notre méthode, illustrée dans (Figure 5.2), avec des diagrammes d'organisations plus détaillés.

7.1. Etape de Collecte et Modélisation de données (Figure 5.3)

Début

- 1:** Détection des nouvelles données par **Agent_S** ;
- 2:** Envoi d'un message de réveil **msg#1** par **Agent_S** à **Agent_M** ;
- 3:** Accès aux sources de données par **Agent_M** ;
- 4:** Adaptation et agrégation des nouvelles données selon les règles de modélisation;
- 5:** Regroupement des nouvelles données modélisées dans une BD temporaire;
- 6-7:** Jointure Externe entre l'ancienne BD globale et la nouvelle BD
→ Insérer les nouvelles données seulement & Mettre à jour les données existantes évoluées;
- 8:** Envoi d'un message d'achèvement par l'**Agent_M** à l'**Agent_S** → **Agent_M** se met à l'état endormi.

Fin.

7.2. Etape d'Echantillonnage (Figure 5.4)

Début

- 1:** Détection d'une mise à jour de la BD globale par **Agent_S** ;
- 2:** Envoi **msg#2** de réveil de **Agent_S** à **Agent_E**;
- 3:** Etiquetage des nouvelles données à partir de la base d'apprentissage représentée par la BD métas données (*ensemble de classes avec leur fonctions et propriétés d'appartenance*) ;

4-5: Répartition de la nouvelle BD globale sur deux sous bases (*Etiquetées et Non-Etiquetées*). La répartition est appliquée seulement sur les nouvelles données non répartitionnées ;

6: Envoi d'un message d'achèvement par l'**Agent_E** à l'**Agent_S** → **Agent_E** se met à l'état endormi.

Fin.

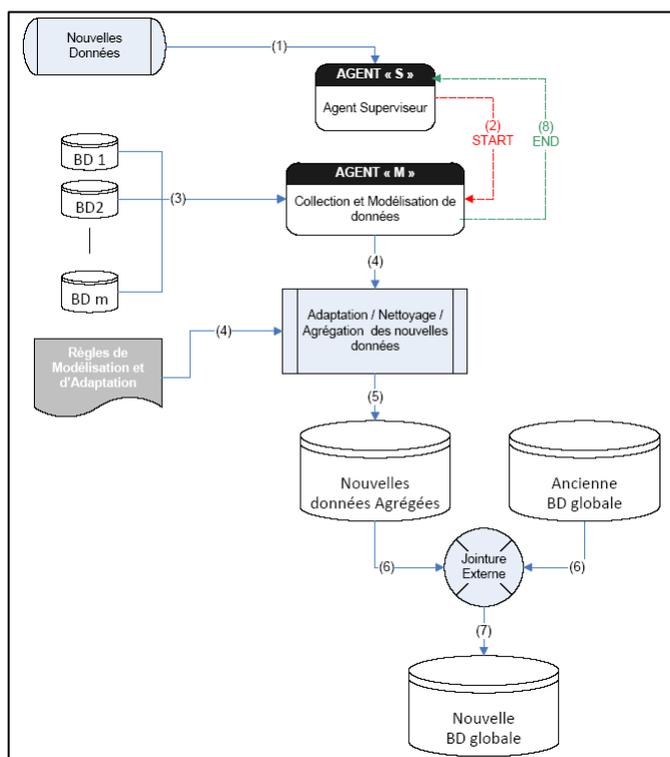


Figure 5.3 : Schéma de fonctionnement de l'étape de Collecte et Modélisation de données.

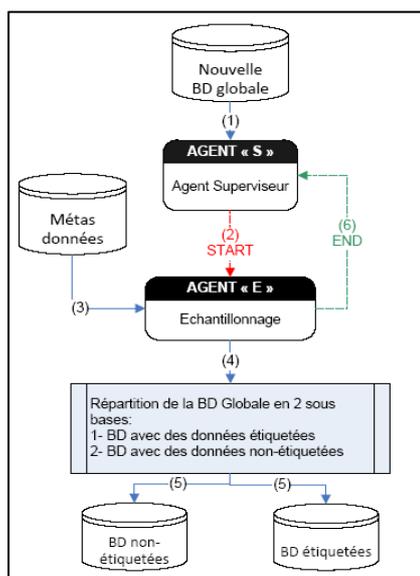


Figure 5.4 : Schéma de fonctionnement de l'étape d'échantillonnage.

7.3. Etape de Classification (Figure 5.5)

Début

1-2: Possibilité d'une mise à jour de la BD étiquetée ou la Base métas données (*Base des classes*);

1''-2'': Possibilité d'une mise à jour de la BD non-étiquetée ou la Base métas données (*Base des classes*);

3: Si (2) est vrais → Envoi **msg#4** de réveil de **Agent_S** aux **Agent_CS**;

3'': Si (2'') est vrais → Envoi **msg#3** de réveil de **Agent_S** aux **Agent_CNS**;

4-5: Classification des nouvelles données étiquetées par les agents **Agent_CS**;

4''-5'': Classification des nouvelles données non-étiquetées par les agents **Agent_CNS**;

6-6'': Envoi des messages d'achèvement par les agents classificateurs **Agent_CS** & **Agent_CNS** à **Agent_S** ;

7: Si **Agent_S** détecte un ajout, élimination ou évolution d'une classe, il applique une mise à jour sur la base des classes (BD métas données) → Réitération d'une nouvelle phase de classification (aller à 1' & 1);

8: Sinon : Fin de la classification → les agents classificateurs se mettent à l'état endormi.

Fin.

7.4. Etape d'Interprétation (Figure 5.6)

Début

1: Détection de nouveaux résultats de classification par **Agent_S**;

2: Envoi **msg#5** de réveil de **Agent_S** aux agents **Agent_I**;

3-4: Chaque **Agent_I** délibère une interprétation selon ses connaissances et sa spécialité, selon un protocole et un langage d'interprétation;

5: Envoi d'un message d'achèvement par les agents interpréteurs à **Agent_S** → **Agent_I** se mettent à l'état endormi.

Fin.

7.5. Etape de Fusion et de décision (Figure 5.7)

Début

- 1: Détection de nouvelles interprétations par **Agent_S**;
- 2: Envoi **msg#6** de réveil de **Agent_S** à **Agent_F**;
- 3-4: Application des règles de fusion et de décision sur les nouvelles interprétations par **Agent_F** → Délibérer une décision final à l'utilisateur du système ;
- 5: Envoi d'un message d'achèvement par **Agent_F** à **Agent_S** → **Agent_F** se met à l'état endormi.

Fin.

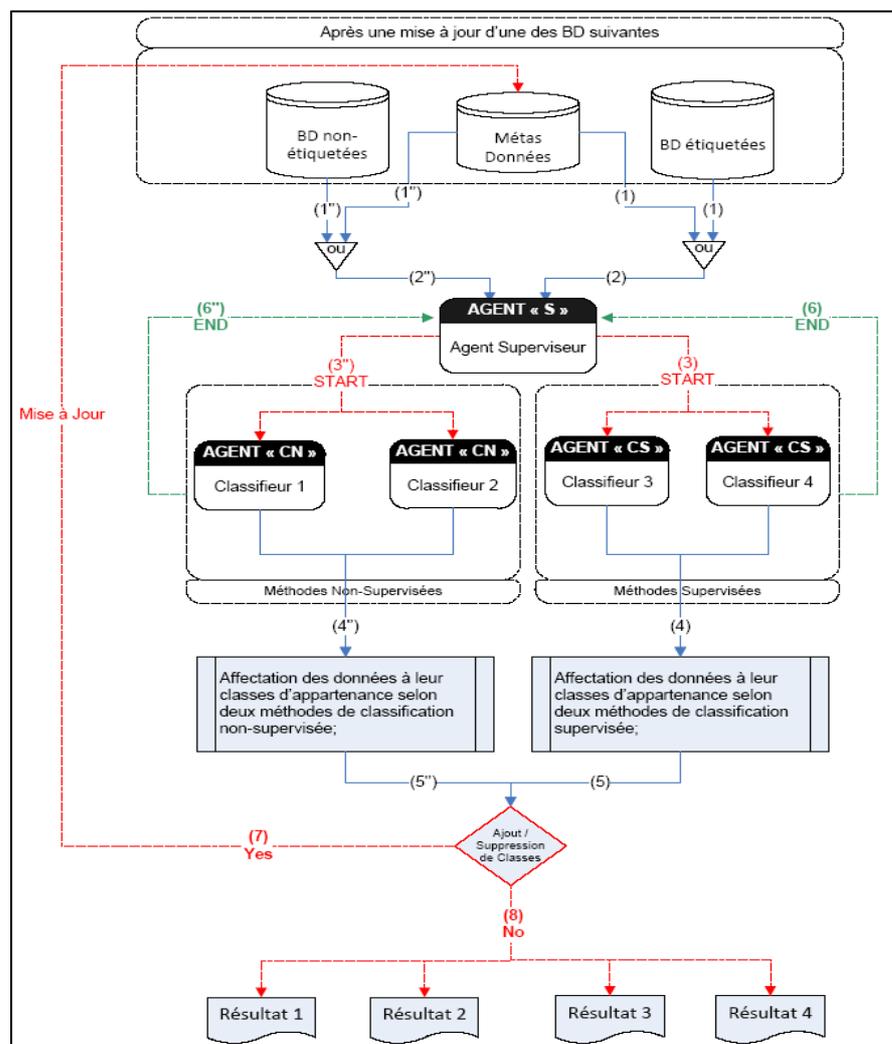


Figure 5.5 : Schéma de fonctionnement de l'étape de classification

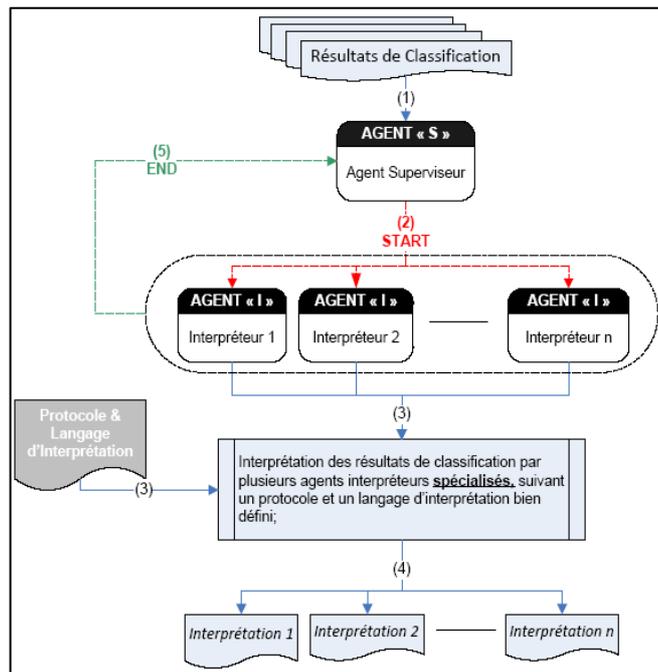


Figure 5.6 : Schéma de fonctionnement de l'étape d'interprétation

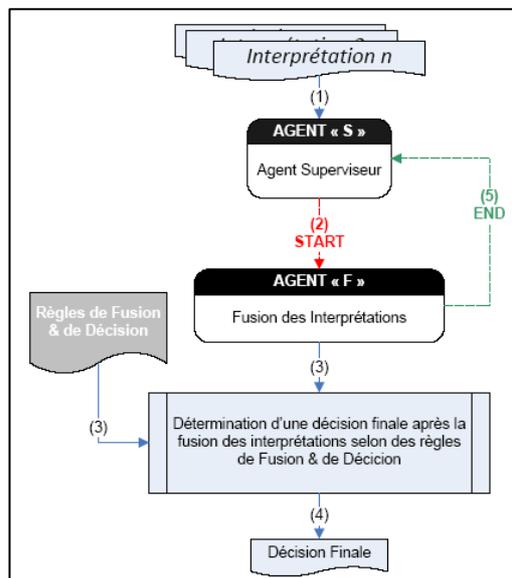


Figure 5.7 : Schéma de fonctionnement de l'étape fusion et de décision

8. Conclusion

Dans cette partie, nous avons conceptualisé un système multi-agents, ayant comme objectif l'adaptation des différentes méthodes de classification dynamique floue avec des données évolutives et multimodales.

Après avoir listé l'ensemble des agents élémentaires pour le fonctionnement du système de classification, nous avons introduit l'architecture générale de la méthode avec les différents niveaux de traitement de données. Ensuite, nous avons défini les différents processus de notre méthode proposée de manière formelle, grâce aux diagrammes d'organisation de données.

Dans la partie suivante, nous présentons une implémentation de notre méthode dans un cadre pratique utilisant la Base de données **METEO(Insolation)** - *issue du Laboratoire de Biologie USTHB.*

CHAPITRE 6

MISE EN ŒUVRE & IMPLEMENTATION

1. Introduction

Dans le but d'évaluer notre approche, une implémentation de quelques méthodes de classification dynamique, proposées dans les chapitres précédents, doit être faite sur des données évolutives, floues et multimodales. L'application développée va encapsuler ces méthodes dans un système complexe, intelligent et distribué basé sur une architecture Multi-Agents. La solution SMA sera évaluée d'une façon quantitative et qualitative, selon différents critères proposés, en fonction de sa capacité de reproduire avec succès le fonctionnement du processus de classification dynamique, ainsi les possibilités qu'elle offre pour enrichir la réflexion du processus décisionnel.

Le présent chapitre décrit l'implémentation et la mise en œuvre de l'approche proposée, afin d'évaluer les performances de l'architecture conçue, avec un déroulement de quelques tests sur des données évolutives, floues et multimodales de type **METEO(Insolation)**.

2. Présentation de la plate forme utilisée

L'implémentation de notre approche conceptuelle, nous a poussée vers la recherche de la plateforme adéquate permettant une bonne élaboration du système multi-agents conçu, dans le but de réussir une bonne schématisation des agents ainsi de réussir un enchaînement facile et juste des interactions entre ces agents. Les critères de choix de la plateforme étés basés sur l'ensemble des contraintes suivantes jugées importantes avec nos attentes:

- Possibilité de communication entre les agents via un échange des messages, où chaque agent est autonome dans sa réflexion et le choix de ses réponses possibles.
- Utiliser une plateforme open source (*gratuite pour les travaux universitaires*).
- Disponibilité des bibliothèques contenant les fonctions fondamentales assurant le bon fonctionnement des agents du système.

- Fiabilité de la plateforme suite à sa réussite au déploiement de multiples projets multi-agents.
- Utilisation d'un langage de programmation adéquat avec l'outil de développement de l'application et ses interfaces.

En fonction de ces critères, nous avons opté pour la plateforme **Jade** (*Java Agent DEvelopment Framework*), qui est une plateforme open source développée en Java conformément aux spécifications FIPA au sein du laboratoire TILAB à l'université de Parma. **Jade** introduit un agent comme un processus autonome possédant une identité, et capable de communiquer avec les autres agents par des messages spécifiés en FIPA-ACL.

2.1. Architecture de la plate-forme JADE

JADE est une plateforme sur laquelle plusieurs agents peuvent vivre, interagir et communiquer entre eux. Elle est composée d'un ou plusieurs conteneurs qui peuvent être exécutés sur des machines différentes (*hôtes distants*), chaque conteneur va contenir zéro ou plusieurs agents. Un conteneur dit principal est obligatoire pour chaque plateforme, il doit être le premier conteneur à lancer dans la plateforme avec les deux agents suivants:

- *Directory Facilitator(DF)* : joue le rôle des pages jaunes ; il permet de mettre en relation les agents avec leurs compétences. Alors, chaque agent doit enregistrer ses compétences dans le *DF* où il peut même interroger le *DF* pour connaître les compétences et services offertes par les autres agents.
- *l'Agent Management System(AMS)* : joue le rôle des pages blanches, il assure la correspondance entre l'agent et son *ID (AID : AgentID)*. Donc, chaque agent est identifié par un *AID* qui sera enregistré dans un répertoire (*AMS*) avec les adresses des agents de la plateforme.

La figure 6.1 représente l'architecture générale de JADE avec ses composants principaux.

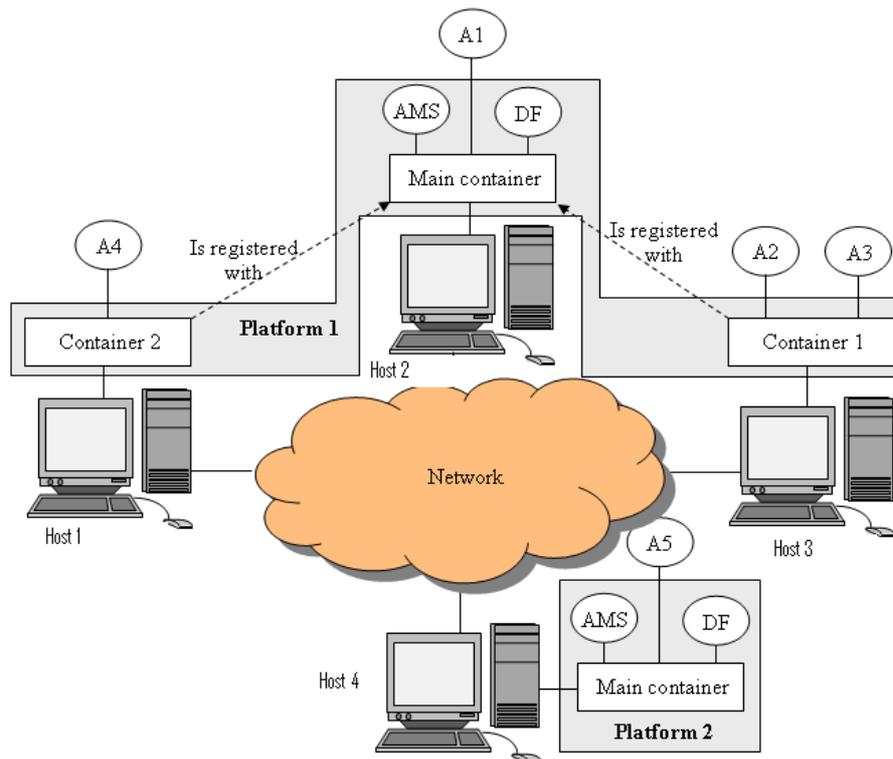


Figure 6.1 : Architecture de la plate forme **JADE**. [<http://jade.tilab.com>]

2.2 Outils de débogage de JADE

La plateforme JADE est enrichie par un ensemble d'outils de débogage mises en service pour l'utilisateur, afin de faciliter sa tâche de développement de son application multi-agent. Chaque outil est défini dans la plateforme comme étant un agent ayant un *AID* unique, un cycle de vie limité et un rôle principal. Nous avons :

- *Remote Management Agent (RMA)* : Il permet de contrôler le cycle de vie de la plateforme ainsi celui de tous les agents exécutés dans cette dernière.
- *Dummy Agent* : C'est une interface d'interaction entre l'utilisateur et les agents de la plateforme à travers l'envoi de messages ACL.
- *Directory Facilitator (DF)* : C'est une interface d'interaction entre l'utilisateur et la *DF* (les pages jaunes de la plateforme).
- *Sniffer Agent* : Cet agent permet d'observer, de tracer et de suivre les messages échangés entre les agents.
- *Inspector Agent*: Il permet de gérer et de contrôler un cycle de vie d'un agent en cours d'exécution ainsi le suivi de ses messages émis et reçus.

2.3. Evaluation de JADE

Voici un tableau récapitulatif d'une évaluation de la plateforme JADE établie par Mikko Laukkanen, suite à un ensemble de tests publié dans son papier [67] :

Critère	Note
Interaction avec l'utilisateur	1/1
Documentation	3.3/4
- Guide développeur	4/4
- Guide utilisateur	3.2/4
- Guide d'installation	3/5
- Documentation Java	3/3
Facilité d'installation	3/3
Disponibilité des fichiers binaires	5/5
Disponibilité des sources	0/4
Avantages techniques	3.7/4.6
- Architecture	4/5
- Conception	4/5
- Implémentation	3.2/4
Facilité de mettre en application des agents	4/4
Futur développement	1/5
Total	21/30.6
Total %	69%

Tableau 6.1 : Résultats sur la qualité de la plateforme JADE1.2 [67].

Selon le tableau récapitulatif, nous pouvons constater que JADE ne demande pas un grand effort à son utilisateur, il est simple à comprendre et à maitre en œuvre, comme il permet facilement le développement des agents. Or nous remarquons bien un manque dans les codes sources ainsi une lenteur dans l'amélioration de l'outil.

3. Choix de la Base de données

Après le choix de la plateforme multi-agent qui sera utilisée dans l'implémentation de notre démarche, il fallait s'assurer que la base de données choisie pour les tests d'évaluation s'accorde bien avec le contexte de notre travail, pour cela nous avons cherché une base de données contenant des données évolutives, floues et multimodales.

Comme nous avons indiqué précédemment, le but principal de notre travail n'est pas une optimisation de l'affectation des données évolutives et multimodales dans les classes, mais plutôt c'est une simulation d'une coopération de plusieurs méthodes de classifications dynamique sur des données évolutives et multimodales via un système intelligent et distribué. Pour cela nous avons opté pour une étude sur la base de données **METEO(Insolation)** représentant le gisement solaire algérien.

3.1 Présentation et description des données

La base de données mise à notre disposition nous provient du Laboratoire de Biologie de l'Université des sciences et technologies HOUARI BOUMEDIENNE (Alger, Algérie), elle est constituée de 52 stations météorologiques du réseau ONM (l'Office National de la Météorologie), couvrant pratiquement le territoire national. Chaque station contient l'enregistrement journalier du paramètre météorologique et climatique « INSOLATION » (en dixième heure, sur une période d'observation de 11 ans [1992-2002]).

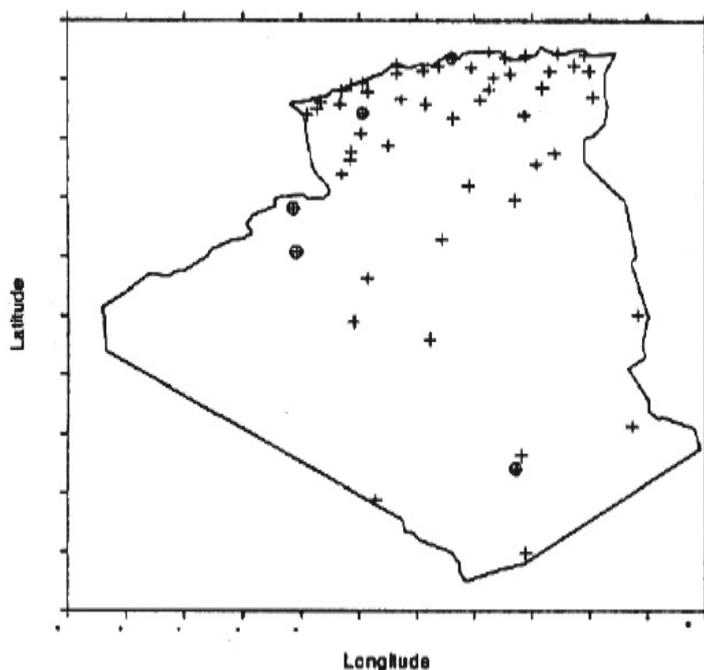


Figure 6.2 : Les stations du réseau météorologique et radiométrique algérien [41].

3.1.1 Description des données

Chaque station est présentée sous forme d'un fichier texte contenant les informations suivantes :

- ➔ Identifiant / Nom Station / Coordonnées géographiques (altitude, latitude, longitude) / Période de mesure ;

Exemple :

- ➔ 620 / ADRAR / (279 m, 27°49 N, 00°11 E) / 1992-2002 ;

Le tableau (6.2) représente les coordonnées géographiques des 52 stations météorologiques stockées dans la base de données **METEO(Insolation)**, à partir des qu'elles nous avons récolté les données expérimentales.

ID	STATION	ALTITUDE	LATITUDE	LONGITUDE
620	ADRAR	279 m	27°49 N	00°11 W
611	AINAMENAS	561 m	28°03 N	09°38 E
560	AINSEFRA	1058 m	32°46 N	00°36 W
360	ANNABA	2 m	36°53 N	07°48 E
453	ARZEW	198 m	35°49 N	00°16 W
471	BARIKA	460 m	35°20 N	05°20 E
468	BATNA	822 m	35°45 N	06°19 E
571	BECHAR	809 m	31°30 N	02°15 W
402	BEJAIA	2 m	36°43 N	05°04 E
602	BENIABBES	500 m	30°08 N	02°10 W
525	BISKRA	82 m	34°48 N	05°44 E
518	BNISAF	68 m	35°81 N	01°21 W
444	BOURJ	928 m	36°04 N	04°46 E
515	BOUSAADA	459 m	35°20 N	04°12 E
419	CONSTANTINE	693 m	36°17 N	06°37 E
390	DAR EL-BEIDA	25 m	36°41 N	03°13 E
387	DELLYS	8 m	36°55 N	03°57 E
670	DJANET	968 m	24°16 N	02°28 E
535	DJELFA	1180 m	34°20 N	03°23 E
425	ECHALEF	143 m	36°12 N	01°20 E
550	ELBAYADH	1341 m	33°40 N	01°00 E
590	ELGOLEA	397 m	30°34 N	02°52 E
559	ELOUED	64 m	33°30 N	06°47 E
566	GHARDIA	468 m	32°24 N	03°48 E
403	GUELMA	227 m	36°28 N	07°28 E
630	INSALAH	268 m	27°14 N	02°30 E
353	JIJEL	2 m	36°49 N	05°43 E
514	KSAR ECHALLALA	800 m	35°10 N	02°19 E
522	MAGHNIA	427 m	34°49 N	01°47 W
506	MASCARA	474 m	35°36 N	00°18 E
549	MECHRIA	1149 m	34°31 N	00°17 W
437	MEDEA	1030 m	36°17 N	02°44 E
430	MELIANA	715 m	36°18 N	02°14 E
457	MOSTAGHANEM	137 m	35°53 N	00°07 W
467	MSILA	441 m	35°40 N	04°30 E
557	NAAMA	1166 m	33°16 N	00°18 W
490	ORAN	90 m	35°38 N	00°36 W
580	OUARGLA	144 m	31°56 N	05°24 E
421	OUMELBOUAGI	889 m	35°52 N	07°07 E
536	SAIDA	750 m	34°52 N	00°09 E
443	SETIF	1033 m	36°11 N	05°15 E

520	SIDIBELABBAS	475 m	35°12 N	00°37 W
355	SKIKDA	2 m	36°53 N	06°54 E
423	SOUKAHRAS	680 m	36°17 N	07°58 E
680	TAMANRASSET	1362 m	22°48 N	05°26 E
475	TEBESSA	821 m	35°25 N	08°07 E
410	TENNES	17 m	36°30 N	01°20 E
511	TIARET	977 m	35°21 N	01°28 E
607	TIMIMOUN	312 m	29°15 N	00°17 E
656	TINDOUF	443 m	27°42 N	08°10 W
531	TLEMCEN ZENATA	246 m	35°01 N	01°28 W
555	TOUGHOURT	87 m	33°04 N	06°05 E

Tableau 6.2 : Coordonnées géographiques des stations météorologiques.

3.1.2 Création de la base de données

Pour chaque station nous avons transformé son fichier texte en un fichier Excel, ensuite nous avons ordonné et consolidé tous ces fichiers dans une seule feuille Excel selon la forme du tableau indiqué ci-dessous, afin de l'exporter dans une base de données Access.

Années	Mois	ID	Station	Jours			
				1	2	-----	31
1992	1	620	ADRAR				
⋮	2	⋮	⋮				
⋮	⋮	⋮	⋮				
⋮	⋮	555	TOUGUOURT				
⋮	12	⋮	⋮				
1996	1	⋮	⋮				
⋮	⋮	620	ADRAR				
⋮	⋮	⋮	⋮				
⋮	⋮	⋮	⋮				
2002	12	555	TOUGHOURT				

Après cette transformation nous avons remarqué que la base de données globale contient des données manquantes, elles sont représentées sous une forme codée (-9999), avec un taux de 31% du nombre total des données, distribué essentiellement dans la période [1950-1979] (figure 6.3). Ce qui nous a conduit à enlever la période allant de 1950 jusqu'à 1991, et établir notre étude sur la période [1992-2002]. Or, il reste toujours quelques données manquantes (soit 519 données manquantes avec un taux de 0.25% sur un nombre total de 205 561 de donnée) (figure 6.4).

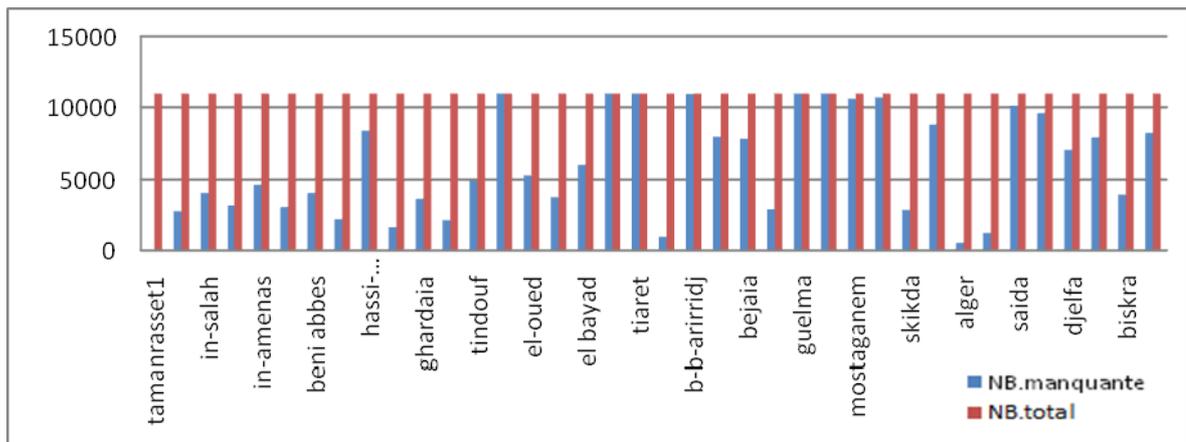


Figure 6.3 : Nombre de données total et manquantes entre 1950 et 1979.

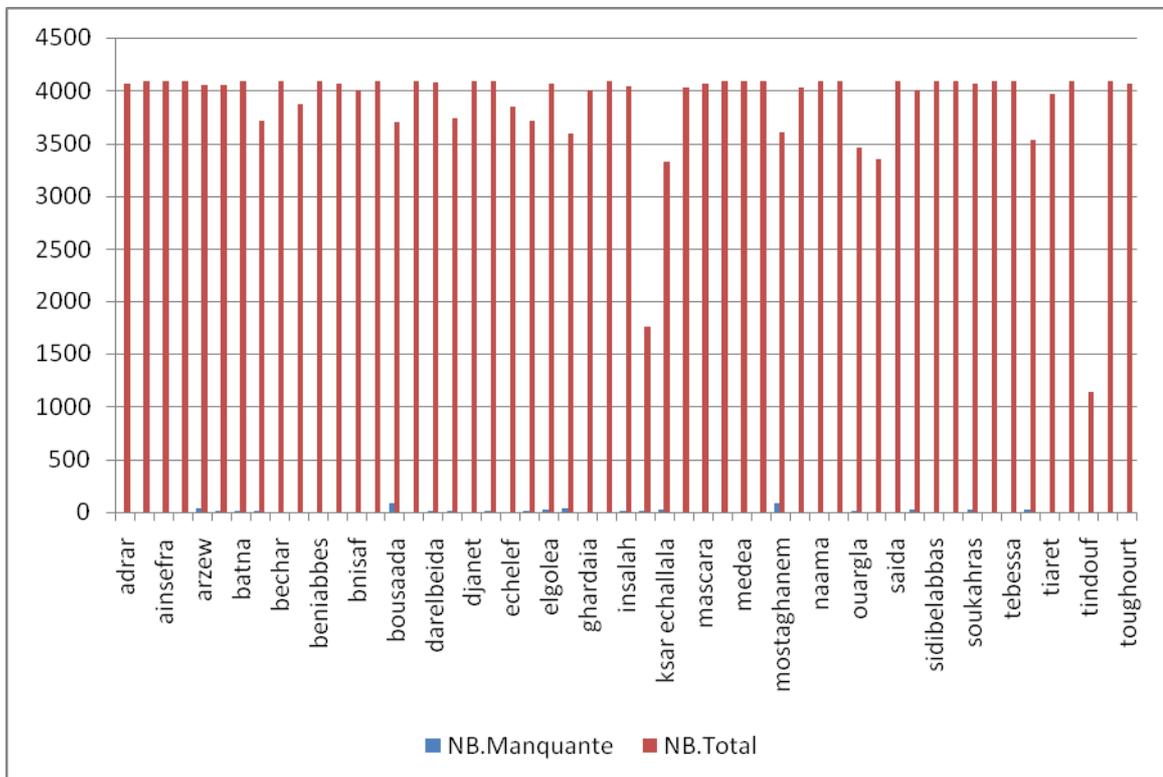


Figure 6.4 : Nombre de données totales et manquantes entre 1992 et 2002.

Pour faire face à ce phénomène, nous avons remplacé les valeurs manquantes comme suit :

- Remplacer les données manquantes situées entre deux valeurs par la moyenne de ces dernières,
- Remplacer les données manquantes successives se trouvant au milieu d'une station par la moyenne des valeurs des deux stations qui la ressemble en termes de climat (la plus proche) dans la même période.

Il faut signaler qu'après toute ces démarches, les données manquantes ont été un handicap pour nous, pour la bonne exploitation de la base de données.

3.2 Caractéristique descriptives des données

Les informations statistiques recueillies à l'état brute sont fastidieux et pratiquement inexploitable, pour leurs donner un sens et une utilité, il faut les ordonner, les classer et les traiter en utilisant comme support les tableaux et les graphes.

La présentation de l'information statistique qualitative ou quantitative sous forme de tableau statistique représente une étape très importante et indispensable pour les opérations statistiques ultérieures comme le calcul des moyennes, de la dispersion, de la régression etc.

3.2.1 Statistique descriptif des données étudiées

STATION	MOYENNE (heure/jour)	ECART-TYPE	MIN (heure/jour)	MAX (heure/jour)	Coeff-Variation
ADRAR	9,3999015	2,3665641	0	13,6	0,2517648
AINAMENAS	9,4252232	2,1618909	0	13,4	0,2293729
AINSEFRA	8,9302938	2,5530012	0	13,6	0,285881
ANNABA	7,8631108	3,0600639	0	14,2	0,3891671
ARZEW	7,7709525	2,8923501	0	13,6	0,3722002
BARIKA	8,7023284	2,5458865	0	13,7	0,2925523
BATNA	8,2544286	2,9083070	0	13,8	0,3523329
BECHAR	9,6329213	2,2705953	0	13,9	0,235712
BEJAIA	7,5650519	3,1695701	0	13,6	0,4189753
BENIABBES	9,7420687	2,1610073	0	13,8	0,2218222
BISKRA	9,3129825	2,4773151	0	13,9	0,2660066
BNISAF	8,2833200	3,0068836	0	13,6	0,3630046
BOURDJ	8,4191550	2,9480245	0	13,7	0,3501568
BOUSAADA	8,4370732	2,8248705	0	13,8	0,3348164
CONSTANTINE	7,8850983	2,8923575	0	13,5	0,3668131
DARELBEIDA	7,9843371	3,0859476	0	14	0,3865002
DELLYS	7,6953615	3,0781830	0	13,5	0,400005

DJANET	9,9890347	1,8693409	0	13,2	0,1871393
DJELFA	8,4593627	2,8811269	0	14	0,3405844
ECHLEF	8,3920412	2,9605118	0	14,2	0,3527761
ELBAYADH	8,3756682	2,6747787	0	13,6	0,3193511
ELGOLEA	9,5290214	2,3152987	0	13,4	0,2429734
ELOUED	9,2417512	2,3363396	0	13,8	0,2528027
GHARDAIA	9,3624300	2,4371886	0	13,9	0,2603158
GUELMA	7,7791649	3,1150196	0	13,8	0,4004311
INSALAH	9,4522276	2,3863591	0	13,4	0,2524653
JIJEL	7,7236087	3,1956691	0	14	0,4137534
KSAR ECHALLALA	8,4772187	2,9684709	0	13,8	0,3501704
MAGHNA	8,6852545	2,8701231	0	13,7	0,3304593
MASCARA	8,2930566	2,8949630	0	13,9	0,3490827
MECHRIA	8,3677313	2,9901907	0	14,2	0,3573478
MEDEA	8,0062387	3,2664064	0	13,9	0,4079826
MELIANA	8,1684711	3,0627711	0	13,6	0,3749503
MOSTA	8,3440352	2,8179373	0	13,7	0,3377188
MSILA	8,7221259	2,7038588	0	14,2	0,31
NAAMA	8,5055050	2,7445680	0	13,6	0,3226814
ORAN	8,4127578	2,7991209	0	13,8	0,3327233
OUARGLA	8,9764712	2,5206037	0	13,7	0,2808012
OUMELBOUAGI	7,8663196	3,0099953	0	13,8	0,3826434
SAIDA	8,4543825	3,0133381	0	13,7	0,3564232
SETIF	8,5063839	2,9133408	0	14	0,3424888
SIDIBELABBAS	8,7191276	2,9165662	0	14,1	0,3345021
SKIKDA	7,7279920	3,0379841	0	14,2	0,3931143
SOUKAHRAS	7,5206100	2,9281208	0	13,5	0,3893462
TAMANRASET	9,3165364	2,3073718	0	13,2	0,2476641
TEBESSA	8,0418393	2,9255327	0	13,6	0,363789
TENNES	8,0449989	2,8319013	0	13,4	0,3520077
TIARET	8,2530649	3,0642981	0	13,7	0,3712921
TIMIMOUN	9,4025416	2,3471474	0	13,6	0,249629
TINDOUF	9,5758042	2,1358528	0	13,4	0,2230468
TLEMCEN ZENATA	8,4448892	2,9406488	0	13,8	0,3482164
TOUGHOURT	9,4141404	2,3459736	0	13,7	0,2491968

Tableau 6.3 : Statistique descriptives des stations.

D'après ce tableau, nous remarquons que l'écart type varie entre 1.87 et 3.27 cela signifie que l'insolation présente un effet variable et dynamique qui est déterminé par une grande activité de la masse nuageuse. Nous remarquons aussi que la moyenne varie entre 7.46 et 9.99 cela signifie que l'Algérie présente une zone particulièrement bien ensoleillée.

3.2.2 Répartition de l'insolation (heure/j) par région et par saison

		Automne	Hiver	Printemps	Eté
Nord	Est	5,34-7,74	3,95-7,82	6,02-10,03	8,71-12,02
	Ouest	5,64-9,00	4,48-8,41	6,09-10,55	8,26-11,99
Centre	Est	6,42-9,67	5,74-9,02	7,27-10,93	7,95-12,22
	Ouest	5,57-9,31	4,40-8,86	6,78-10,68	7,77-11,89
Sud	Est	7,01-10,29	6,82-9,96	7,13-11,28	8,00-11,94
	Ouest	6,55-10,00	6,49-9,73	7,71-11,41	7,59-11,94

Tableau 6.4 : Répartition de l'insolation (*par région et par saison*) sur le territoire national.

L'insolation varie selon les saisons, elle est plus élevée en saison sèche que pendant les autres saisons. En été, le ciel est relativement clair et l'insolation est presque la même dans toutes les régions de l'Algérie. Pendant les autres saisons (*automne, hiver, printemps*) l'insolation croît en allant du nord vers le sud puisque le ciel devient de plus en plus dégagé.

Donc, nous constatons que l'énergie solaire est disponible dans toutes les régions de notre pays et durant toute l'année, mais avec des capacités qui dépendent essentiellement de la saison.

3.2.3 Coefficient de variation de l'insolation mensuelle par région

	Nord		Centre		Sud	
	Est	Ouest	Est	Ouest	Est	Ouest
Janvier	0,37-0,59	0,41-0,49	0,30-0,48	0,41-0,46	0,23-0,26	0,21-0,27
Février	0,40-0,55	0,46-0,49	0,28-0,49	0,42-0,44	0,20-0,25	0,24-0,28
Mars	0,37-0,54	0,39-0,44	0,30-0,41	0,38-0,38	0,22-0,28	0,22-0,27
Avril	0,35-0,51	0,38-0,43	0,28-0,37	0,36-0,37	0,24-0,32	0,20-0,27
Mai	0,32-0,48	0,37-0,42	0,28-0,35	0,33-0,37	0,25-0,33	0,22-0,27
Juin	0,26-0,35	0,27-0,35	0,19-0,27	0,24-0,30	0,20-0,38	0,18-0,23
Juillet	0,16-0,27	0,20-0,27	0,12-0,21	0,19-0,25	0,07-0,28	0,17-0,26
Août	0,21-0,29	0,22-0,27	0,15-0,25	0,23-0,26	0,09-0,29	0,19-0,27
Septembre	0,28-0,36	0,29-0,31	0,25-0,31	0,26-0,3	0,14-0,28	0,23-0,28
Octobre	0,34-0,45	0,36-0,38	0,28-0,36	0,31-0,34	0,16-0,26	0,26-0,27
Novembre	0,44-0,56	0,46-0,49	0,31-0,46	0,42-0,43	0,16-0,27	0,23-0,28
Décembre	0,47-0,67	0,49-0,59	0,34-0,55	0,48-0,54	0,24-0,29	0,24-0,29

Tableau 6.5 : Coefficient de variation de l'insolation mensuelle par région.

D'après ce tableau, nous remarquons que le CV varie beaucoup selon les régions et mensuellement, il est faible dans les régions sud, et plus élevé dans les régions du centre, cela signifie, que l'ensoleillement en Algérie est très variable.

Pour les régions nord et centre, le CV varie énormément pendant l'hiver que l'été, cela peut être expliqué par une forte activité nuageuse en hiver. Or dans les régions sud, le CV varie plus en été que pendant l'hiver, surtout pour la région Sud-est, et cela est dû à l'effet de saison.

3.3 Critères du choix de la base de données

Le choix de la base de données **METEO(Insolation)** été basé principalement sur les raisons suivantes:

- Disponibilité d'un volume très important de données évolutives dans le temps (205 561 données représentant les durées d'insolation journalières sur une dizaine d'années [1992–2002]).
- Existence de quelques enregistrements avec des données manquantes ou floues.
- Multi-modalité des données caractérisant les stations météorologiques (*numériques, textuelles, coordonnées géométriques*).
- Consistance de la base, elle est exploitée dans plusieurs travaux de recherches en classification.
- Disponibilité de l'ensemble des classes, ainsi leurs domaines et fonctions d'appartenance.

Le tableau suivant représente les caractéristiques de la base de données utilisée :

Nom de la BDD	METEO (Insolation)
Définition BDD	Classification des régions géographiques selon l'irradiation solaire globale journalière
Nombre d'enregistrements	6 631
Nombre de Classes	Six zones climatiques dynamiques : elles varient selon la saison (Hiver, Printemps, Eté et Automne)
Qualité BDD	Bien remplie

Tableau 6.6 : Caractéristique de la base de données utilisée.

4. Mise en œuvre

Dans cette section, nous allons définir l'architecture fonctionnelle de notre système (que nous avons nommé **MASDyC** « *Multi-Agent System for Dynamic Classification* »), par une présentation des différents mécanismes de fonctionnement et de l'ensemble des modules de bases utilisés dans la phase d'implémentation, ainsi une brève description concernant les métriques retenues pour l'évaluation du système développé.

4.1 Architecture fonctionnelle

Notre étude est basée essentiellement sur une architecture multi-agents. Les éléments et les modules de base sont des classes d'agents, leur nombre est variable selon le besoin en nombre de classificateurs et d'interpréteurs utilisé dans le système. Dans notre implémentation, nous avons fixé le nombre des agents à onze agents regrouper en cinq packages (tableau 6.7).

Ci-après les principales classes utilisées dans notre implémentation du système MASDyC:

Classe	Package	Description
Classes des agents communicantes avec l'utilisateur		
Agent_main	MASDyC.main	Agent permettant à l'utilisateur de configurer, lancer et arrêter les phases de traitements de données,
Agent_S	MASDyC.main	Agent permettant le contrôle des agents traiteur de données
Classes des agents préparateurs de données		
Agent_M	MASDyC.BeforeClassification	Agent permettant la collecte et modélisation de données
Agent_E	MASDyC.BeforeClassification	Agent permettant l'échantillonnage de données
Classes des agents classificateurs		
Agent_CS	MASDyC.Classification	Agent implémentant le classificateur Supervisé
Agent_CNS	MASDyC.Classification	Agent implémentant le classificateur Non-Supervisée
Classes des agents décideurs		
Agent_I	MASDyC.AfterClassification	Agent permettant d'interpréter les résultats de classification
Agent_F	MASDyC.AfterClassification	Agent permettant la fusion des résultats d'interprétation
Classes des agents évaluateur du SMA		
Agent_Time	MASDyC.evaluation	Agent permettant d'évaluer le SMA selon le critère "TIME"
Agent_Log	MASDyC. evaluation	Agent permettant le suivi et l'observation des agents
Agent_sniffer	//	Agent déjà implémenté dans la plateforme Jade, utilisé pour évaluer le SMA selon le critère "Charge de communication"

Tableau. 6.7 : Principales classes d'agents utilisées dans notre implémentation.

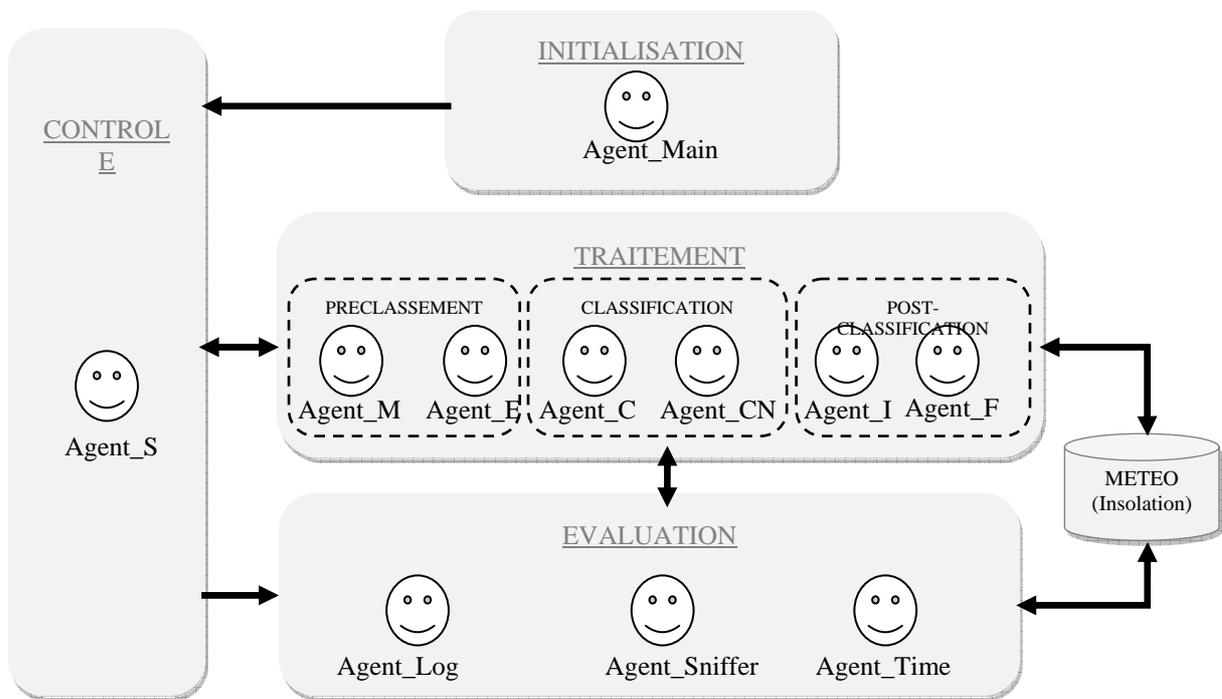


Figure 6.5 : Architecture fonctionnelle du système MASDyC.

Pour la partie fonctionnelle (figure 6.5), notre système s'exécute sur trois phases, en premier lieu nous avons une étape de paramétrage (*Initialisation*) où l'utilisateur a la possibilité de choisir les phases de traitement (*Modélisation, échantillonnage, classification, interprétation et fusion*) ainsi le type de classification à faire (*Supervisée ou Non-supervisée*).



Figure 6.6 : Barre de paramétrage utilisateur.

Une fois les paramètres seront initialisés, le traitement des données commence avec un accès direct à la base de données, où l'utilisateur a la possibilité de suivre l'enchaînement des actions des agents via l'échange des messages de communication entre eux, ces messages seront affichés successivement sur des interfaces de communications (*Boites de dialogue pour chaque agent*). Chaque agent a sa propre boîte de dialogue.

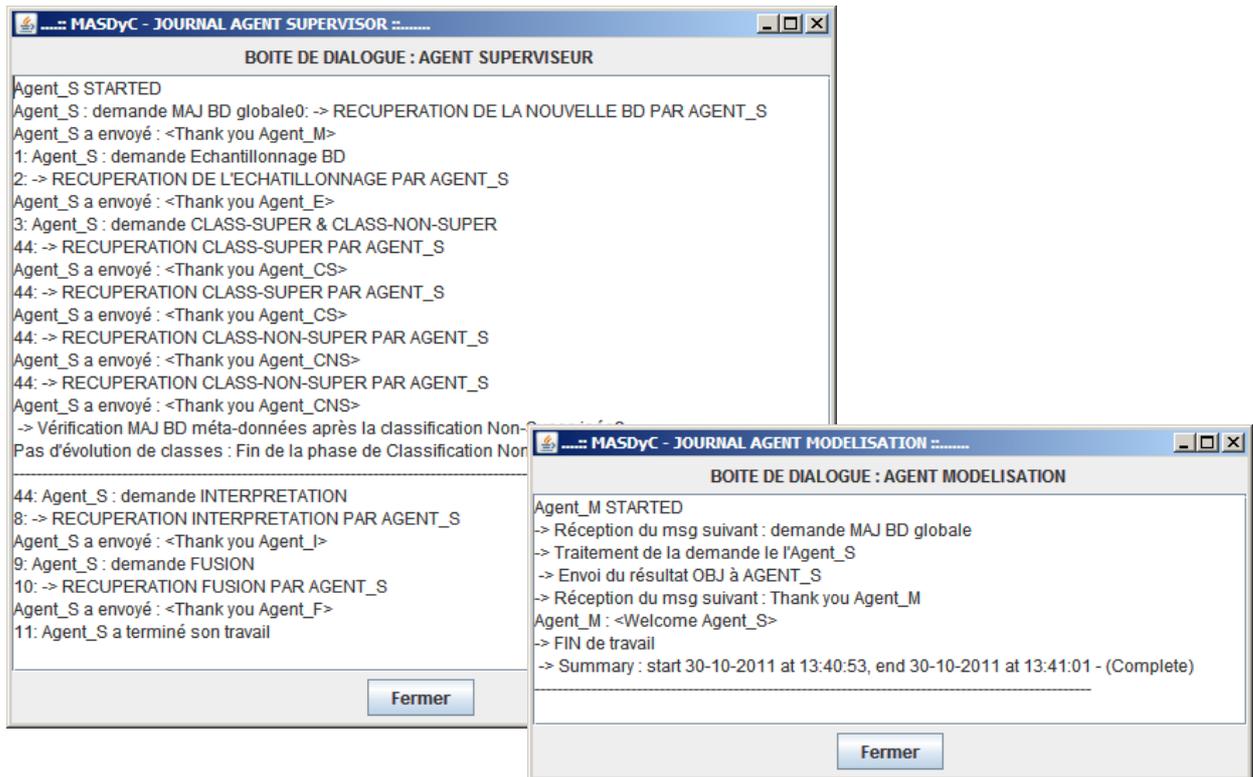


Figure 6.7 : Boites de dialogue pour les agents.

En finalité, après l'exécution de tous les tâches choisies, l'utilisateur va pouvoir visualiser les résultats du traitement ainsi un récapitulatif descriptif représentant les valeurs des critères d'évaluation d'un cycle complet d'exécution de notre système MASDyC.

4.2 Métriques retenues pour l'évaluation

Pour évaluer le bon fonctionnement du système MASDyC nous devons essayer d'examiner un ensemble de questions diverses : est-ce que le système est capable de se comporté selon les objectifs pour lesquels il est conçu ? Est-ce que le système répond efficacement à notre problématique ? Est-ce que l'ensemble des agents du système se comportent d'une façon autonome ? Est-ce que le système est susceptible de suivre l'itinéraire décisionnel défini par l'utilisateur ? Est-ce que le système a réussi de jumeler entre les méthodes supervisées et non-supervisées ? Etc.

Pour y répondre, nous allons analyser le comportement du système ainsi ses résultats finaux une fois après l'accomplissement d'un cycle d'exécution.

MASDyC est un système auto-fonctionnel, l'exécution des tâches par les agents se fait d'une façon autonome, ce qui rend leur contrôle plus difficile dans la pratique. Pour cela, l'agent superviseur va jouer le rôle du contrôleur sur l'ensemble des agents du système, il éveille chaque agent et suit son comportement, via l'échange des messages ACL, durant le cycle d'exécution de cet agent. De plus, l'agent superviseur contrôle le démarrage et l'arrêt du système MASDyC.

Les métriques d'évaluation vont être choisies pour étudier les résultats d'application du système avec la base de données **METEO(Insolation)**. Nous considérons les trois métriques suivantes : le temps, la charge de communication et la précision.

4.2.1 Temps

Nous pouvons considérer que cette métrique (*temps*) est l'une des premières métriques qui doit être utilisée pour l'évaluation d'un système multi-agent [35]. Plusieurs techniques ont été proposées pour étudier cette métrique, soit par un calcul du temps d'exécution CPU (*pour cela les caractéristiques des machines et les plateformes utilisées doivent être prises en compte*), soit par l'addition du nombre d'opérations atomiques réalisées par les agents. Eventuellement, plusieurs auteurs ont proposé d'autres méthodes pour l'estimation du *temps* dans l'évaluation des systèmes multi-agents [42]. Dans notre cas, le calcul du temps CPU pour chaque agent du système est le moyen le plus adéquat et facile pour mesurer cette métrique, vu que l'ensemble des agents seront exécutés sur une seule machine et dans la même plateforme. En effet, nous allons traduire ces mesures de temps par agent T_{agent} en pourcentage par rapport au temps global T_{total} de l'exécution du système, afin d'estimer le gain de temps que nous pouvons obtenir suite à l'utilisation de plusieurs méthodes en parallèle selon différents scénarios.

4.2.2 Charge de communication

Le seul moyen de communication et de collaboration entre les agents du système MASDyC est l'échange des messages. L'étude de la charge de communication permet de déterminer la capacité du système en matière de gestion des coopérations entre les agents, ainsi d'éviter la surcharge du réseau de communication. Le nombre des messages échangés par agent est un bon moyen pour mesurer cette charge de communication, or cette mesure ne rapporte pas assez de valeur sur la capacité du système dans la gestion de la charge de communication. Pour cela nous avons introduit un nouveau critère de mesure représentant le *nombre des messages perdus* par agent. Tant que ce dernier est réduit tant que la charge de

communication est bien maîtrisée. Il existe deux types de message : les messages de coordination (*demandes d'exécution d'une tâche*), et les messages d'information (*publication des résultats suite à l'exécution d'une tâche demandée*). Le nombre total des messages échangés par un agent est lié au nombre de cycle d'exécution de cette agent, plus il y'a du travail à faire plus l'agent a besoin d'échanger des messages.

4.2.3 Précision

Cette métrique permet de mesurer la conformité fonctionnelle du système multi-agent MASDyC ainsi ses résultats d'application avec un système de référence dans le domaine d'application : est ce que notre système est utile ? Est-ce que les entités sont bien classifiées ? La comparaison entre la classification résultante et la classification optimale est un bon critère pour mesurer la précision, or la détermination de cette classification optimale reste un problème majeur qui doit être résolu. Pour notre cas, nous allons utiliser une classification référence pour la régionalisation de l'insolation établie avec la méthode de classification ascendante hiérarchique par une équipe de chercheurs au niveau du laboratoire de biologie de l'université des sciences et technologies HOUARI BOUMEDIENNE, sous la tutelle du Pr. BACHARI (**Annexe**). Elle représente la classification des stations météorologiques nationales selon l'irradiation globale par ciel moyen pour les quatre saisons.

5. Expérimentation

L'expérimentation du système MASDyC a été réalisée sur la base de données **METEO(Insolation)**. Le but du processus de classification est d'affecter chaque station météorologique vers l'une des classes représentant une zone climatique selon le paramètre de classification : ***fraction d'insolation*** (SS/SS_0). Il est défini comme étant la fraction de la durée d'insolation mesurée brute SS (*durées d'insolation journalière enregistrés par la station*) sur la durée d'insolation théorique SS_0 . Cette dernière définie la durée du jour astronomique, elle représente la durée comprise entre le lever et le coucher du soleil pour un lieu donné. Elle évolue naturellement au cours de l'année avec le cycle des saisons, elle dépend uniquement de la latitude du lieu Lat et du numéro du jour dans l'année j . Elle est donnée par la formule suivante :

$$SS_0(j) = (2/15) \text{ Arc } (\cos) [-\text{Tan} (Lat) \text{ Tan } [23,45 \text{ Sin } (0,98 (j+284))]]$$

La fraction d'insolation élimine l'effet déterministe de l'insolation brute (*variation selon la latitude et le temps*), de même elle permet de soustraire l'effet de la tendance saisonnière. Prenant comme exemple la station de Constantine [693 m; 36°17 N; 06°37 E] (figure 6.7 - figure 6.8), où nous pouvons remarquer l'aspect dynamique de l'insolation qui réside dans ses changements brusques.

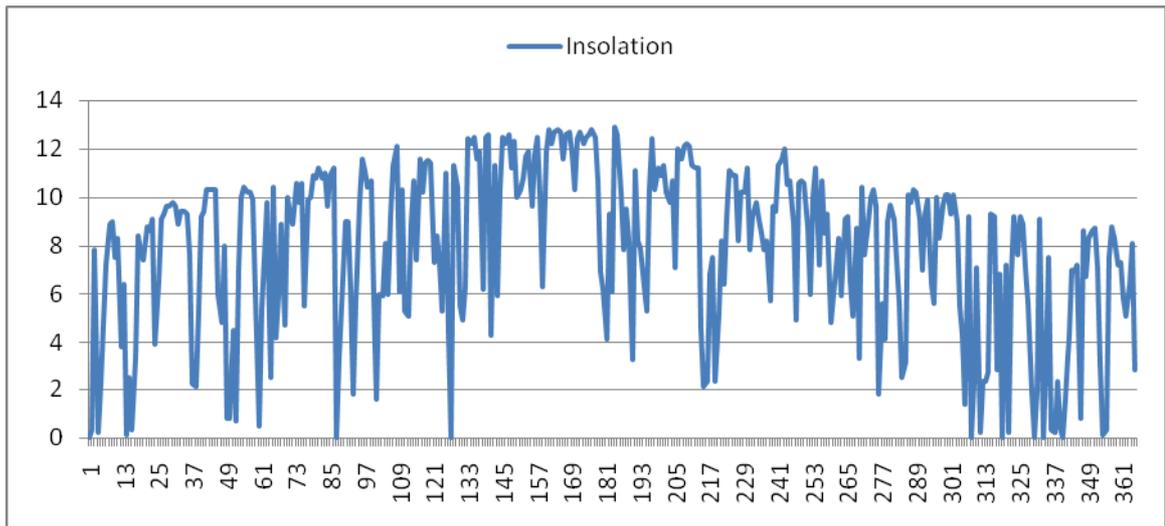


Figure 6.8 : Evolution de l'insolation journalière brute au cours de l'année 2002 dans la station de Constantine.

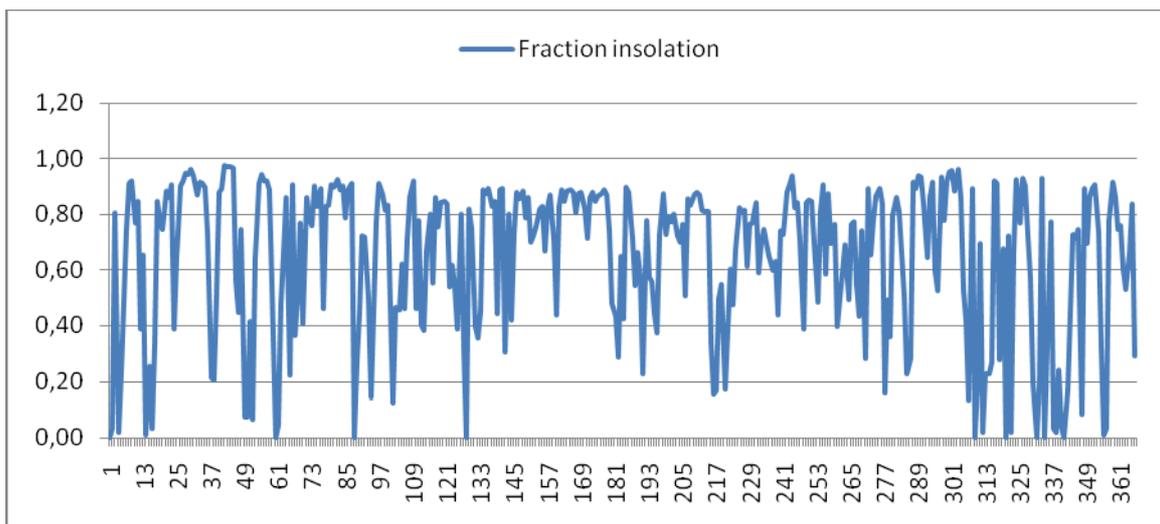


Figure 6.9 : Evolution de la fraction d'insolation journalière au cours de l'année 2002 dans la station de Constantine.

5.1 Application et Résultats

La bonne répartition des tâches entre les agents selon l'architecture fonctionnelle définie précédemment nous a permis d'appliquer facilement notre système MASDyC suivant la conception de base. En effet, les étapes de modélisation et d'échantillonnage ont été basées sur les critères et modèles établis par les propriétaires de la base de données METEO(Insolation) :

- Nettoyage et préparation des données sources (*traitement des données manquantes et floues*) ;
- Agrégation, fusion et normalisation des données (*calcul des fractions d'insolation SS/SS0*) ;
- Alimentation de la base de données globale seulement avec les nouvelles données ou les données évoluées ;
- Répartition des données à partir de la base de données globale sur les deux sous bases : *supervisée* et *non-supervisée*. De même, nous avons fourni à l'utilisateur d'autres paramètres d'échantillonnage : le mois, l'année, la saison et la région.

Le but de la phase de classification été de regrouper les stations dans des classes homogènes. Nous avons confié à l'utilisateur le choix du type de la méthode à utiliser (supervisé et non-supervisé), soit en parallèle soit séparément selon ses besoins. Pour ce qui concerne les deux dernières phases (*Interprétation* et *Fusion* des interprétations), les agents ont été déployée et tester, l'échange des messages d'interactions avec les autres agents été réalisé, sauf les méthodes algorithmiques pour l'interprétation et la fusion n'étaient pas encore disponibles. Eventuellement, il reste toujours la possibilité de les intégrer au cœur de notre système d'une façon élémentaire dans le future, selon les besoins des spécialistes du domaine, afin d'avantager le plus l'utilité de notre approche.

Le (tableau 6.8) représente les résultats obtenus en appliquant notre système de classification dynamique sur la base de données **METEO(Insolation)** avec les deux méthodes supervisée et non-supervisée. Ces résultats présentent la répartition des stations selon la moyenne globale de la fraction d'insolation.

Classes	Stations
Classe n°1	ANNABA, BEJAIA, CONSTANTINE, DELLYS, GUELMA, JIJEL, OUMELBOUAGI, SETIF, SKIKDA, SOUKAHRAS
Classe n°2	BATNA, BNISAF, DARELBEIDA, ECHELEF, MASCARA, MEDEA, MELIANA, MOSTA, TEBESSA, TENNES, TIARET
Classe n°3	BARIKA, BOURDJ, BOUSAADA, DJELFA, ELBAYADH, KSAR ECHALLALA, MAGHNA, MECHRIA, MSILA, NAAMA, ORAN, SAIDA, ARZEW, SIDIBELABBAS, TLEMCCEN-ZENATA
Classe n°4	AINSEFRA, OUARGLA
Classe n°5	ADRAR, AINAMENAS, BISKRA, ELGOLEA, ELOUED, GHARDAIA, INSALAH, TAMANRASET, TIMIMOUN, TOUGHOURT
Classe n°6	BECHAR, BENIABBES, DJANET, TINDOUF

Tableau. 6.8 : Résultats de la classification globale des stations.

La répartition obtenue découpe le territoire national Algérien en six régions climatiques homogènes, comme suit :

- **Classe n°1** : regroupe les régions côtières de l'Est, du centre et l'ensemble des hauts plateaux. Elle est caractérisée par une fraction d'insolation faible dont la répartition est très différente durant l'année entre les mois d'hiver et les mois d'été.
- **Classe n° 2 et 3** : s'étend sur les régions côtières de l'Ouest et l'ensemble des régions intérieur. Elle est caractérisée par une fraction d'insolation faible mais un peu plus élevée que celle de la classe n°1. La répartition est très différente durant l'année. Comme pour la classe n°1, les mois d'été sont beaucoup plus ensoleillés que les mois d'hiver.
- **Classe n°4** : touche une partie du Sahara. Elle est caractérisée par une fraction d'insolation élevée et répartie différemment le long de l'année.
- **Classe n°5** : représente la majeure partie du Sahara. L'insolation totale y est moins élevée et l'effet des saisons est accentué. Les mois d'hiver possèdent une fraction d'insolation plus élevée que les mois d'été, (climat tropical).
- **Classe n°6** : représente le centre et le nord du Sahara, elle est caractérisée par une fraction d'insolation très élevée, répartie presque uniformément durant l'année.

IdClasse	Désignation	MIN (SS/SS0)	MAX (SS/SS0)
C1	CLASSE n°1	0,62	0,65
C2	CLASSE n°2	0,65	0,69
C3	CLASSE n°3	0,69	0,77
C4	CLASSE n°4	0,73	0,77
C5	CLASSE n°5	0,77	0,81
C6	CLASSE n°6	0,81	0,84

Tableau. 6.9 : Classes des zones climatiques résultantes après la classification.

5.2 Évaluation

Pour une meilleure discussion des résultats d'application de notre système MASDyC, nous avons recouru à des agents d'évaluation spécialisés, où chaque agent est dédié à utiliser l'une des métriques d'évaluation définie précédemment.

5.2.1 Analyse suivant la métrique Temps

A partir de la journalisation effectuée par l'*Agent_Time*, nous avons établi une évaluation chronométrique sur les activités des agents du système. Le calcul des taux T_{agent} , représentant la fraction du temps d'exécution d'un agent par rapport au temps d'exécution global du système T_{total} , nous a permis de déterminer les agents qui ont une charge de travail plus importante. Or cette charge n'a pas une grande influence sur la qualité du service offerte par le système, suite à l'exécution des cycles des agents en parallèle.

Dans l'étude de la métrique *temps*, nous avons aperçu un grand rapprochement entre les T_{agent} , soit pour tous agents du système (figure 6.9), soit pour les agents de la même phase de traitement des données : pré-classement (figure 6.10), classification (figure 6.11) et post-classification (figure 6.12).

De même certaines T_{agent} se rapprochent au temps global de l'exécution du système T_{total} (tableau 6.10). Cela explique bien le gain de temps que nous avons obtenu suite à l'utilisation d'un système distribué pour l'analyse de données, car dans un système non distribué nous devons avoir : $T_{total} = \sum T_{agent}$.

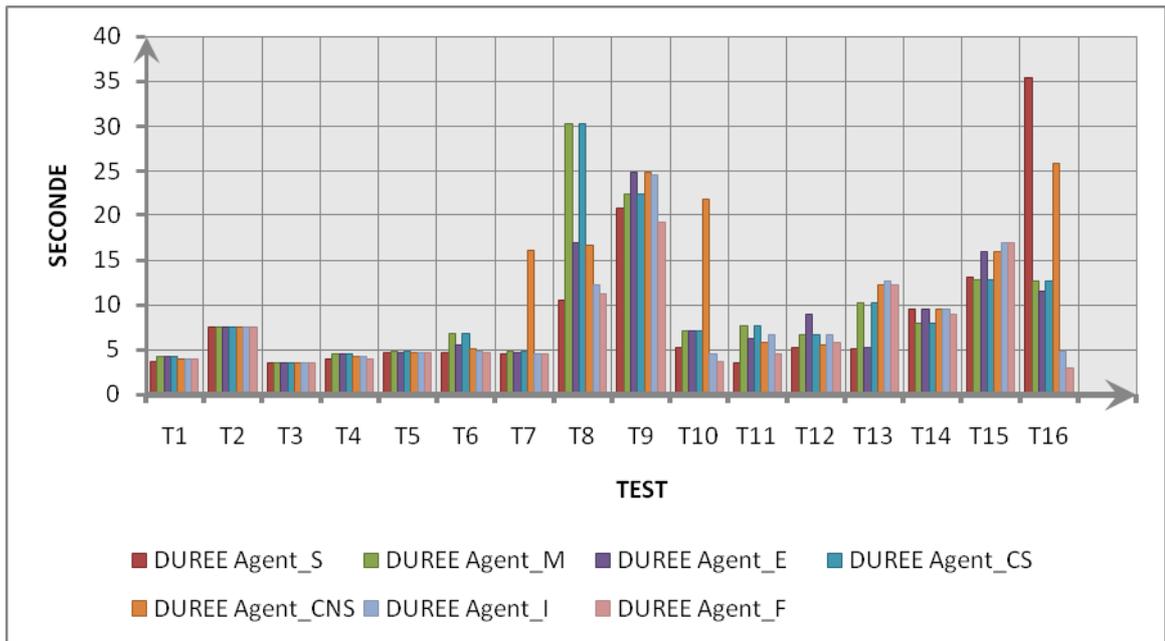


Figure 6.10 : Evolution des durées d'exécution des agents suite à une série de Tests.

Avec une durée d'exécution qui dépasse rarement les trentaines de secondes, nous remarquons sa variation mutuelle chez tous les agents du système durant les séries de tests que nous avons réalisés. Cette variation est expliquée par rapport au volume de données manipulées. Dans les tests **T8** et **T9** les durées d'exécution ont atteint des valeurs maximales, car nous avons introduit des nouvelles données, ce qui a poussé les agents de prendre plus de temps dans leur traitement. Evidemment, juste après les durées se sont rétablies vu que les agents ont déjà traité les données existantes dans la base, reste les types de paramétrage effectués par l'utilisateur comme étant les causes principales de la variation de ces durées.

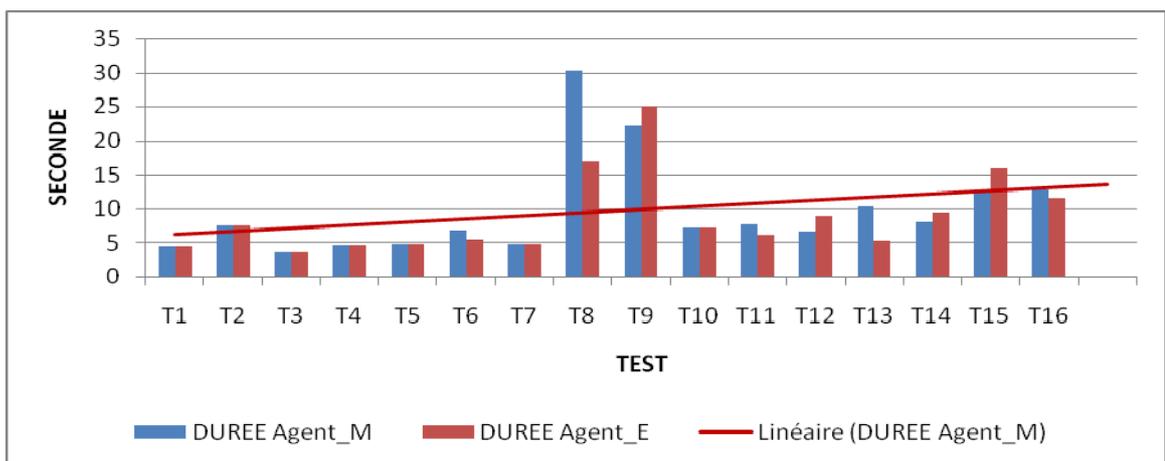


Figure 6.11 : Evolution des durées d'exécution des agents de la phase **pré-classement** suite à une série de Tests.

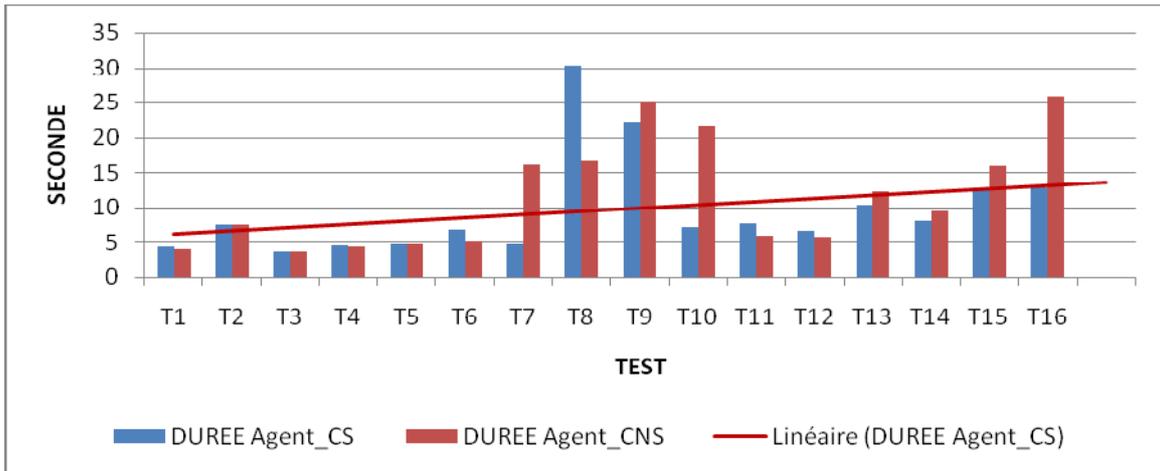


Figure 6.12 : Evolution des durées d'exécution des agents de la phase **classification** suite à une série de Tests.

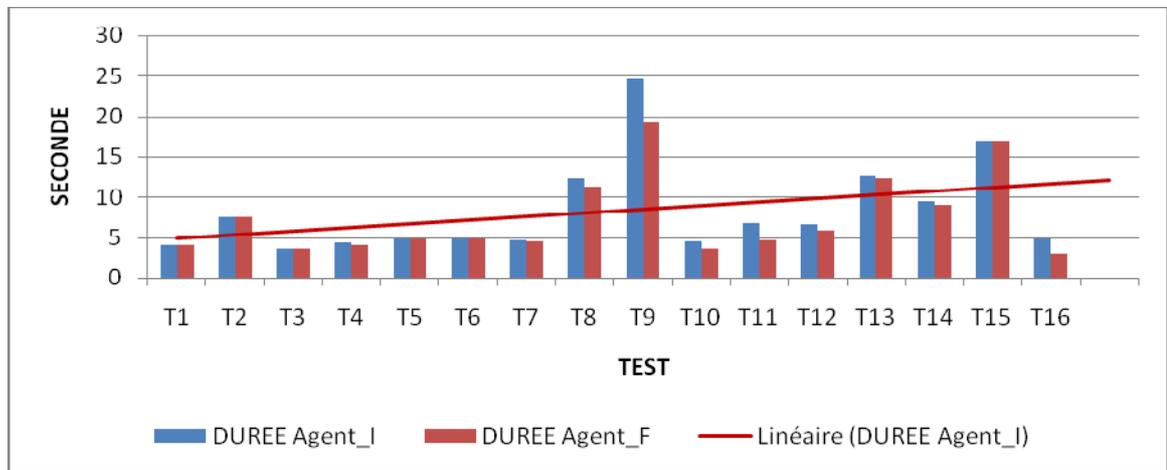


Figure 6.13 : Evolution des durées d'exécution des agents de la phase **post-classification** suite à une série de Tests.

Agent	Moyenne durée d'exécution (T_{agent} / T_{total})	Durée Max (T_{agent} / T_{total})	Durée Min (T_{agent} / T_{total})
Agent_M	47,84%	85,47%	10,21%
Agent_E	40,33%	70,45%	10,21%
Agent_CS	47,84%	85,47%	10,21%
Agent_CN	41,61%	73,01%	10,21%
Agent_I	34,36%	59,51%	9,21%
Agent_F	31,35%	54,24%	8,45%

Tableau. 6.10 : Moyenne globale des durées d'exécution par agent.

Le tableau (6.10) illustre la variation de la durée d'exécution pour tous les agents, avec un pourcentage plus élevé pour certains agents jugés plus actives. Les agents de classification ainsi l'agent de modélisation sont en tête du classement vu le volume de travail qu'ils effectuent. De plus ils doivent répéter leurs tâches dans le cas d'une évolution des données ou des classes. L'agent de fusion et d'interprétation sont en bas du classement à cause du manque des méthodes de traitement des données à leur niveau (en cours d'implémentation).

5.2.2 Analyse suivant la métrique *Charge de communication*

L'échange des messages de communications entre les agents nous permet de suivre l'interaction de chaque agent du système. Pour la capture de ces messages nous avons fait appel à l'*Agent_Sniffer* de la plateforme multi-agents *Jade*. Il permet d'observer, de détecter, de suivre et de tracer chaque message circulant sur une plateforme ou un conteneur d'agents. Une fois le système MASDyC est lancé, il suffit de choisir ses conteneurs au niveau du gestionnaire des agents (l'outil *RMA* de *Jade*), ensuite nous pouvons lancer l'*Agent_Sniffer* dans une nouvelle fenêtre, où la trace de chaque message échangé entre les agents sera affichée sous forme d'une flèche partant de son expéditeur vers son destinataire.

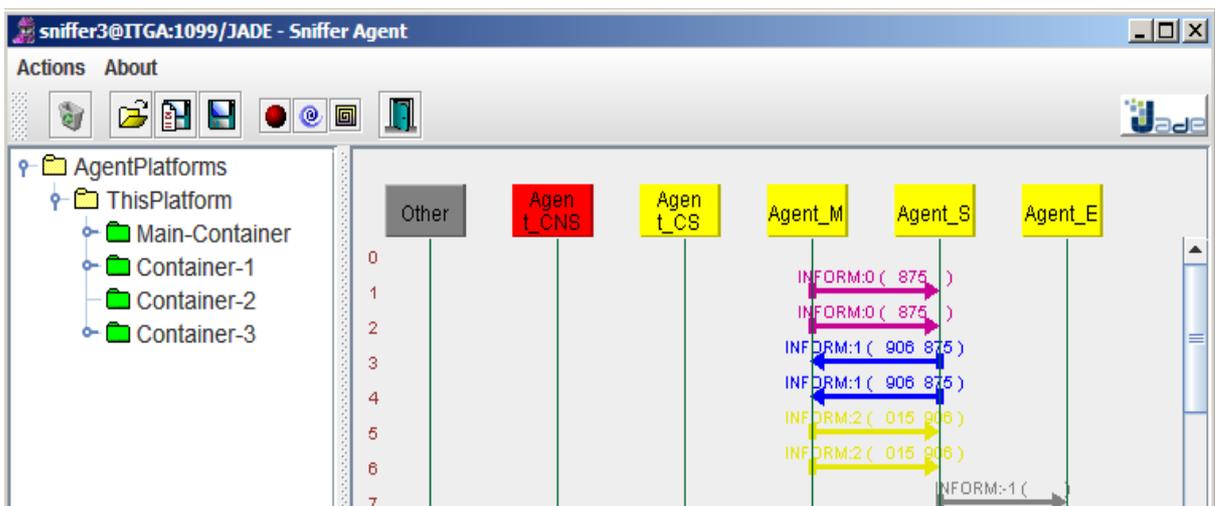


Figure 6.14 : Interface de l'*Agent_Sniffer*.

Pour visualiser le contenu d'un message ACL il suffit de sélectionner sa flèche correspondante (avec un double clic) et lire son contenu : le nom de l'expéditeur, le nom du destinataire et le contenu du message.

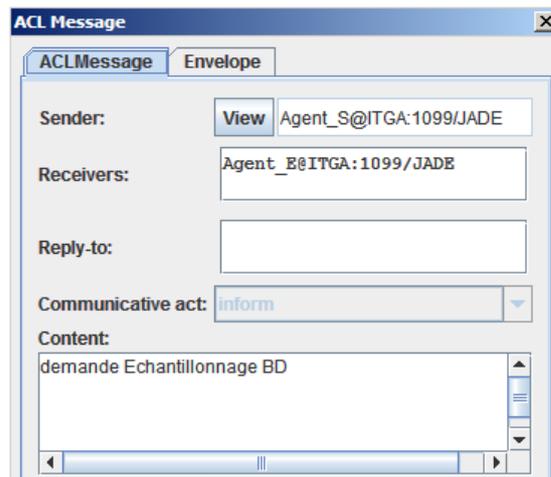


Figure 6.15 : Visualisation du contenu d'un message ACL.

L'*Agent_Sniffer* mentionne un message perdu par l'indice (*FAILURE*) au dessus de sa flèche, un message non perdu est mentionné avec les indices (*INFORM* ou *REQUEST*).

Après la réalisation de plusieurs tests de simulation du système MASDyC sous l'observation de l'*Agent_Sniffer*, nous avons apprécié la qualité de notre implémentation avec un taux de perte de message égale à zéro. Tous les messages parvenaient efficacement à leurs destinataires sans aucune perte.

Ceci est dû à la règle que nous avons introduit dans notre conception, chaque agent doit rester à l'état endormi jusqu'à ce qu'il soit sollicité par l'agent superviseur. Alors tous les agents participant au traitement des données vont communiquer seulement avec l'agent superviseur, afin de se concentrer le plus sur leurs tâches principales et d'éviter de perturber ou de se faire perturber par les autres agents. Cela va exclure la possibilité de saturation du réseau de communication.

L'agent superviseur ne va jamais accéder aux données, il sera plutôt occupé par la coordination du travail collaboratif entre les agents. Donc un seul agent, dégagé de toute sorte de tâche de traitement de données (toujours disponible pour répondre et agir sur place), est destinataire de tous les messages qui proviennent de tous les autres agents impliqués dans le traitement des données (ils ont un accès fréquent à la base de données). Ces derniers sont caractérisés par une lenteur dans la lecture des messages reçus.

Dans la figure (6.13), nous avons un historique des messages échangés entre tous les agents du système au cours d'une simulation d'un cycle complet de classification, incluant tous les phases de traitement possible. Soit une totalité de 53 messages, la majorité d'entre elles (78%) considèrent l'agent superviseur comme étant un expéditeur ou un destinataire.

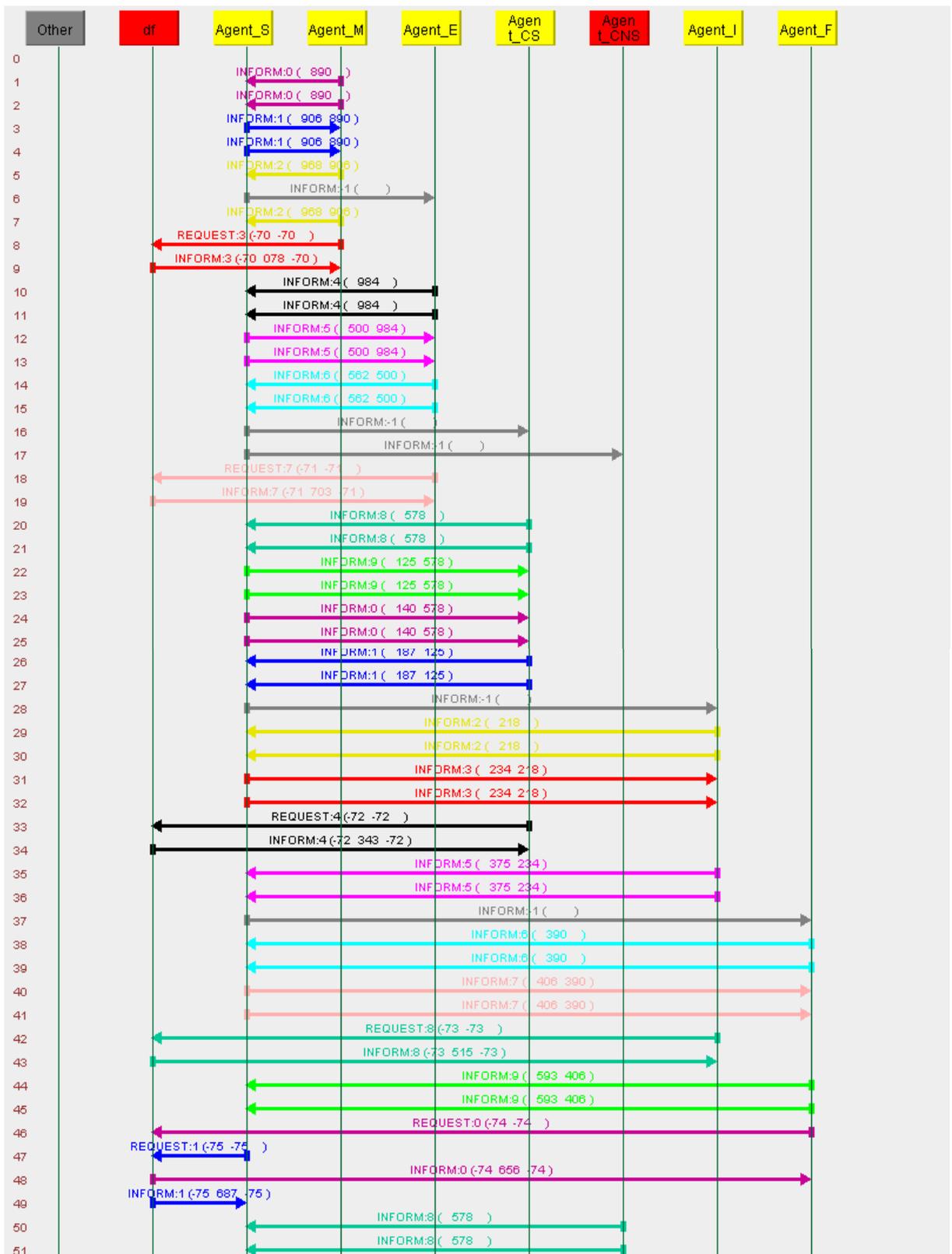


Figure 6.16 : Charge de communication entre les agents pour un cycle d'exécution complet.

5.2.3 Analyse suivant la métrique *Précision*

L'objectif de cette analyse est de vérifier l'adéquation de notre système de classification avec le domaine d'application. A cet effet, nous avons associé à chaque classe résultante de notre classification sa métrique de précision relative. La précision représente la probabilité qu'un objet prédit de la classe C_i soit effectivement de la classe C_i . Soit :

- P_c est un objet correctement classifié dans la classe C_i ;
- P_f est un objet mal classifié dans la classe C_i ;

La métrique de précision Pr est calculée pour chaque classe selon la formule suivante :

$$Pr = P_c / (P_c + P_f)$$

Le tableau (6.12) présente les taux de précision des résultats de notre classification par rapport aux résultats de référence détaillés dans l'annexe. Cette dernière classifie 30 stations météorologiques dans 6 classes homogènes selon l'irradiation globale (tableau 6.11). Donc l'estimation des taux de précisions est basée sur la classification de ces 30 stations.

Classes	Stations
Classe n°1	MSILA, DARELBEIDA, ANNABA, BATNA, BOURDJ, BEJAIA, BEJAIA, SETIF, SKIKDA
Classe n°2	BNISAF, MOSTA, ORAN, TLEMCEN ZENATA
Classe n°3	Al-kheiter, DJELFA, ELBAYADH, SAIDA, ECHELEF
Classe n°4	BISKRA, ELGOLEA, ELOUED, GHARDAIA, TOUGHOURT
Classe n°5	TAMANRASET, AINAMENAS, DJANET
Classe n°6	ADRAR, BECHAR, BENIABBES, INSALAH, TIMIMOUN

Tableau. 6.11 : Résultats de la classification référence.

Classe i	Taux de précision
CLASSE n°1	75,00%
CLASSE n°2	50,00%
CLASSE n°3	37,50%
CLASSE n°5	33,33%
CLASSE n°6	66,67%

Tableau. 6.12 : Taux des précisions estimées sur l'ensemble des classes.

Dans ce tableau la classe n°4 n'est pas introduite, parce que ses entités ne figurent pas dans la liste des 30 stations choisies pour le calcul des taux de précision.

Avec une moyenne du taux de précision dépassant les 50%, notre méthode de classification nécessitent plus d'amélioration pour augmenter cette métrique. L'utilisation d'autres résultats de classification comme référence dans l'analyse de la métrique de précision seraient plus favorable pour l'évaluation de notre système, ainsi l'implémentation des méthodes de classification adéquates au domaine d'application augmente sa fiabilité.

6. Conclusion

La mise en œuvre d'une solution de classification dynamique basée sur une approche multi-agents nous a permis de faire coopérer et exécuter les deux types de méthodes *supervisée* et *non-supervisée* simultanément, afin d'élargir l'espace de données *étiquetées* et *non-étiquetées* traitées. De même, la décentralisation intelligente des tâches entre les agents a garanti une rapidité dans l'exécution de chaque phase du processus décisionnel d'une façon autonome et parallèle.

Dans ce chapitre nous avons procédé à une implémentation de notre approche par une simulation sur la base de données **METEO(Insolation)**, afin de valider la technique proposée. Or, une validation analytique ou expérimentale est plus exacte que la validation par simulation, mais elles sont plus complexes à réaliser en réalité. Notre technique a permis de renforcer :

- La manipulation de données évolutives et multimodales (*collecte et échantillonnage*),
- La détection des évolutions des données et des classes,
- La répartition adéquate des tâches entre les agents,
- La rapidité dans l'exécution du processus décisionnel.

CONCLUSION GENERALE ET PERSPECTIVES

Le travail présenté dans ce mémoire s'insère dans le domaine de la classification dynamique floue à partir des données multimodales et évolutives. Au départ, nous avons évoqué une bonne synthèse bibliographique des méthodes existantes pour la classification dynamique de données. Il y a deux types de classification : classification supervisée et classification non supervisée. Ces derniers sont aussi classés selon deux catégories : les méthodes floues et les méthodes non-floues. Cependant, en combinant les deux types de méthodes : supervisées et non-supervisées, nous avons proposé une architecture de classification coopérative et adaptative à base d'un système multi-agents. L'approche développée est composée de cinq phases : acquisition et stockage des données, échantillonnage des données, classification des données, interprétation des classifications et fusion des interprétations.

Nous avons simulé le comportement de cette approche avec une base de données évolutive caractérisant les mesures d'insolation journalières enregistrées par un ensemble de stations météorologiques réparties sur le territoire national Algérien. En coopérant l'ensemble des agents classificateurs supervisée et non-supervisées, ainsi les agents de modélisation et d'interprétation sous la supervision d'un seul agent superviseur, le système a donné des résultats satisfaisants en termes d'amélioration du temps d'exécution et de la flexibilité dans le choix de la méthode de classification adéquate d'une façon automatique. Cela a permis de réduire considérablement la complexité du processus décisionnel, par une séparation entre les phases de traitement, ainsi de réduire le temps d'obtention des résultats décisionnels, tout en garantissant de bonnes performances en classification suite à la maîtrise de la non stationnarité des données et des classes.

Nous nous sommes placés dans une étude standard sans aucun domaine d'application précis, ce qui nous a mis dans un cadre plutôt défavorable où nous avons trouvé des difficultés dans la validation de notre travail. Mais plus au moins, l'utilisation des trois métriques d'évaluation du système multi-agents à savoir : le temps, la charge de communication et la précision, nous a donné une confiance à notre contribution suite aux résultats obtenus.

En résumer, nous avons pu avec ce travail de concevoir une nouvelle stratégie de classification dynamique coopérative et adaptative pouvant utiliser un nombre variant de méthode.

Dans la suite de ce travail, nous suggérons de choisir un domaine d'application bien précis afin d'adapter notre solution sur des bases et des règles solides. De même, nous devons améliorer les techniques des phases d'interprétation et de fusion des interprétations.

ANNEXE

CLASSIFICATION ASCENDANTE HIERARCHIQUE

La CAH est une méthode de classification permettant une construction de hiérarchies indicées. On parle de classification hiérarchique, car chaque classe d'une partition est incluse dans une classe de la partition suivante. Un menu de sélection associe plusieurs métriques pour passer d'un groupement à l'autre. Ce que l'on peut dire c'est que chaque niveau de hiérarchie représente une classe et que des individus appartiennent à ces classes. La représentation visuelle est beaucoup plus explicite.

Principes de la méthode

La classification ascendante hiérarchique (CAH) est une méthode de classification itérative dont le principe est simple :

- On commence par calculer la dis-similarité entre les N objets.
- Puis on regroupe les deux objets dont le regroupement minimise un critère d'agrégation donné, créant ainsi une classe comprenant ces deux objets.
- On calcule ensuite la dis-similarité entre cette classe et les N-2 autres objets en utilisant le critère d'agrégation. Puis on regroupe les deux objets ou classes d'objets dont le regroupement minimise le critère d'agrégation.

Les objectifs de la classification (CAH)

La classification a pour principal objectif de rassembler les éléments (individus ou variables) qui se ressemblent et/ou de séparer ceux qui diffèrent. C'est-à-dire qu'il s'agit de créer des classes homogènes les plus éloignées les unes des autres. Si cet objectif est facilement compréhensible, il n'en est pas moins compliqué à atteindre. Nous sous-entendons lorsque nous cherchons à classer des éléments, qu'il existe des regroupements, soit en nombre inconnu soit en nombre supposé, si nous cherchons souvent à regrouper des éléments entre eux, c'est afin de mieux interpréter une grande quantité de données.

		VARIABLES		
		1 k K
INDIVIDUS	1	<div style="border: 1px solid black; padding: 10px; display: inline-block;"> <div style="display: flex; flex-direction: column; align-items: center; justify-content: center;"> <div style="display: flex; flex-direction: column; align-items: center;"> ⋮ ⋮ ⋮ ⋮ ⋮ </div> <div style="display: flex; align-items: center; justify-content: center;"> x_{ik} </div> </div> </div>		
	⋮			
	⋮			
	i			
	⋮			
	I			

Tableau: Représentation des données pour la classification.

Le critère d'agrégation

La distance saut minimal constitue un critère de classification selon le saut minimal fondé sur le principe qu'on agrège, après le calcul de la matrice de distance entre les n individus, les deux individus les plus proches en un seul élément. Cependant, et pour certaines applications, ce critère peut conduire à des résultats critiquables.

D'autres critères d'agrégation donnent éventuellement des résultats plus fiables, par exemple les techniques d'agrégation selon la variance qui cherchent à optimiser, à chaque étape, selon des critères liés à des calculs d'inertie, la partition obtenue par agrégation de deux éléments. Cette technique est particulièrement aisée à mettre en œuvre lorsque l'agrégation est effectuée après une analyse factorielle, les objets à classer étant repérés par leurs coordonnées sur les premiers axes factoriels.

Méthodes d'agrégations

Plusieurs méthodes d'agrégation sont disponibles :

- Méthode de Ward (inertie).
- Méthode de Ward (variance).
- Lien complet.
- Lien simple.
- Lien fort.
- Lien flexible.
- Lien moyen.
- Lien proportionnel.

Notations

Nous considérons les n objets à classer comme un nuage de points (nuage des individus) d'un espace à p dimensions (espace de variables), et chaque point x_i (vecteur à p composants) est muni d'une masse m_i , on a :

- $m = \sum_1^n m_i$: la masse totale ;
- $\|x_i - x_i'\|^2 = d^2(x_i - x_i')$: le carré de la distance entre le point x_i et x_i' ;
- $I = \sum_1^n m_i \|x_i - g\|^2$: l'inertie totale du nuage avec $g = \frac{1}{m} \sum_1^n (m_i x_i)$ centre de gravité du nuage.
- s'il existe une partition de l'ensemble des éléments en s classes, la q^{ieme} classe a pour masse $m_q = \sum_{i \in q} m_i$ et pour centre de gravité $g_q = \frac{1}{m_q} \sum_{i \in q} (m_i x_i)$

La relation du Huygens fournit une décomposition de la quantité I en inerties intra classes et interclasses suivant la formule :

$$I = \underbrace{\sum_q m_q \|g_q - g\|^2}_{\text{inter-classe}} + \underbrace{\sum_{q_i \in q} \sum m_i \|x_i - g_q\|^2}_{\text{intra-classe}}$$

Le principe de l'algorithme selon la variance à rechercher à chaque étape une partition telle que la variance interne de chaque classe soit minimale et par conséquent la variance entre les classes soit maximal.

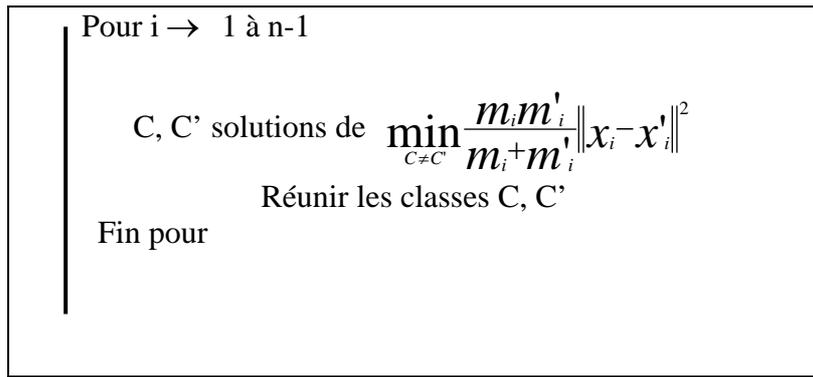
A l'étape initiale, l'inertie intra-classes est nulle et l'inertie interclasses est égale à l'inertie totale. Par conséquent au fur et à mesure que l'on effectue des regroupements, l'inertie intra-classes augmente (l'inertie interclasses diminue) d'une quantité donnée égale à :

$$\Delta_s = I_{\text{inter}}(p_s) - I_{\text{inter}}(p_s - 1) = \frac{m_i m_i'}{m_i + m_i'} \|x_i - x_i'\|^2 = \frac{m_i m_i'}{m_i + m_i'} d^2(x_i, x_i')$$

La stratégie d'agrégation fondée sur le critère de la perte d'inertie minimale, dit critère de Ward, est la suivante : au lieu de chercher les deux éléments les plus proches, on cherchera les éléments x_i et x_i' correspondant à Δ_s minimale.

L'algorithme de la classification selon le critère de Ward est comme suit :

Soient C_1, \dots, C_n partitions ou toutes les classes ont un élément (individu).



Le choix du nombre de classes

L'algorithme ne fournit pas une partition en q classes d'un ensemble de n objets mais une hiérarchie de partitions, se présentant sous la forme d'arbres appelés également dendrogrammes et contenant $n-1$ partitions. L'intérêt de ces arbres est qu'ils peuvent donner une idée du nombre de classe existant effectivement dans la population.

Le choix du niveau de la coupure, et ainsi du nombre de classes de la partition peut être facilité par une inspection visuelle de l'arbre. La coupure doit être faite après les agrégations correspondant à des valeurs peu élevées de l'indice, qui regroupe les éléments les plus proches les uns des autres, et avant les agrégations correspondant à des valeurs élevées de l'indice qui associent les groupes bien distincts dans la population. Donc on obtiendra une « bonne » partition en coupant l'arbre au premier saut significatif.

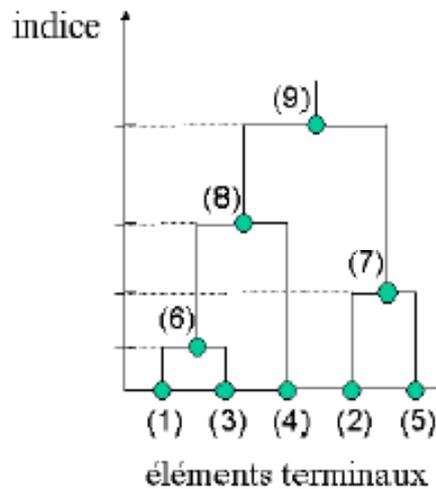


Figure: Coupure de l'arbre (dendrogramme).

- Les éléments terminaux de l'arbre (ou de la hiérarchie) sont les individus (ou variables selon ce qui est classé).
- Les nœuds de l'arbre correspondent aux regroupements de deux éléments appelés aîné et benjamin.

- L'agrégation repose sur les inégalités des distances entre elles. Nous pouvons obtenir le même classement en des couples d'éléments en classant ces couples par ordre croissant des distances. Un tel classement est appelé ordonnance.

Dans tous les cas, on décrit les groupes au moyen des variables de classification et, éventuellement, d'indicateurs supplémentaires. On obtient ainsi un profil (de comportement, d'attitude, sociodémographiques) des individus composant les différents groupes. Suivant le type de variable, on utilisera la moyenne, la médiane ou l'écart type.

Pour réaliser cette méthode, il est nécessaire de suivre les étapes suivantes :

- Le choix de la distance : dans cette analyse, on utilise la distance euclidienne.
- Le choix de l'algorithme de classification : on applique la classification ascendante hiérarchique CAH.
- Le choix d'agrégation : le critère à retenir est celui de Ward.
- Le choix du nombre de classes à retenir : coupure automatique de l'arbre selon le critère de Ward.
- Description des classes.

Avantages de la classification ascendante hiérarchique

La classification ascendante hiérarchique (CAH) est une méthode de classification qui présente les avantages suivants :

- On travaille à partir des dis-similarités entre les objets que l'on veut regrouper. On peut donc choisir un type de dis-similarité adapté au sujet étudié et à la nature des données.
- L'un des résultats est le dendrogramme, qui permet de visualiser le regroupement progressif des données. On peut alors se faire une idée d'un nombre adéquat de classes dans lesquelles les données peuvent être regroupées.

Application et Résultats

Nombre de Classes:

Le choix du nombre de classe pour le critère 'fraction d'insolation' a été réalisé à l'aide du « diagramme des niveaux », et du « dendrogramme » (arbre hiérarchique).

Diagramme Des Niveaux

Le diagramme des niveaux des nœuds donne des informations sur la structure des données. Lorsque des sauts importants sont observés, on a une agrégation de structures homogènes. Sur cette base que le logiciel crée la troncature automatique qui permet de déterminer le nombre de groupes homogènes.

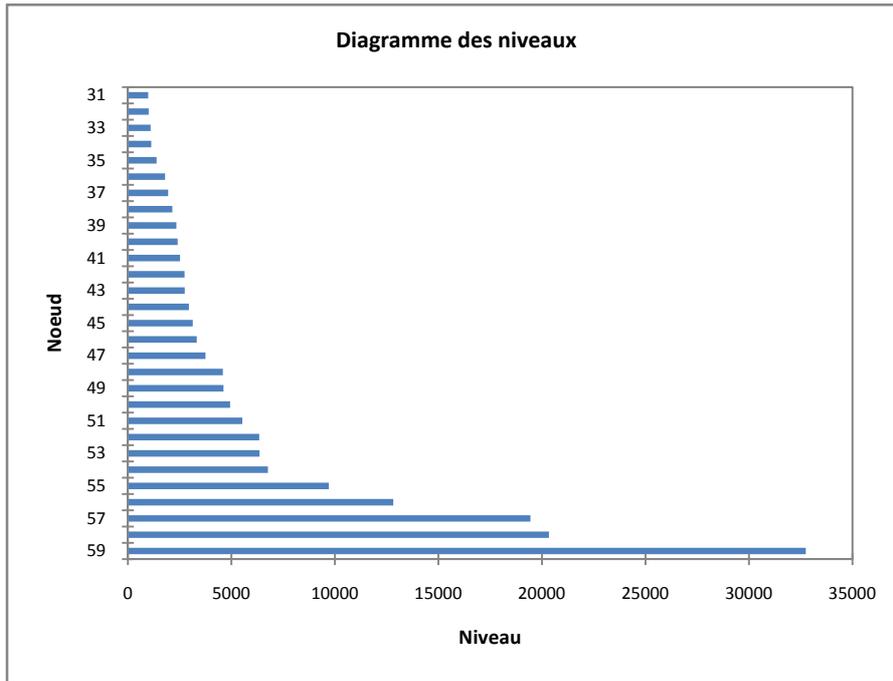


Figure : Diagramme des niveaux

Dendrogramme (Arbre Hiérarchique):

Le dendrogramme représente de manière claire la façon dont l'algorithme procède pour regrouper les variables puis les sous-groupes. Au final, l'algorithme a progressivement regroupé toutes les observations:

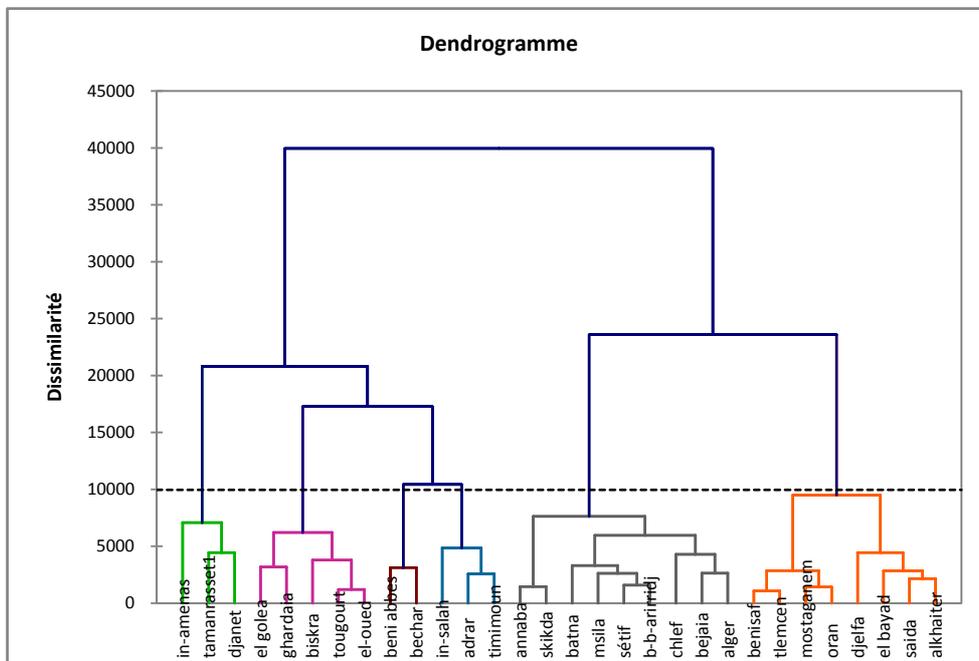


Figure : Dendrogramme de la Classification Ascendante Hiérarchique

La ligne en pointillé représente la troncature effectuée par le logiciel *XLSTAT* qui permet de visualiser que 06 classes homogènes ont été identifiées :

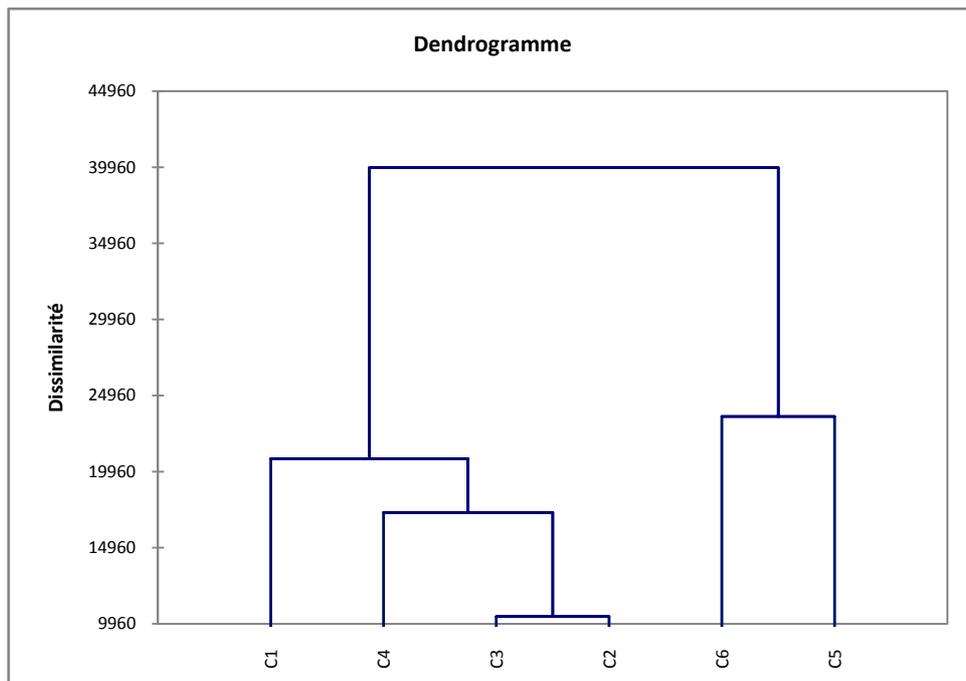


Figure : Dendrogramme du nombre de classes.

REFERENCES

1. Habiboulaye Amadou Boubacar ; « *Classification Dynamique De Données Non-Stationnaires Apprentissage Et Suivi De Classes Evolutives* » ; Thèse Doctorat, Université des Sciences et Technologies de Lille (USTL), France, 2006.
2. Abdullah Almaksour, Eric Anquetil ; « *Systèmes d'inférence floue auto-évolutifs : apprentissage incrémental pour la reconnaissance de gestes manuscrits* » ; Colloque International Francophone sur l'Ecrit et le Document (CIFED2010), Tunisie, 2010.
3. Eric ANQUETIL ; « *Modélisation et reconnaissance par la logique floue : application à la lecture automatique en-ligne de l'écriture manuscrite omni-scripteur* » ; Thèse Doctorat, Université des RENNES I, France, 1997.
4. Alexandre BARGETON ; « *Fusion multi sources pour l'interprétation de l'environnement routier* » ; Thèse Doctorat, Ecole Nationale Supérieure des Mines de Paris, France, 2009.
5. Vincent BARRA ; « *Fusion d'Images 3D du Cerveau : Etude de Modèles et Applications* » ; Thèse Doctorat, Université Blaise Pascal Clermont Ferrand II & Université D'Auvergne Clermont Ferrand I, France, 2000.
6. P Barralon, « *Classification et Fusion de Données Actimétriques pour la Télévigilance Médicale* », Thèse Doctorat, L'université Joseph Fourier, 2005.
7. I. BLOCH, H. MAITRE ; « *Fusion de données en traitement d'images : modèles d'information et décisions* » ; Traitement du signal, 1994, vol.11, n°6, pp 435 - 446.
8. Mohamed Saïd BOUGUELID ; « *Contribution à l'application de la reconnaissance des formes et la théorie des possibilités au diagnostic adaptatif et prédictif des systèmes dynamiques* » ; Thèse Doctorat ; Université de Reims Champagne-Ardenne, France, 2007.
9. Berrichi Fatima Zohra, Benyettou Mohamed ; « *L'Interprétation des Images Médicales Cérébrales : Segmentation par L'approche Split et Merge* » ; Proceedings of the 2nd Conférence Internationale sur l'Informatique et ses Applications (CIIA'09), Saida, Algeria, May, 2009.
10. Alexandre BLANSCHÉ ; « *Classification non supervisée avec pondération d'attributs par des méthodes évolutionnaires* » ; Thèse Doctorat, Université Louis Pasteur – Strasbourg I, France, 2006.
11. I Bloch, « *Information Combination Operators for Data Fusion: A Comparative Review with Classification* », IEEE Transactions on Systems, Man, and Cybernetics, 1:52-67, 1996.

12. M. BOUMAHDJ, J.P. DRON, S. RECHAK et O. COUSINARD ; « *Utilisation de l'arbre de décision pour la détection des défauts de roulements* » ; Revue Internationale sur l'Ingénierie des Risques Industriels (JI-IRI), Vol 2, No 1, 2009.
13. Jean-Patrick Baudry ; « *Sélection de Modèle pour la Classification Non Supervisée. Choix du Nombre de Classes* » ; Thèse Doctorat ; Université Paris-Sud XI, France, 2009.
14. Guido Boella, Guido Governatori, Joris Hulstijn ; « *Time and Defeasibility in FIPA ACL Semantics* ». International Conference on Web Intelligence and Intelligent Agent Technology (WI'08), University of Technology, Sydney, Australia November 27th, 2008.
15. Carlos Argel Iglesias , Mercedes Garijo , José Centeno-González , Juan R. Velasco; « *Analysis and design of multiagent systems using MAS-CommonKADS* ». The 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages (ATAL'97), Providence, Rhode Island, USA, July 24-26, 1997.
16. Nadjim Chelghoum, Karine Zeitouni, Azedine Boulmakoul ; « *Fouille de données spatiales par arbre de décision multi-thèmes* » ; Conférence : Journées Francophones d'Extraction et de Gestion des Connaissances (EGC 2002), Université Montpellier 2, Montpellier, France, 2002.
17. Alzenny GOMES DA SILVA ; « *Analyse des données évolutives : application aux données d'usage du Web* » ; Thèse Doctorat, Université Paris IX Dauphine, France, 2009.
18. Luc Descroix, Jean-Yves Loyer, Stéphanie Moriaud ; « *Unités de paysage pour l'hydrologie au Nord du Mexique* » ; Atelier international (Télédétection et ressources en eau), France, 1995.
19. Sebastien Destercke, Didier Dubois, Eric Chojnacki ; « *Possibilistic information fusion using maximal coherent subsets* » ; 5th Conference of the European Society for Fuzzy Logic and Technology (LFA'2007), France, Novembre, 2007.
20. Olivier Dupuis, « *Fusion entre les données ultrasonores et les images de radioscopie à haute résolution : Application au contrôle de cordon de soudure* » ; Thèse Doctorat ; Institut National des Sciences Appliquées de Lyon, France, 2000.
21. Viviane Torres da Silva, Ricardo Choren and Carlos J. P. de Lucena; « *A UML Based Approach for Modeling and Implementing Multi-Agent Systems* ». Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04), Columbia, University in New York City, USA, 19- 23 July, 2004.
22. Alexis Drogoul & J.-D. Zucker ; « *Methodological Issues for Designing Multi-Agent Systems with Machine Learning Techniques: Capitalizing Experiences from the RoboCup Challenge* ». Applied Artificial Intelligence, Volume 12, Issue 2 & 3 January 1998 , pages 127 – 147.
23. Alexis Drogoul, Anne Collinot ; « *Using the Cassiopeia method to design a Robot Soccer Team* ». Applied Artificial Intelligence, Volume 12, Issue 2 & 3, pages 127 – 147. January 1998.

24. Jacques Ferber ; «*Les systèmes multi-agents : vers une intelligence collective*», InterEditions, 1995.
25. Didier FRAGNE ; «*Proposition de l'architecture de l'agent gestionnaire du modèle de l'apprenant dans un système tuteur multi-agents en apprentissage de la lecture : contribution au projet AMICAL* ». Thèse Doctorat, Université Blaise Pascal Clermont-Ferrand II, France, 2009.
26. E Frenoux, «*Applications et validation de méthodologies de fusion de données en imagerie cérébrale* », Thèse Doctorat, Université d'Auvergne, 2004.
27. Abdessamed Réda GHOMARI ; «*Approche Méthodologique d'Acquisition de Connaissances Agrégées à base d'Agents cognitifs coopérants pour les systèmes d'aide à la décision stratégiques* ». Thèse Doctorat, Ecole nationale Supérieure en Informatique, Alger, Algérie, 2008.
28. Norbert GLASER; «*The CoMoMAS methodology and environment for multi-agent system development*». Lecture Notes in Computer Science, Volume 1286/1997, 1-16, 1997.
29. Veyis GUNES ; «*Reconnaissance des formes évolutives par combinaison, coopération et sélection de classifieurs* » ; Thèse Doctorat, Université de la Rochelle, France, 2001.
30. P. Hart et R. Duda. «*Pattern Classification and Scene Analysis* ». Wiley, New- York, 1973.
31. Lamis HAWARAH ; «*Une approche probabiliste pour le classement d'objets incomplètement connus dans un arbre de décision* » ; Thèse Doctorat, Université Joseph Fourier Grenoble_I, France, 2008.
32. Hebboul Amel ; «*Approches incrémentales pour l'apprentissage en reconnaissance de formes* » ; Mémoire de Magistère ; Université Mentouri – Constantine, Algérie, 2008.
33. Soufiane IDBRAIM ; «*Méthodes d'extraction de l'information spatiale et de classification en imagerie de télédétection : Applications à la cartographie thématique de la région d'Agadir (Maroc)* » ; Thèse Doctorat, Université de Toulouse III - Paul Sabatier, France, 2009.
34. Imed Jarras et Brahim Chaib-draa ; «*Aperçu sur les systèmes multiagents* ». Série scientifique, 2002s-67. Montréal, Juillet 2002.
35. E. Kaddoum, M.-P. Gleizes, J.-P. George, P. Glize, and G. Picard. «*Analyse des critères d'évaluation de systèmes multi-agents adaptatifs* ». In Z. Guessoum and S. Hassas, editors, Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09), Lyon, France, October 2009, pp. 123-124.
36. David Kinny and Michael Georgeff ; «*Modeling and design of Multi-Agent Systems* ». Proceedings of the third International Workshop on Agent Theories, Architectures, and Languages (ATAL'96). Budapest, Hungary, 12-13 August, 1996.

37. François KLEIN ; « *Contrôle d'un SMA réactif par modélisation et apprentissage de sa dynamique globale* ». Thèse Doctorat, Ecole doctorale IAEM Lorraine, France, 2009.
38. S. Labidi, W. Lejouad ; « *De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents* », Rapport de recherche, INRIA, N° 2004, 1993.
39. G Lecomte, « *Analyse d'images Radioscopiques et Fusion d'informations multimodales pour l'amélioration du contrôle de pièces de fonderie* », Thèse Doctorat, Institut National des Sciences Appliquées de Lyon, 2005.
40. Christophe Lurette ; « *Développement d'une technique neuronale auto-adaptative pour la classification dynamique de données évolutives. Application à la supervision d'une presse hydraulique* » ; Thèse Doctorat, Université des Sciences et Technologies de Lille (USTL), France, 2003.
41. A. Mefti et M.Y. Bouroubi, « *Estimation et Cartographie de la Composante Globale du Rayonnement Solaire* » ; Revue Energies Renouvelables Valorisation Tome2 ; 1999, pp. 219-224.
42. A. Meisels I. Razgon E. Kaplansky and R. Zivan. « *Comparing performance of Distributed Constraints Processing Algorithms* ». In Proc. AAMAS-2002 Workshop on Distributed Constraint Reasoning DCR, July 2002, pp. 86-93.
43. Nicolas MONMARCHE ; « *Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation* » ; Thèse Doctorat, Université de François Rabelais Tours, France, 2000.
44. Abdelouaheb MOUSSAOUI, Moussa SEMCHEDINE, Lyazid TOUMI; « *Un Système Multi-agents pour La Classification Coopérative D'images IRM Cérébrales* » ; Le premier séminaire national sur le langage naturel et l'intelligence artificielle (LANIA'2007), Chlef, Algérie 20-21 Novembre 2007.
45. Abdelouahab MOUSSAOUI, Hafida CHELLAKH; « *La Segmentation des IRM Cérébrales Pathologiques par une combinaison Floue Possibiliste Markovienne* » ; 1^{ières} Doctoriales, Sciences et Technologies de l' Information et de la Communication (STIC'09), Msila, Algérie, Décembre, 2009.
46. Nicolas MORIZET ; « *Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris* » ; Ecole Nationale Supérieure des Télécommunications de Paris , France, 2009.
47. Abdelouahab Moussaoui, Nassima Arbouche : « *Fouille de données biomédicales complexes. Application la segmentation d'images médicales* » ; 1^{ières} Doctoriales, Sciences et Technologies de l'Information et de la Communication (STIC'09), Msila, Algérie, Décembre, 2009.
48. Moussaoui Abdelouahab, Zouaoui Hakima ; « *Clustering par fusion floue de données appliqué à la segmentation d'images IRM cérébrales* » ; Conférence Internationale des Technologies de l'Information et de la Communication (CITIC'2009), Sétif, Algérie, Mai, 2009.

49. Bernhard Bauer, Jörg P. Müller, James Odell ; «*Agent UML: A Formalism for Specifying Multiagent Interaction*». Congrès Agent-Oriented Software Engineering (AOSE 2000), Limerick, Irlande (10/06/2000) , vol. 1957, pp. 91-103.
50. Michel OCCELLO ; «*Méthodologie et architectures pour la conception de systèmes multi-agents* ». Thèse Doctorat, Université Joseph Fourier de Grenoble, France, 2003.
51. Olivier PASSALACQUA ; «*Contrôle et implantation des systèmes répartis de fusion d'informations* » ; Thèse doctorat ; Université de Savoie ; France, 2009.
52. François Portet, René Quiniou, Marie-Odile Cordier, Guy Carrault ; «*Decision Tree Learning for On-Line Selection of Detection Algorithms in Electrocardiogram*» ; *16e conférence Reconnaissance des Formes et Intelligence Artificielle (RFIA'08) (2008)*, Amiens, France, Janvier, 2008.
53. Lilia REJEB ; «*Simulation multi-agents de modèles économiques Vers des systèmes multi-agents adaptatifs* ». Thèse Doctorat, Université de Reims Champagne-Ardennes, France, 2005.
54. Pierre-Michel Ricordel ; «*Programmation Orientée Multi-Agents : Développement et Déploiement de Systèmes Multi-Agents Voyelles* ». Thèse Doctorat, Institut national polytechnique de Grenoble, France, 2001.
55. Ronan Querrec ; «*Les systèmes multi agents pour les environnements Virtuels de Formation, application a la sécurité civile* ». Thèse Doctorat, Laboratoire d'Informatique Industrielle (LI2/ENIB), Université de Bretagne Occidentale, France, 2002.
56. R. Kohavi, «*Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid* », In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, pp. 202-207, 1996.
57. Amine SAIDANE, Herman AKDAG, Isis TRUCK ; «*Une Approche SMA de l'Agrégation et de la Coopération des Classifieurs* » ; 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT 2005), Tunisia, March, 2005.
58. Karim TABIA ; «*Développement de mécanismes de coopération entre algorithmes d'apprentissage automatique/classification dans un environnement incertain* » ; Mémoire de Magistère, Université Mouloud Mammeri de Tizi Ouzou, Algérie, 2005.
59. Thomas Laloë ; «*Sur Quelques Problèmes d'Apprentissage Supervisée et Non Supervisé* » ; Thèse Doctorat, Université de Montpellier II, France, 2009.
60. Pierre TIRILLY ; «*Evaluation des performances des Réseaux de Neurones Aléatoires et application à la Bio-informatique* » ; Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), 2006.
61. Andrew Tanenbaum ; «*Systèmes d'exploitation : systèmes centralisés et systèmes distribués* ». Interéditions, Paris, 1994.

62. Franco Zambonelli, Nicholas R. Jennings and Michael Wooldridge ; «*Developing Multiagent Systems: The Gaia Methodology* ». Journal ACM Transactions on Software Engineering and Methodology (TOSEM) Volume 12 Issue 3, July 2003.
63. Yannis Labrou and Tim Finin ; «*A Proposal for a new KQML Specification* ». Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, MD 21250, TR CS-97-03, February 1997
64. L Zadeh," *Fuzzy Sets, Information and Control*", 8:338-353, 1965.
65. L Zadeh , « *Fuzzy Sets as a Basis for Theory of Possibility* », International Journal of Fuzzy Sets and Systems, 1:3-28, 1978
66. I Bloch, « *Fusion d'informations en traitement du signal et des images* », 2003.
67. Laukkanen Mikko, « *Evaluation of JADE 1.2* », Helsinki, 16th February 2000.