ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre 3201

THESE

En vue de l'obtention du : ***DOCTORAT***

Centre de Recherche :
Structure de Recherche : **Laboratoire Conception et Systèmes**
Discipline : **Sciences de l'ingénieur**
Spécialité : **Informatique**

Présentée et soutenue le 25/04/2019 par :

Amina DIK

Nouveaux modèles flous d'apprentissage non supervisé des données multidimensionnelles

JURY

Mohamed JEDRA	PES, Faculté des sciences Rabat, Université Mohammed V	Président
Aziz ETTOUHAMI	PES, Faculté des sciences Rabat, Université Mohammed V	Directeur de Thèse
Noureddine ZAHID	PES, Faculté des sciences Rabat, Université Mohammed V	Rapporteur
Khalid JEBARI HASSANI	PH, Faculté des Sciences et Technologies de Tanger, Université Abdelmalek Essaâdi	Rapporteur
Abderrahim EL QADI	PES, Ecole Supérieure de Technologie – Salé, Université Mohammed V	Examinateur

2018- 2019

Faculté des Sciences, 4 Avenue Ibn Battouta B.P. 1014 RP, Rabat- Maroc
Tel +212 (05) 37 77 18 34/35/38, Fax: +212 (05) 37 77 42 61, http://www.fsr.ac.ma.

Dédicaces

A

Ma famille,

Mes amis(es),

Mes collègues,

Mes professeurs.

AVANT PROPOS

Les travaux présentés dans cette thèse ont été effectués au Laboratoire Conception et Systèmes (Signaux, Electronique et Informatique) de la Faculté des Sciences de Rabat sous la direction du professeur Aziz ETTOUHAMI.

Ces remerciements s'adressent tout naturellement à la personne qui m'a guidé tout au long de ce travail, m'assurant d'un soutien permanent, m'imposant la rigueur scientifique nécessaire à ma tâche, prodiguant son temps et ses conseils: Monsieur Aziz ETTOUHAMI, mon encadrant, professeur à la faculté des Sciences de Rabat et directeur du laboratoire LCS, qui m'a encadré et dirigé tout au long de ce travail. Sa disponibilité et son encadrement de proximité m'a permis de mieux cerner la problématique et essayer de mieux y répondre.

Je tiens à exprimer ma profonde gratitude à monsieur Abdelaziz BOUROUMI, professeur à la faculté des sciences Ben'Msik de Casablanca, pour sa grande rigueur scientifique et l'aide qu'il m'a constamment octroyée, qu'il trouve, en ces mots, le témoignage de mes sincères remerciements.

J'exprime ma profonde gratitude à monsieur Mohamed JEDRA, professeur à la faculté des Sciences de Rabat, pour ses remarques pertinentes et l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

J'adresse mes sincères remerciements à monsieur Noureddine ZAHID, professeur à la faculté des Sciences de Rabat, pour l'intérêt qu'il a porté à ce travail en acceptant d'en être rapporteur et de participer au jury de cette thèse. Je le remercie vivement pour ses remarques pertinentes, ses nombreux conseils.

Je souhaite également exprimer ma reconnaissance à monsieur Khalid JEBARI HASSANI, professeur à la Faculté des Sciences et Technologies de Tanger, pour sa précieuse aide et son soutien illimité. Je le remercie vivement pour ses remarques et conseils et pour l'intérêt qu'il a manifesté à ce travail en acceptant d'en être rapporteur.

Je remercie également monsieur Abderrahim EL QADI, professeur à l'Ecole Supérieure de Technologie à Salé, d'avoir accepté d'être membre de jury et d'examiner ce travail.

Liste des publications

Publications dans des revues spécialisées :

1. K. Jebari, A. El moujahid, A. Dik, A. Bouroumi, A. Ettouhami, « Unsupervised fuzzy tournament selection », *in Applied Mathematical Sciences*, Vol .5, no.58, pp.2863 – 2881, 2011.
2. A. Dik, El moujahid, A. Bouroumi, A. Ettouhami, « Weighted distances for fuzzy clustering », *in Applied Mathematical Sciences*, Vol. 8, no.4, pp.147- 156, 2014.
3. A. Dik, K. Jebari, A. Bouroumi, A. Ettouhami, « Fractional metrics for fuzzy c-Means », *International Journal of Computer and Information Technology*, Vol. 03, no. 06, pp.1490-1495, 2014.
4. A. Dik, K. Jebari, A. Bouroumi, A. Ettouhami, « A new fuzzy clustering by outliers », *Journal of Engineering and Applied Sciences*, Vol. 9 (10-12), pp.372-377, 2014.
5. K. Jebari, A. Dik, A. Ettouhami, « Combined crossover operator », *Research Journal of Applied Sciences*, 10(3), 75-79, 2015.
6. A. Dik, A. El moujahid, K. Jebari, A. Ettouhami, « A new dynamic algorithm for unsupervised learning », *International journal of innovative computing, information & control: IJICIC*, Vol.11, no.5, pp. 1325-1339, 2015.
7. A. Dik, K. Jebari, A. Ettouhami, « An improved robust fuzzy algorithm for unsupervised learning », *Journal of Intelligent Systems*. DOI: https://doi.org/10.1515/jisys-2018-0030. 2018.

Communications

8. A. Dik, A. El moujahid, A. Bouroumi, A. Ettouhami, « Etude comparative de mesures de similarités », RNCJP6, Faculté des sciences Ben'Msik, Casablanca, 2009.
9. A. Dik, H.Benrachid, A. Bouroumi, A. Ettouhami, « Apprentissage flou non supervisé et règles de décision », 1ère Journée de l'Informatique Décisionnelle (JID'10), El Jadida, Mars 2010.

10. H. Benrachid, A. Dik, A. Bouroumi, A. Ettouhami, « Etude d'un algorithme d'apprentissage flou non supervisé AFNS », 1ère Journée de l'Informatique Décisionnelle (JID'10), Faculté des Sciences, Université Chouaib Doukkali, El Jadida, Mars 2010.

11. A. Dik, A. Bouroumi, A. Ettouhami, « Improving fuzzy clustering by r-metric », MNOTSI 2012, ENSA – Kenitra, 2012.

RESUME

Nos travaux de recherche ont pour objectif, d'une part, de réduire l'impact du chevauchement des classes de données, lorsque les limites entre les classes d'une partition sont fortement ambiguës et males définies, et où l'incertitude et la difficulté à prendre une décision sont grandes, et d'autre part, à identifier les valeurs aberrantes qui peuvent déséquilibrer l'apprentissage en se voyant accorder une importance plus grande qu'elles n'ont. Ainsi, on a proposé de nouveaux algorithmes d'apprentissage flou non supervisé à partir de données non étiquetées et en présence d'éventuelles valeurs aberrantes. On s'est intéressé ainsi aux points suivants: 1) les mesures de similarités entre les données et leur rôle crucial pour former les classes, ainsi qu'à la caractérisation de ces classes par des prototypes, 2) la quantification de l'imprécision et la tolérance de l'incertitude dans le cas du chevauchement aigu des classes où il s'avère difficile d'émettre une décision dans un environnement imprécis sans avoir suffisamment d'informations, 3) l'impact des valeurs aberrantes sur l'apprentissage, et les techniques proposées dans la littérature pour pouvoir effectuer un apprentissage des données en présence des valeurs aberrantes. Les expériences menées sur des données du monde réel montrent l'efficacité des algorithmes proposés pour l'apprentissage des données et la gestion de l'incertitude.

Mots-clefs : Apprentissage flou non supervisé, Mesure de Similarité, Valeurs aberrantes, Degré de proximité.

Liste des abréviations des algorithmes

FCM	Fuzzy c-means
PCM	c-moyennes possibilistes
ISODATA	Iterative Self-Organizing Data Analysis Technique
AFNS	Algorithme d'apprentissage flou non supervisé
UFL	Unsupervised fuzzy learning
IUFL	Improved Unsupervised fuzzy learning
POFCM	Possible outliers FCM
RDUFL	Robust dynamic Unsupervised fuzzy learning

Table des matières

INTRODUCTION GENERALE

Plusieurs méthodes ont été développées par l'intelligence artificielle (I.A) pour simuler certains aspects de l'intelligence humaine. Ces méthodes essayent de faire en sorte qu'un ordinateur puisse élaborer un raisonnement tel que le ferait un être humain. Leur principe consiste ainsi à s'approcher de la démarche humaine dans le sens que les variables traitées ne sont pas des variables logiques, mais des variables linguistiques proches du langage humain. Ces variables linguistiques telles que très, trop, assez, insuffisant, traduisent l'incertitude rencontrée avec les caractéristiques de certains objets que l'être humain est interpelé à reconnaitre pour prendre une décision. Or, la reconnaissance est sous-jacente à l'apprentissage. L'apprentissage est donc une phase primordiale du processus de la reconnaissance pendant lequel on cherche des règles de décision qui permettent de reconnaitre un objet en fonction de sa description. D'où l'utilité de concevoir des techniques d'aide à la décision destinées à complémenter l'homme sur le plan de l'apprentissage.

Schématiquement, l'apprentissage peut être présenté sous forme d'un algorithme qui reçoit en entrée des éléments d'une base de données et produit en sortie des connaissances sur la structure des données analysées. Il consiste ainsi à exploiter les données d'une base d'apprentissage dans le but de faciliter la prise de décision. Il peut être supervisé, semi supervisé, ou non supervisé.

L'apprentissage supervisé œuvre à établir des règles à partir d'une base d'apprentissage contenant des exemples étiquetés. Formellement parlant, la base de données est représentée par un ensemble de couples {(A, B)}, où A est une des entrées et B la sortie correspondante. L'apprentissage concerne alors la prédiction de la sortie pour toute nouvelle entrée. Autrement dit, il se fonde sur les éléments déjà classés pour en classer de nouveaux.

L'apprentissage non supervisé ou clustering des données est un processus qui consiste à regrouper un ensemble de n points de données en sous-groupes appelés classes. L'idée de regroupement des données est naturelle et émane de la pensée humaine: l'homme a habituellement tendance à regrouper un grand nombre de données dans un petit nombre de groupes ou de catégories afin de faciliter davantage leur analyse. Or, la recherche de ces groupes n'est pas une tâche simple pour les humains quand les données sont de grande

dimension. Plusieurs techniques ont été proposées pour de telles situations. Leur principe de base est de former à partir des données non étiquetées des groupes qui soient les plus homogènes et naturels possible. “Homogène” et “naturel” signifient que les groupes obtenus doivent contenir des individus les plus similaires possible, tandis que des individus de groupes différents doivent être les plus dissimilaires possible. Pour quantifier ce degré de ressemblance entre les paires de données, plusieurs mesures de similarité ont été proposées dans la littérature au cours des dernières décennies. Toutefois, aucune mesure proposée n’a été universellement adaptée aux différents domaines d'application ou aux différentes structures de données.

L’apprentissage semi supervisé consiste à traiter des données non étiquetées en s'aidant des données étiquetées. Ainsi, il se situe entre l’apprentissage supervisé et l’apprentissage non supervisé.

Plusieurs méthodes, concernant l’apprentissage, ont été proposées dans la littérature. Toutefois, il n’existe aucune méthode d’apprentissage qui soit déterministe pour un problème donné [Roux, 1985]. De plus, même si l’apprentissage est largement appliqué dans les problèmes du monde réel, le choix de la méthode d’apprentissage est l’un des problèmes difficiles et pour lequel aucune solution universelle n’est encore trouvée. A cette difficulté du choix de l’algorithme d’apprentissage, s’ajoutent deux problèmes pouvant éventuellement survenir. Le premier concerne le chevauchement des classes de données, où l’incertitude et la difficulté à prendre une décision sont grandes, ce qui réduit parfois l'efficacité de la méthode d’apprentissage. Le second problème concerne la présence des valeurs aberrantes dans les données. Ces dernières déséquilibrent l’analyse en se voyant accorder une importance plus grande qu’elles n’ont. Ainsi, des méthodes d’apprentissage non supervisé sensibles à ces valeurs aberrantes peuvent générer des partitions erronées.

Les travaux de cette thèse prennent place dans le contexte de l’apprentissage non supervisé. L’objectif principal de ces travaux est de contribuer à la résolution de ce problème, en concevant deux algorithmes d’apprentissage non supervisé basés essentiellement sur une mesure de similarité et un seuil correspondant. Ces algorithmes sont deux versions améliorées d’une approche particulière d’apprentissage non supervisé appelée « apprentissage flou non supervisé AFNS ». L’amélioration porte sur deux volets:

- Le premier a trait au cas où les informations dont nous disposons ne sont pas précises lors du chevauchement aigu des classes. C'est le cas où il s'avère difficile d'émettre une décision sur l'appartenance d'un objet à une classe particulière si cet objet ne porte pas suffisamment d'informations. Pour cela, une nouvelle règle d'apprentissage a été définie. Elle consiste à n'utiliser l'information apportée par un objet exploré que si cette information est pertinente pour prendre des décisions. En effet, elle permet soit: la création d'une nouvelle classe autour d'un objet dissemblable aux objets examinés précédemment, la mise à jour des prototypes des classes existants, ou le report de l'examen de cet objet jusqu'à ce que l'une des deux décisions antérieures puisse être faite sans confusion. Elle permet ainsi à l'algorithme d'évaluer les informations incomplètes, et d'admettre un nouveau cas du doute en dehors de l'ensemble {créer, ne pas créer}. Cette règle d'apprentissage confère à l'algorithme un aspect dynamique en ce qu'elle dépend du nombre de classes, qui varie au cours du processus d'apprentissage. Un seuil dynamique est utilisé pour décider s'il faut laisser un objet n'apportant pas assez d'information à un réexamen ultérieur.

- L'autre volet a trait au cas des données aberrantes qui peuvent éventuellement fausser l'apprentissage. En effet, ces valeurs aberrantes déséquilibrent l'apprentissage en se voyant accorder une importance plus grande qu'elles n'ont. Ainsi, des algorithmes d'apprentissage non supervisé peuvent générer des partitions erronées à cause de ces valeurs aberrantes. Pour pouvoir effectuer simultanément un apprentissage des données et isoler ces valeurs aberrantes ou réduire leur impact, nous avons proposé dans ce mémoire une approche basée sur trois étapes. La première étape est une méthode de prétraitement qui identifie les points qui peuvent être considérés comme des "possibles" valeurs aberrantes en utilisant le concept de degré de proximité. La seconde étape est un algorithme d'apprentissage flou non supervisé qui détecte les classes existantes formées par les données mais sans considérer les possibles valeurs aberrantes. La création des classes se fait selon un seuil dynamique recalculé automatiquement à chaque itération de l'algorithme. Ce seuil se base sur la similarité entre les prototypes des classes détectées et mis à jour à l'exploration de tout nouvel objet. Quant à la dernière étape, elle consiste à traiter les possibles valeurs aberrantes isolées lors de la première phase, et ce en se basant sur la similarité de chaque valeur aberrante à l'objet le plus loin de la classe correspondant au plus proche prototype à cette valeur aberrante.

Le manuscrit est ainsi séparé en quatre chapitres organisés comme suit :

Dans le premier chapitre de ce mémoire, nous présenterons les notions et les concepts de base sur lesquels s'appuiera la suite de cette thèse, et soulignerons la diversité qui existe parmi les différentes méthodes d'apprentissage non supervisé. Ainsi en premier lieu, nous présenterons les définitions, la représentation mathématique et les types d'apprentissage non supervisé. Ensuite, nous mènerons une étude de synthèse des principaux algorithmes d'apprentissage non supervisé, en présentant leurs principaux avantages et inconvénients. Nous verrons essentiellement les algorithmes connus sous le nom de c-moyennes floues, et de nombreuses variantes qui en dérivent, ainsi que l'algorithme AFNS objet de nos futures améliorations.

Dans le second chapitre, nous introduisons le concept de similarité et son utilité à représenter la similitude et la proximité d'une paire d'objets. Nous examinerons ainsi les mesures de similarités basées sur les distances usuelles, en effectuant une étude comparative entre elles [Dik, 2012]. Nous présenterons également les distances fractionnaires en essayant de déterminer le lien qui existe entre les données et la valeur du coefficient « *r* » de la distance fractionnaire [Dik, 2014 a], ainsi que la définition de nouvelles distances normalisées en montrant leur utilité pour améliorer les résultats de l'apprentissage [Dik, 2014 b].

Dans le troisième chapitre, nous présenterons le premier algorithme d'apprentissage flou intitulé « *Improved Unsupervised Fuzzy Learning*(IUFL) *algorithm* » qui permet de pallier le problème de l'incertitude de décision dû au chevauchement des classes non séparées. Les expériences menées sur des données et des images médicales montrent l'efficacité de cet algorithme à mieux mener l'apprentissage et gérer l'incertitude rencontrée [Dik, 2015].

Dans le quatrième chapitre, nous présenterons le second algorithme d'apprentissage flou développé, intitulé« *Robust Dynamic and Unsupervised Fuzzy Learning* (RDUFL) *algorithm*». Ce dernier constitue une version de l'AFNS destinée aux données affectées par des valeurs aberrantes. Les résultats obtenus montrent que cet algorithme permet de détecter également ces valeurs et de minimiser leurs effets. Ces résultats montrent aussi que l'apprentissage est amélioré même en absence de valeurs aberrantes [Dik, 2018].

A la fin de ce manuscrit, nous présentons les conclusions de la thèse et un résumé des futures lignes de recherche.

CHAPITRE 1

ELEMENTS DE BASE SUR L'APPRENTISSAGE FLOU NON SUPERVISE

Introduction

Le problème d'apprentissage consiste à exploiter les éléments d'une base de données pour chercher des règles de décision qui permettent de reconnaitre un élément en fonction de sa description. Ce problème se décline principalement en deux variantes: l'apprentissage supervisé (ou assisté) et l'apprentissage non supervisé.

L'objectif de ce chapitre est d'introduire un ensemble de notions sur l'apprentissage non supervisé, nécessaires à la compréhension de la thèse. Nous commençons par rappeler quelques concepts et définitions, avant de présenter quelques approches utilisées en apprentissage non supervisé, tout en soulevant leurs principales limites. Nous présentons également les règles de décision les plus utilisées en apprentissage non supervisé. Nous terminons ce chapitre sur la question de l'évaluation d'un apprentissage à l'aide des critères de validité.

1.1 Apprentissage non supervisé

Les méthodes d'apprentissage non supervisé permettent de regrouper des objets en classes plus homogènes sans aucune information préalable sur la structure de ces objets. Les objets regroupés sont similaires mais se distinguent des objets des autres classes par leurs caractéristiques. Généralement, une tâche d'apprentissage non supervisé nécessite les étapes suivantes [Laetitia, 2003]:

- Représentation des données ;
- définition d'une mesure de similarité ;
- définition de la méthode d'apprentissage non supervisé (flou ou classique) ;
- abstraction des données, qui se traduit généralement en terme de prototype ou centre ;
- évaluation des résultats ou validation.

Dans ce paragraphe, nous présentons ces fondements, les traits communs aux algorithmes d'apprentissage non supervisé ainsi que les règles de décision sur lesquelles se basent les algorithmes d'apprentissage non supervisé les plus usuels.

1.1.1 Concepts et notations

Représentation des objets et notion de prototype

L'apprentissage non supervisé peut être appliqué à des données numériques, catégoriques ou à un ensemble des deux. Dans la réalité, les données sont généralement des observations de certains processus physiques où chaque observation x_i est constituée de p variables mesurées. Ainsi, une observation est représentée par le vecteur $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$. En terminologie scientifique, une observation est également appelée objet, instance, individu, etc. La représentation matricielle de l'ensemble $X=\{x_1, x_2, \ldots, x_n\} \subset R^p$ est donnée par la matrice suivante :

$$X = \begin{bmatrix} x_{11} & x_{12} & & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ & & & \\ & & & \\ x_{n1} & x_{2n} & & x_{np} \end{bmatrix} \tag{1.1}$$

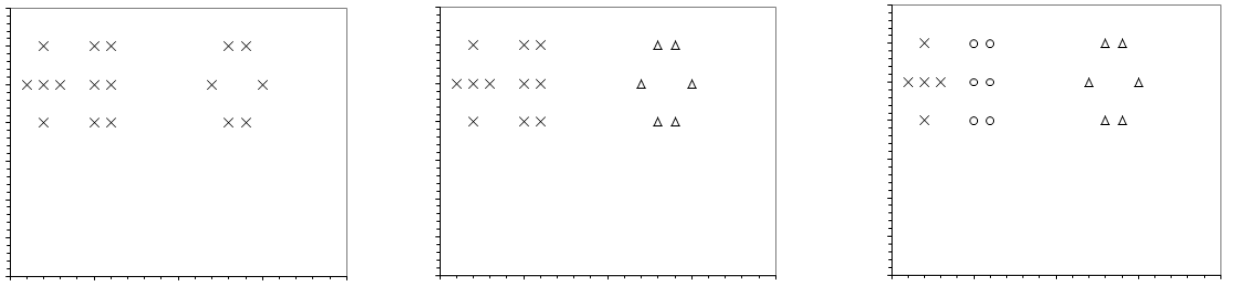
Où chaque ligne représente un objet, et chaque colonne représente un attribut qui décrit ces objets.

Schématiquement parlant, l'apprentissage non supervisé organise un ensemble de données X en un ensemble fini de classes $C= \{c_1, c_2, \ldots c_k\}$. En conséquence, le nombre de partitions possibles est très grand et la recherche de la meilleure partition devient difficile et s'accroit avec le nombre d'objets considérés. En effet, le nombre total de partitions possibles de l'ensemble X en k classes est égal à :

$$C_n^k = \frac{n!}{k!\,(n-k)!} \tag{1.2}$$

Par ailleurs, si le nombre de classes est inconnu, le nombre de partitions possibles augmente d'avantage [Khodja, 1997]. D'où l'utilité de proposer une solution qui parait intéressante et qualifiée de profitable dans un temps raisonnable, au lieu d'en chercher la meilleure en parcourant toutes les possibilités données [Jolloi, 2003].

La figure 1.1 illustre un exemple de différentes classes qui peuvent être construites à partir d'un même jeu de données. La figure (a) représente les données initiales avant l'apprentissage, la figure (b) représente le résultat de l'apprentissage non supervisé avec deux classes, alors que la figure (c) représente l'apprentissage non supervisé des même données avec trois classes.



(a) Données initiales (b) Apprentissage avec 2 classes (c) Apprentissage avec 3 classes

Figure 1.1 Exemple de différentes classes construites à partir d'un même jeu de données.

D'autre part, les principales méthodes de l'apprentissage non supervisé, utilisant une distance pour découvrir les classes existantes, sont les méthodes à base de prototypes. Ces dernières assurent que chaque classe est représentée par un objet qu'on appelle également prototype de classe. Un prototype est considéré comme un objet dont l'appartenance à la classe est garantie et qui est plus représentatif que tous les autres objets de cette classe. D'ailleurs la détermination de l'appartenance d'un objet à une classe se base sur le degré de sa similarité avec le prototype de ladite classe. Les prototypes ne sont généralement pas connus à l'avance et sont déterminés au fur et à mesure du partitionnement des données.

Traits communs aux algorithmes d'apprentissage non supervisé

Les méthodes d'apprentissage non supervisé nécessitent généralement un ensemble de prérequis relatifs aux données, et des paramètres à fournir avant de procéder au partitionnement de ces données.

Prétraitement des données

L'apprentissage non supervisé nécessite généralement le passage par une étape de prétraitement des données. Cette dernière désigne l'ensemble des opérations qui consistent à traiter et améliorer les données avant d'appliquer les algorithmes d'apprentissage non supervisé. Elle se décompose en deux phases principales.

La phase du nettoyage des données: elle consiste à filtrer les données inutiles pour augmenter la qualité d'apprentissage. Elle englobe l'isolement ou la suppression des valeurs aberrantes, et la correction des attributs manquants.

La phase de transformation des données: elle consiste à structurer les données selon un formalisme souhaité pour pouvoir leur appliquer les algorithmes d'apprentissage flou. La transformation des données peut englober plusieurs opérations possibles, notamment les opérations les plus usuelles suivantes:

- La normalisation des données dans l'intervalle [0,1] ou la pondération par des coefficients typiques.
- Le lissage des données qui considère les échantillons très proches comme étant le même échantillon.
- La réduction de dimension des données en détectant celles qui sont corrélées.
- La suppression des attributs dont l'importance dans la caractérisation des données est faible.

Problème du nombre de classe

Définir le nombre de classes est un des problèmes des méthodes d'apprentissage non supervisé basées sur la distance [Duda, 1973] [Forestier, 2010]. En effet, ces méthodes exigent généralement la connaissance a priori de ce nombre. Or ceci risque d'imposer une structure non réelle aux données considérées [Ball, 1966] [Duda, 2000]. Pour remédier à ce problème, plusieurs approches ont été proposées. La plus simple de ces approches se caractérise par le fait que l'algorithme est exécuté plusieurs fois avec un nombre de classes différent. Les résultats sont ensuite analysés et comparés. Ceci rend l'automatisation du processus difficile. Une autre approche [Figueiredo et Jain, 2000] [Hansen et Yu, 1998]

consiste à commencer avec un nombre de classes élevé, et de fusionner itérativement deux classes selon un critère. Toutefois, ces approches sont coûteuses et nécessitent un temps d'exécution énorme [Forestier, 2010]. En plus, elles exigent plusieurs paramètres de contrôle qui ne sont pas triviaux.

Problème d'initialisation

L'impact de l'initialisation des prototypes est significatif dans la plupart des algorithmes de partitionnement [Sun, 2004]. En effet, à chaque initialisation correspond une solution différente (optimum local) qui peut dans certains cas être très loin de la solution optimale (optimum global). A cet effet, plusieurs méthodes ont été proposées pour résoudre ce problème [Damodar, 2012]. La plus simple de ces méthodes est celle qui consiste à choisir aléatoirement ces prototypes. Une autre solution naïve est celle qui consiste à lancer l'algorithme plusieurs fois avec différentes initialisations et retenir la meilleure partition, ce qui est coûteux. On trouve également dans la littérature une autre méthode qui consiste à les choisir initialement placés le plus loin possible les uns des autres [Benrabah, 2005]. Mais, les plus usuelles méthodes proposées [Abouelala, 2002] sont les suivantes:

Initialisation de Forgy [Forgy, 1965]: Cette initialisation consiste à initialiser la matrice U^0 par des zéros et choisir de façon aléatoire c objets comme centres. Elle attribue ensuite chaque point à l'un des c classes de manière uniforme et aléatoire.
Cette approche affirme que si on choisit des objets aléatoires, on est plus susceptible de choisir un objet situé près d'un centre de classe. Cependant, rien ne garantit qu'on ne choisisse pas deux objets près du centre de la même classe, ou qu'on ne choisisse pas un point isolé.

Initialisation de MacQueen [MacQueen, 1967]: Elle consiste à choisir aléatoirement k centres, et affecter les objets non choisis à la classe représentée par le centre le plus proche. A chaque affectation on doit recalculer le centre de chaque classe. De ce fait, cette méthode diffère de celle de Forgy dans le sens qu'au lieu d'affecter tous les objets restants à l'un des k classes de départ les plus proches et d'itérer l'algorithme K-means jusqu'à sa convergence, nous affectons un objet à la fois, dans l'ordre dans lequel ils apparaissent, au centre de classe la plus proche.

Initialisation de Tou et Gonzales [Tou, 1974] suggèrent la méthode dite « *Simple Cluster-Seeking »* (SCS). Elle consiste à initialiser la première classe avec le premier objet, calculer la distance entre cet objet et l'objet suivant. Si elle est supérieure à un seuil choisi, on sélectionne cet objet en tant que deuxième centre, sinon on passe à l'objet suivant et on répète l'opération. Une fois que le deuxième centre est choisi, on passe à l'objet suivant et on calcule la distance qui le sépare des deux centres déjà choisis. Si ces deux distances sont supérieures au seuil, on sélectionne cet objet comme troisième centre. On continue jusqu'à ce que *k* centres soient choisis.

Cette méthode dépend de l'ordre des objets et nécessite de choisir une valeur de seuil, ce qui affecte les résultats.

Heuristique de Windham [Windham, 1981]: la matrice d'appartenance *U* est initialisée par :

$$U^0 = \begin{cases} 0{,}707 + \dfrac{0{,}293}{c} & si \quad i = k \\ \dfrac{0{,}293}{c} & si \quad i \neq k \end{cases} \tag{1.3}$$

Où *c* représente le nombre de classes.

Initialisation de Kaufman [Kaufman, 1990]: Le premier représentant est l'objet le plus proche du centre de gravité globale de l'ensemble de données, les autres représentants sont sélectionnés en choisissant l'objet ayant pour voisins le plus grand nombre d'objets non sélectionnés.

L'inconvénient évident de cette technique est les calculs considérables impliqués dans le choix de chaque centre.

Initialisation de Bezdek [Pal, 1995]: les représentants des classes sont initialisés par :

$$v_{ij} = m_j + (i-1)\left(\frac{M_j - m_j}{c-1}\right) \quad 1 \le i \le c \quad 1 \le j \le p \tag{1.4}$$

Où :

$m_j = \underset{1 \le j \le p}{Min}(x_{kj})$ Le minimum du k^{ème} paramètre

$M_j = \underset{1 \le j \le p}{Max}(x_{kj})$ Le maximum du k^{ème} paramètre

Initialisation de Bradley et Fayyad [Bradley, 1998] : C'est une technique permettant d'initialiser l'algorithme K-means. Les auteurs commencent par diviser de manière aléatoire les données en 10 sous-ensembles. Ils effectuent ensuite un partitionnement par K-means de chacun des 10 sous-ensembles, en commençant par des centres initiaux choisis selon la méthode de Forgy. Le résultat des 10 partitionnements est 10**K* objets centraux. Ces 10**K* objets sont alors eux-mêmes partitionnés par l'algorithme K-means qui est exécuté 10 fois. Lors de chacune de ces exécutions, l'algorithme K-means est initialisé par les *K* centres finaux de l'exécution qui la précède. Les emplacements des centres *K* résultants de la 10^{éme} exécution sont utilisés pour initialiser l'algorithme K-means pour l'ensemble de données.

D'autres méthodes ont été proposées récemment en littérature [Eltibi, 2011]. Une de ces méthodes consiste à effectuer un partitionnement initial avec une méthode n'exigeant pas l'initialisation des prototypes, et d'exécuter ensuite le second partitionnement en initialisant les prototypes découverts lors du premier partitionnement [Bouroumi, 2000].

1.1.2 Types d'apprentissage non supervisé

Les données peuvent constituer des classes de différentes tailles, densités et formes géométriques. En plus, les classes peuvent se chevaucher les unes aux autres (apprentissage non supervisé flou) ou être bien séparées (apprentissage non supervisé classique).

Apprentissage non supervisé classique

L'apprentissage non supervisé classique ou dur (Hard clustering) consiste à diviser l'ensemble des points de données en classes telles que chaque classe doit contenir au moins un objet et chaque objet doit appartenir exclusivement à une classe unique. Ceci suppose que les limites entre les classes sont bien définies. Ainsi, le partitionnement classique d'une base d'apprentissage $X=\{x_1, x_2,...,x_n\} \subset \Re^p$ en c classes est défini par une famille de sous-ensembles $\{A_k / 1 \le k \le c\}$ avec les propriétés suivantes:

$$\bigcup_{k=1}^{c} A_k = X \qquad (1.5)$$

$$A_i \cap A_j = \phi, \quad 1 \le i \ne j \le c \tag{1.6}$$

$$\phi \subset A_i \subset Z, \quad 1 \le i \le c \tag{1.7}$$

L'équation (1.5) signifie que l'union des sous-ensembles A_i forme l'ensemble considéré X et contient ainsi toutes les données. L'équation (1.6) signifie que les A_i sont disjoints, alors que l'équation (1.7) signifie qu'aucun d'entre ces A_i n'est vide ni contient toutes les données de X.

En termes de fonctions d'appartenance, ceci peut être représenté par une (c x n) matrice de partition $U= [u_{ik}]$, où les éléments doivent satisfaire les conditions suivantes:

$$u_{ik} \in \{0, 1\} \quad 1 \le k \le c \quad 1 \le i \le n \tag{1.8}$$

$$0 < \sum_{i=1}^{n} u_{ik} < n \qquad 1 \le k \le c \tag{1.9}$$

L'espace des matrices de partition de l'ensemble X, appelé espace du partitionnement dur (Bezdek, 1981), est ainsi défini par :

$$M_{hc} = \left\{ U \in \Re^{cxn} / u_{ik} \in \{0,1\} \; \forall i,k \; ; \; 0 < \sum_{i=1}^{n} u_{ik} < n \; , \forall k \right\}$$

L'apprentissage non supervisé flou

Dans la réalité, les limites entre les classes sont souvent incertaines et ambiguës. L'incertitude s'exprime par le fait qu'un objet possède des caractéristiques qui permettent de l'assigner à plus d'une classe. La modélisation s'effectue alors en considérant des frontières graduelles au lieu des frontières nettes entre les classes. Ainsi, en apprentissage non supervisé flou, un point n'appartient pas complètement à une seule classe, mais possède un degré d'appartenance à toutes les classes existantes. Le degré d'appartenance se situe dans l'intervalle [0, 1] et les classes obtenues ne sont pas forcément disjointes. L'apprentissage non supervisé flou est ainsi une généralisation de l'apprentissage non supervisé classique qui permet de traiter le chevauchement éventuel des classes [Bezdek 1981]. Cette généralisation a été introduite par le concept de degré d'appartenance u_{ik} qui est interprété comme le degré d'appartenance du point i à la $k^{éme}$ classe ($1 \le k \le c, \; 1 \le i \le n$) [Bezdek 1981, Bezdek 1984].

Mathématiquement parlant, le partitionnement d'une base de données X en c classes floues est défini par c ensemble flous $E_1,\ldots, E_c$ et une fonction d'appartenance [Bezdek 1981] qui prend des valeurs de l'intervalle [0, 1] tel que:

$$E_k = \{\mu_k(x_i) / \ x_i \in X, 1 \le i \le n\} \tag{1.10}$$

$$\forall i,k \quad \mu_k : \begin{cases} X \to [0,1] \\ x_i \to \mu_k(x_i) = u_{ik} \end{cases} \tag{1.11}$$

Ainsi, une (c x n) matrice d'appartenance floue $U= [u_{ik}]$ peut être utilisée pour représenter la partition qui résulte de l'analyse de X. La $k^{éme}$ ligne de la matrice contient des valeurs de la $k^{éme}$ fonction d'appartenance μ_k du sous ensemble E_k. Les éléments u_{ik} satisfont les conditions suivantes:

$$0 \le u_{ik} \le 1 \quad 1 \le k \le c \quad 1 \le i \le n \tag{1.12}$$

$$\sum_{k=1}^{c} u_{ik} = 1 \quad 1 \le i \le n \tag{1.13}$$

$$0 < \sum_{i=1}^{n} u_{ik} < n \qquad 1 \le k \le c \tag{1.14}$$

Les degrés d'appartenance u_{ik} constituent la matrice U à c lignes (une ligne par classe) et n colonnes (une colonne par objet). En plus, chaque classe k est associée à un vecteur prototype $v_k= (v_{k1}, v_{k2}, \ldots, v_{kp})$ de dimension p. Ainsi, les algorithmes d'apprentissage permettent non seulement de déterminer les prototypes de classes, mais aussi les degrés d'appartenance de chaque objet aux classes.

1.1.3 Règles de décision

Pour structurer l'ensemble des données considéré, le recours à un critère de décision est nécessaire. Les critères de décision englobent plusieurs types : règle de décision, arbre de décision, réseaux de neurones...etc. Dans ce travail, on s'intéressera particulièrement aux règles de décision. En effet, dans plusieurs méthodes d'apprentissage non supervisé, l'appartenance d'un objet à une classe s'effectue selon une règle de décision basée généralement sur son degré de similarité soit avec les prototypes, soit avec ses voisins. Les règles de décision les plus utilisées sont présentées dans ce qui suit.

Règle du plus proche prototype (*1-PPP*)

L'objectif de cette règle, appelée également règle *1- NP*, est d'assigner un objet *y* à la classe qui correspond à son plus proche prototype v_k [Wilson, 1972]. Elle consiste donc à trouver en premier lieu le prototype le plus proche parmi les prototypes détectés [Devijver, 1980]. Généralement, la notion de proximité se base sur la distance entre l'objet et les prototypes. Il s'agit d'affecter un objet *y* à la $i^{ème}$ classe telle que :

$$i = \arg \min_{\substack{j \neq k \\ 1 \leq k \leq c}} \{ \| y - v_k \| \} \tag{1.15}$$

Règle des K plus proches voisins (*K-PPV*)

L'objectif de cette règle, notée également *K-NN* (k nearest neighbors), est de prédire la classe d'un nouvel objet en utilisant les exemples qui existent [Duda, 1973] [Dasarathy, 1990]. Il s'agit de connaître les *k* objets qui lui sont plus similaires, et affecter ce nouvel objet à la classe majoritaire. Mathématiquement parlant, on assigne l'objet *y* à la classe *i* si et seulement si la condition suivante est vérifiée :

$$i = \arg \max_{1 \leq j' \leq c} \{ V_{jk} \} \tag{1.16}$$

Où V_{jk} est le nombre de voisins de l'objet *y* issus de la classe *j*.

Cette règle est considérée comme une généralisation du principe qui consiste à affecter chaque nouvel objet à la classe de son plus proche voisin (*1-PPV*). Toutefois, l'avantage de *K-PPP* par rapport à *1-PPV* réside dans le fait que des grandes valeurs de *k* produisent un lissage qui réduit le risque de sur-apprentissage dû au bruit dans les données d'apprentissage [Turenne, 2006], alors que son inconvénient principal est lié au choix de *k* et au temps nécessaire pour calculer les *k* voisins.

Plusieurs méthodes ont été proposées pour choisir *k*. On en cite par exemple:

- l'utilisation d'un ensemble de test,
- la validation croisée,
- une heuristique qui choisit comme valeur de k, le nombre d'attributs augmenté d'un ($k = p + 1$).

Règle de maximum d'appartenance

La règle du maximum d'appartenance, appelée également règle majoritaire, permet d'assigner un objet à la classe pour laquelle son degré d'appartenance est le plus grand. Elle doit satisfaire les contraintes d'être maximale et voisine de 1 pour les objets qui appartiennent certainement à la classe, et de décroître vers 0 au fur et à mesure que les objets s'en éloignent.

Ainsi, la classe *i* d'affectation d'un objet *y* est définie comme suit :

$$i = \arg \max_{1 \le j \le c} \{ u_{jk} \} \tag{1.17}$$

1.2 Quelques approches non supervisées

Plusieurs algorithmes ont été intensivement étudiés dans la littérature pour décrire un apprentissage flou non supervisé des données [Bandemer, 1992]. Dans ce paragraphe, nous ne prétendons pas présenter exhaustivement les algorithmes existants, mais de présenter sommairement un aperçu général des algorithmes ayant été objet d'études comparatives dans les travaux de cette thèse.

Certains de ces algorithmes nécessitent que le nombre de classes soit spécifié à l'avance par l'utilisateur, alors que d'autres ne requièrent pas sa connaissance a priori [Suganya, 2012].

1.2.1 Approches nécessitant le nombre de classes

Parmi ces algorithmes, on trouve en premier le très populaire algorithme K-means [Macqueen, 1967] et l'algorithme c-moyennes floues (FCM) qui a été proposé pour modéliser l'incertitude d'appartenance [Dunn, 1973- Bezdek, 1981]. On trouve également l'algorithme c-moyennes possibilistes (PCM) [Krishnapuram, 1993]. Ces algorithmes sont présentés dans ce qui suit.

L'algorithme K-Means

k-means est un algorithme reconnu comme une technique d'apprentissage non supervisé classique. Il est très populaire et vise à partitionner un ensemble d'objets *X* en *k* classes séparées. La méthode commence par choisir, en général de façon aléatoire, *k* objets de *X* en tant que centres initiaux de classes. Pendant le partitionnement, chaque objet est affecté

uniquement à la classe la plus proche. La moyenne de chaque classe est recalculée et les objets peuvent se déplacer d'une classe à une autre. Le processus s'arrête lorsque les prototypes des classes ne changent plus.

Mathématiquement, l'algorithme K-means procède itérativement à la recherche d'une matrice de partition en minimisant une fonction objective qui représente la somme des carrés des distances entre tous les points d'une classe et son prototype:

$$J_k(X) = \sum_{j=1}^{k} \sum_{i \in C_j} d^2(x_i, c_j) \tag{1.18}$$

Où:

x_i est un vecteur représentant l'$i^{ème}$ objet.

c_j est le prototype de la classe C_j.

$d(x_i,c_j)$ est la distance entre le $j^{ème}$ prototype et le $i^{ème}$ objet.

L'apprentissage non supervisé, étant classique, chaque objet est assigné à une unique classe dont le prototype lui est le plus proche. Une partition de X en k classes peut être alors représentée par k sous-groupes mutuellement disjoints $X_i \ldots X_k$, tel que $X_1 \cup X_2 \ldots \cup X_k = X$. En plus, le degré d'appartenance des objets aux classes détectées vérifient la condition suivante :

$$u_{ij} \in \{0,1\} \quad \forall 1 \le i \le n \quad \forall 1 \le j \le k \tag{1.19}$$

En plus, puisque chaque objet appartient à une classe et une seule, et aucune classe n'est vide, les équations suivantes sont vérifiées :

$$\sum_{j=1}^{k} u_{ij} = 1, \quad \forall 1 \le i \le n \tag{1.20}$$

$$0 < \sum_{i=1}^{n} u_{ij} < n \qquad 1 \le j \le k \tag{1.21}$$

En résumé, K-means passe principalement par trois étapes [Rammal, 2010]:

- initialisation des prototypes ;
- affectation de chaque objet à la classe dont il est le plus proche ;
- recalcul des nouveaux prototypes.

Ces étapes sont représentées par le pseudo-code suivant:

Algorithme 1.1 : Algorithme K-means

Entrées: Ensemble de données non étiquetées $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$;
Le nombre de classe k; les k centres initiaux

Sorties : k classes avec leurs centres.

Début

Répéter

Attribuer chaque objet x_i à son centre le plus proche.
Re-calculer le centre de chaque classe en utilisant l'équation suivante:

$$c_j = \frac{\sum_{i=1}^{n_j} x_i}{n_j} \qquad 1 \leq j \leq k \tag{1.22}$$

où n_j représente le nombre d'objets de la $j^{ème}$ classe

Jusqu'à les centres des classes ne changent plus

Fin

Le choix initial des prototypes influence le résultat. Il est donc souvent nécessaire de procéder à plusieurs classifications, et comparer leurs résultats afin d'en extraire les classes stables [Rammal, 2010]. Toutefois, il est souvent difficile de déterminer à quelle classe on doit associer un objet ayant des caractéristiques communes avec plusieurs prototypes. Le choix d'établir une partition stricte s'avère ainsi limitatif. Il est possible de pallier cette limite en permettant que l'appartenance soit non plus binaire mais graduelle. C'est le but de l'algorithme FCM.

L'algorithme FCM

FCM (fuzzy c-means) est un algorithme d'apprentissage flou non supervisé qui constitue une généralisation de K-means [Parizeau, 2004]. Développé par Bezdek [Bezdek 1981] à partir des idées de Ruspini [Ruspini 1969] et de Dunn [Dunn 1973], l'algorithme introduit la notion des ensembles flous dans la définition des degrés d'appartenance. En effet, il associe à chaque objet des degrés d'appartenance à chacune des classes. Ces degrés d'appartenance forment une matrice de partition floue $\boldsymbol{U}= (u_{ik})$ calculée en minimisant une fonction objective J_m définie par :

$$J_m(U,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d^2(x_k, v_i) \tag{1.23}$$

Où:

m (1 < m <∞) est un exposant de pondération utilisé pour contrôler la contribution de chaque objet x_i. On remarque que plus m tend vers 1, plus la classification devient dure et u_{ik} se rapproche de 0 ou de 1. Inversement quand m devient trop grand, la distribution des degrés d'appartenance tend à se concentrer autour de *1/c*. Le coefficient *m* mesure également la convergence de l'algorithme. Il n'existe pas de critère permettant de choisir *m* de manière optimale, la plupart des auteurs préconisent *m=2* [Pal, 1995] [Khodja, 1997].

$V= (v_1, v_{2,} \ldots ,v_c)$ représente un *c* triplet de prototypes, chaque prototype caractérise l'une des *c* classes.

$d(x_k,v_i)$ est la distance entre le $i^{ème}$ prototype et le $k^{ème}$ objet.

Cette fonction objective J_m est définie en fonction de la matrice de données, de la matrice d'appartenance et des prototypes de classes. Elle mesure la dissimilarité globale des objets de données au sein de chaque classe. Par conséquent, en minimisant la fonction objective, nous pouvons obtenir la meilleure partition de l'ensemble de données.
Bezdek a prouvé que FCM converge vers une solution approchée sous deux conditions [Bezdek, 1981]:

$$u_{ik} = \left[\sum_{j=1}^{c}\left(\frac{d(x_k, v_i)}{d(x_k, v_j)}\right)^{2/m-1}\right]^{-1} \quad 1 \le i \le c \quad 1 \le k \le n \tag{1.24}$$

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik})^m \; x_k}{\sum_{k=1}^{n}(u_{ik})^m} \qquad 1 \le i \le c \tag{1.25}$$

La matrice $\boldsymbol{U}= (u_{ik})$ satisfait les trois conditions suivantes [Bezdek, 1981]:

$$0 \le u_{ik} \le 1 \qquad 1 \le i \le c \quad 1 \le k \le n \tag{1.26}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \qquad 1 \le i \le c \tag{1.27}$$

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad 1 \le k \le n \tag{1.28}$$

Où u_{ik} est le degré avec lequel l'élément x_k appartient à la $i^{ème}$ classe ($1 \le i \le c$ *et* $1 \le k \le n$).

L'équation (1.26) reflète la généralisation de la fonction caractéristique qui prend des valeurs dans {0, 1}. Ainsi, une valeur de u_{ik} proche de 1 indique une forte appartenance de l'objet x_i à la classe k, alors qu'une valeur proche de 0 indique une faible appartenance. D'autre part, l'équation (1.27) garantit qu'aucune classe n'est totalement vide ou égale à X. Quant à la dernière équation (1.28), elle assure que les degrés d'appartenance de chaque objet sont répartis sur toutes les c classes.

Sur le plan procédural, FCM est un processus itératif qui commence par initialiser les prototypes ou les degrés d'appartenance et choisir une norme pour calculer la fonction J_m. A chaque itération, ces derniers sont mis à jour et un objet est associé à la classe dont le degré d'appartenance est le plus élevé. Cette procédure de mise à jour est itérée tant que la variation maximale des degrés d'appartenance ou des prototypes reste en dessous d'un seuil [Chelloug, 2006], ou jusqu'à ce qu'un nombre maximum d'itérations t_{max} soit atteint. Le pseudo-code de l'algorithme FCM est donné par l'algorithme 1.2.

Algorithme 1.2 : Algorithme FCM

Entrées: Ensemble de données non étiquetées $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$
Le nombre de classe c
Le paramètre m ($m > 1$); t_{max} (Nombre maximal des itérations); le seuil ε
prototypes $V_0 = (v_{1,0}, v_{2,0}, \ldots, v_{c,0}) \in \Re^{cxp}$

Sorties : c classes avec leurs centres
La matrice d'appartenance U

Début
$t \leftarrow 0$;
Faire
$t \leftarrow t+1$;
Calculer U_t en utilisant V_{t-1} et (Eq.1.24);
Calculer V_t en utilisant U_t et (Eq.1.25);
Tant que ($\|V_t - V_{t-1}\|_{err} > \varepsilon$) et ($t < t_{max}$));
$U^* \leftarrow U_t$; $V^* \leftarrow V_t$;
Fin

Dans l'algorithme FCM, les degrés d'appartenance sont liés (Eq.1.28). Selon Zadeh [Zadeh, 1965], les degrés d'appartenance ne devraient qu'appartenir à l'intervalle [0,1]. Pour surmonter cette contrainte, l'algorithme PCM a été proposé.

L'algorithme PCM

Krishnapuram et Keller ont suivi les idées de Zadeh sur les sous-ensembles flous et l'appartenance des degrés d'appartenance à l'intervalle [0,1] et ont proposé l'algorithme c-moyennes possibilistes (PCM) [Krishnapuram, 1993]. Cet algorithme considère les degrés d'appartenance indépendants et n'exige pas que la somme des degrés d'appartenance d'un objet soit nécessairement égale à 1, telle mentionnée par l'équation (Eq.1.28). Ainsi, un nouvel ensemble de contraintes est défini :

$$\begin{cases} 0 < \sum_{k=1}^{n} u_{ik} < n \quad 1 \le i \le c \\ \max_{i}(u_{ik}) > 0 \end{cases} \tag{1.29}$$

Cette équation assure que la partition floue résultante de l'algorithme recouvre l'ensemble X en entier [Jeongmin, 2012].

PCM optimise la fonction objective J_m définie comme suit:

$$J_m(U,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d^2(x_k,v_i) + \sum_{i=1}^{c}\eta_i\sum_{k=1}^{n}(1-u_{ik})^m \tag{1.30}$$

Avec:

η_i ($1 < i < c$) est un paramètre d'échelle défini par :

$$\eta_i = K\frac{\sum_{k=1}^{n}(u_{ik})^m d^2(x_k,v_i)}{\sum_{k=1}^{n}(u_{ik})^m} \qquad K > 0 \tag{1.31}$$

Ainsi, le degré d'appartenance u_{ik} est défini par :

$$u_{ik} = \left[1+\left(\frac{d^2(x_k,v_i)}{\eta_i}\right)^{1/m-1}\right]^{-1} \quad 1 \le i \le c \quad 1 \le k \le n \tag{1.32}$$

Cette équation (1.32) montre que le degré d'appartenance u_{ik} ne dépend que de la distance de l'objet x_i à la classe k.

L'avantage majeur des algorithmes K-means, FCM et PCM est le temps de calcul. Leurs résultats dépendent de l'initialisation, mais PCM en souffre plus. En effet, un mauvais choix des centres initiaux risque de limiter la performance de l'algorithme, et même empêcher sa convergence [Parizeau, 2004]. Pour cela, certains chercheurs proposent de commencer avec d'autres algorithmes, notamment FCM, pour initialiser les données [Ménard. M, 1998]. Ce dernier donne une première estimation des matrices des degrés d'appartenance U et du vecteur des prototypes V.

1.2.2 Approches ne nécessitant pas le nombre de classes

Les algorithmes K-means, FCM et PCM nécessitent de spécifier le nombre de classes à priori. Leur critère d'arrêt est basé sur la stabilité des centres des classes. Ceci constitue une limite. ISODATA et AFNS ont été proposés comme méthodes de partitionnement qui ne nécessitent pas le nombre de classes.

L'algorithme ISODATA flou

L'algorithme ISODATA (Iterative Self-Organizing Data Analysis Technique) est un algorithme auto-organisateur et itératif basé sur l'approche par centre mobile. Son principe consiste à introduire au cours des itérations successives, de nouveaux paramètres afin de modifier le nombre de classes. Ainsi il effectue plusieurs itérations avant sa stabilité [Theodoridis et Koutroumbas, 1999]. Pour améliorer le processus de l'apprentissage, l'algorithme emploie trois processus ou opérations: l'élimination, le fractionnement et la fusion [Ball, 1966].

L'algorithme commence avec k_{init} prototypes, où K_{init} est le nombre initial choisi des classes. Comme dans FCM, Isodata choisit les K_{init} premiers objets comme prototypes de classes. Les autres objets sont ensuite affectés aux classes selon le principe de la distance minimale, en respectant un certain nombre de contraintes imposées pour empêcher la formation de classes avec un faible cardinal ou de diamètre trop grand [Rammal, 2010]. De ce fait, toutes les classes sont reconsidérées dans les phases d'élimination, de division ou de fusion des classes.

- Dans la phase d'élimination, les classes dont le nombre d'objets est inférieur à la taille minimale des classes n_{min} sont supprimées et leurs objets sont réaffectés.

– Dans la phase de division, une classe est divisée en deux classes, si son écart-type est supérieur à une valeur seuil (par exemple la variance). Les prototypes sont recalculés pour les deux nouvelles classes obtenues.

– Lors de la phase de fusion, deux ou plusieurs classes peuvent être regroupées si la distance entre ces classes est inférieure à un seuil (distance minimum entre les centres δ).

Ces trois opérations ont été ajoutées afin d'améliorer l'apprentissage. Ainsi, à chaque itération, toutes les classes sont examinées et le nombre de classes peut varier tout au long de l'apprentissage.

En conséquence, l'ajout de ces règles permet de remédier à l'inconvénient du choix du nombre de classes. Toutefois, la performance de l'algorithme Isodata est fonction des paramètres suivants:

- K_{init}, le nombre initial de classes ;
- n_{min}, le nombre minimal d'objets dans une classe ;
- δ , la distance minimale entre deux prototypes ;
- V_{max}, la variance maximale autorisée dans une classe.
- t_{max}, le nombre maximal d'itérations.

Algorithme 1.3 : Algorithme ISODATA

Entrées: Ensemble de données non étiquetées $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$
Le nombre de classe c ; K_{init}
t_{max} : Nombre maximal des itérations
n_{min} : nombre minimal d'objets dans les classes
δ : distance minimale entre les centres
$V_{\max}$: variance maximale dans une classe
prototypes $V_0 = (v_{1,0}, v_{2,0}, \ldots, v_{c,0}) \in \Re^{cxp}$

Début

$c \leftarrow K\_init$
initialiser $V_0 = (x_1, x_2, \ldots, x_c) \in \Re^{cxp}$
$t \leftarrow 0$

Répéter
$t \leftarrow t+1$
Pour *chaque* centre $c_{k,t-1}$ ***faire***
Construire $S_k = \{ x_i / d(x_i, c_{k,t-1}) < d(x_i, c_{j,t-1}) \} \quad \forall j \neq k$
Si (*Card* (S_k) < n_{min}) ***Alors***
Supprimer (c_k)
Finsi
Si ($\exists c_{j \neq k} / d(c_k, c_j) < \delta$ **) Alors**
Fusionner (c_k,c_j); $c \leftarrow c - 1$
Finsi
Si ($\mathrm{var}(S_k) > V_{\max}$ **) Alors**
Diviser (c_k); $c \leftarrow c + 1$
Finsi
Calculer V_t
Finpour
jusqu'à **(** $\exists k \in [1, c] / c_{k,t} \neq c_{k,t-1}$ et $(t < t_{max})$ **)**;
$V^* \leftarrow V_t$
Fin

L'algorithme AFNS (Apprentissage flou non supervisé)

Cet algorithme tente de combiner les avantages des approches hiérarchiques et de partitionnement [Bouroumi, 2000] pour partitionner l'ensemble X et déterminer automatiquement le nombre de classes. Pour cela, l'algorithme commence par générer une première classe dont le prototype est initialisé avec le premier objet. Ensuite, les autres *n-1* objets sont successivement examinés de façon itérative. Ainsi, l'algorithme analyse leurs similarités en utilisant une nouvelle mesure de similarité définie par l'équation (1.35). Il utilise

également un seuil ξ pour détecter si un objet courant n'est pas reconnu et, par conséquent, différent de tous les prototypes existants. Dans ce cas, une nouvelle classe est créée et son prototype est initialisé avec cet objet. Ce seuil ξ représente le minimum de similarité que chaque objet doit avoir avec son prototype le plus proche.

L'algorithme AFNS utilise la mesure de similarité et son seuil associé ξ pour construire les classes. Deux cas sont considérés:

Cas 1 :

$$\underset{1\le k\le c}{Max}\,(S(i,k)) \;<\; \xi \tag{1.33}$$

Cela signifie que l'élément courant x_i ne répond à aucun critère de similarité reconnue dans les prototypes détectés précédemment et donc doit être choisi pour représenter une nouvelle classe. Ainsi, on pose $c = c+1$ et $v_c = x_i$.

Cas 2 :

$$\underset{1\le k\le c}{Max}\,(S(i,k)) \;\ge\; \xi \tag{1.34}$$

x_i est considéré comme ayant la similarité minimale requise pour les classes précédemment détectées et on ne crée pas de nouvelle classe.

La fonction de similarité utilisée pour mesurer les similarités des objets par l'algorithme AFNS est définie par:

$$S(x_i,x_k)=1-\frac{\|x_i-x_k\|_A^2}{p} \tag{1.35}$$

Où A est la matrice pxp définie par:

$$A_{jt}=\begin{cases}(r_j)^{-2} & \text{Si } \; j=t\\ 0 & \text{Sinon}\end{cases} \tag{1.36}$$

r_j représente l'écart des valeurs de la $j^{ème}$ caractéristique des objets de l'ensemble X $(1\le j\le p)$. Il est défini par:

$$r_j=\max_{1\le i\le n}\{x_{ij}\}-\min_{1\le i\le n}\{x_{ij}\} \qquad 1\le j\le p \tag{1.37}$$

L'algorithme ne fait aucune hypothèse sur le nombre de classes c. Le choix de c dépend automatiquement du choix du seuil ξ puisque la création d'une nouvelle classe dépend de sa

valeur. Ainsi, la qualité des classes détectées dépend principalement de ce seuil. En variant ξ entre deux valeurs: $\underset{\substack{1\le i,j\le n\\ i\neq j}}{Min}(S(i,j))$ et $\underset{\substack{1\le i,j\le n\\ i\neq j}}{Max}(S(i,j))$, différents ensembles de c prototypes peuvent être détectés.

Les prototypes des classes précédemment créées sont ensuite mis à jour selon le schéma d'apprentissage suivant:

$$v_k(i) = v_k(i-1) + \frac{S(x_i, v_k)}{n_k(i)}\left[x_i - v_k(i-1)\right] \quad 1 \le k \le c \quad c \ge 2 \tag{1.38}$$

Avec:

$v_k(i)$, $v_k(i-1)$ sont respectivement le prototype de la $k^{ème}$ classe avant et après l'analyse de x_i.

$n_i(k)$ désigne le cardinal flou de la $k^{ème}$ classe après analyse du x_i, défini par:

$$n_i(k) = \sum_{j=1}^{k} S(x_i, v_k) \qquad 1 \le k \le c,\ i \le n \tag{1.39}$$

Algorithme 1.4 : Algorithme AFNS

Entrées : Ensemble de données non étiquetés $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$;

Sorties: Nombre de classes c;
Centres des classes; Matrice des degrés d'appartenance

Début

$c \leftarrow 1$
$v_1 \leftarrow x_1$
Pour i=2 à n **faire**
Si $(\underset{1\le j\le c}{Max}(Sim(i,j)) < \xi)$ **alors**
$c = c \leftarrow 1;\ v_i \leftarrow x_i$
Sinon
Mise à jour des prototypes v_j, $1 \le j \le c$ (Eq. 1.38)
Finsi
Calculer U selon (Eq.1.24)
Calculer V selon (Eq.1.25)
Finpour

Fin

1.3 Critères de validité

En principe, les algorithmes d'apprentissage non supervisé sont conçus d'une manière qui donne une « *plausible*» partition. Cependant le choix des différents paramètres requis par ces algorithmes peut conduire à des partitions différentes, surtout dans le contexte non supervisé où il n'existe aucune information a priori sur la structure interne des données ou sur le nombre de classes. Ainsi, la partition obtenue peut ne pas être naturelle et le nombre de classes peut ne pas être correct. De ce fait, on recourt à des critères de validité pour valider la qualité des classes obtenues [Bezdek, 1998] [Bensaid, 1999]. Ce sont des indices de performance qui servent à déterminer le nombre de classes, et à évaluer la qualité de la structure des données produite par un algorithme d'apprentissage non supervisé [Halkidi, 2001] [Weng, 2007].

Différents critères de validité ont été proposés dans la littérature. Cependant, aucune d'eux n'est complétement parfait et un seul indice de validité ne peut pas produire des résultats consistants pour différentes structures de données. Par conséquent, les chercheurs utilisent plusieurs indices. On peut les classer généralement en deux catégories [Aboualala, 2002]:

- Critères de validité basés uniquement sur la matrice d'appartenance associée à la partition floue découverte.

- Critères de validité basés sur la matrice d'appartenance et la distribution géométrique des objets.

Dans ce paragraphe, on décrit les indices les plus usuels [Jansen, 2007].

1.3.1 Critères associés à la matrice d'appartenance

Ce sont des indices qui utilisent uniquement les valeurs des degrés d'appartenance floues pour évaluer les résultats d'apprentissage non supervisé, tels l'entropie et le coefficient de partition.

L'entropie

Ce coefficient mesure le flou de la classe [Bezdek, 1981]. Il est calculé par :

$$PE(U) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c}[u_{ik}\log_a(u_{ik})] \tag{1.40}$$

L'indice *PE* est calculé pour des valeurs de c supérieures à 1, et ses valeurs sont comprises dans l'intervalle [*0, $log_a c$*]. Par ailleurs, plus la valeur de *PE* est proche de 0, plus la partition est stable. Alors que les valeurs de *PE* proches de $log_a c$ indiquent l'absence de structure de groupement dans l'ensemble de données considéré et la présence de partition floue.

Coefficient de partition

Ce coefficient mesure le degré de chevauchement entre les classes. Il est défini par Bezdek [Bezdek, 1981] comme suit:

$$PC(U) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c}(u_{ik})^2 \tag{1.41}$$

Où u_{ik} est le degré d'appartenance du point de données x_i à la classe *k*. Les valeurs de *PC* se situent dans l'intervalle *[1/c, 1]*. Plus la valeur de l'indice est proche de 1, plus la partition n'est pas floue. Dans le cas où toutes les valeurs d'appartenance à une partition floue sont égales, c'est-à-dire $u_{ij} = 1/c$, la valeur du coefficient *PC* est minimale et la partition est floue. Ainsi, le nombre optimal des classes correspond à la valeur maximale de ce coefficient.

Coefficient de partition de Dave

Les indices de validité *PC* et *PE* possèdent une tendance d'évolution monotone avec *c*. Dave a proposé une modification de l'indice *PC* pour réduire cette tendance [Kuo-Lung Wu, 2004]. Il est défini par :

$$MPC(U) = 1 - \frac{c}{c-1}(1 - PC(U)) \tag{1.42}$$

Avec $0 \leq MPC(U) \leq 1$. En plus, le nombre optimal de classes *c** correspond au maximum de *MPC(U)*.

1.3.2 Critères associés à la matrice d'appartenance et à la structure des données

Au lieu d'avoir des indices de validité qui ne prennent en compte que les valeurs des degrés flous d'appartenance dans leur définition, des auteurs ont introduit des critères de validité qui prennent en compte également la structure des données. Ces critères sont définis en se basant sur la compacité floue et la séparation.

Indice de Xie and Beni

C'est un indice qui mesure la compacité globale et la séparation des classes [Xie, 1991]. Il est défini par :

$$XB(U,V;X) = \frac{1}{n}\frac{J_m(U,V;X)}{\min_{i\neq r}\left\{\|v_i - v_r\|^2\right\}} \tag{1.43}$$

Où :

$J_m(U;V;X)$ est la fonction objective définie par l'équation (1.23)

$\min_{i\neq r}\left\{\|v_i - v_r\|^2\right\}$ mesure la séparation entre les centres/prototypes des classes.

La valeur minimale de l'indice *XB* indique le nombre optimal des classes.

Indice de Fukuyama and Sugeno

Cet indice est défini par l'équation [Fukuyama, 1989] :

$$FS(U,V;X) = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m}\left(\|x_i - v_j\|_A^2 - \|v_j - \bar{v}\|_A^2\right) \tag{1.44}$$

Où : $\bar{v} = \frac{1}{n}\sum_{i=1}^{n} x_i$ est la moyenne des données.

Cet indice est la différence entre deux termes. Le premier terme est la fonction objective J_m définie par l'équation (Eq.1.23) et qui mesure la compacité des classes. Le deuxième terme mesure les distances entre les prototypes des classes et peut être considéré comme une mesure floue de séparation entre les c prototypes. Selon cet indice, la meilleure partition est celle qui minimise *FS*.

Indice de Zahid

L'indice de validité défini par Zahid [Zahid, 1999] se base sur la combinaison de deux fonctions qui mesurent la séparation et la compacité. La première, notée SC_1, considère les propriétés géométriques de la structure des données et les degrés d'appartenance flous. La seconde, notée SC_2, considère uniquement les degrés d'appartenance flous. L'indice est présenté comme suit :

$$Z(U,V;X) = SC_1(U,V;X) - SC_2(U) \tag{1.45}$$

Avec :

$$SC_1(U,V;X) = \frac{\frac{1}{c}\sum_{i=1}^{c}\left\|v_i - \bar{v}\right\|^2}{\sum_{i=1}^{c}\left\{\frac{1}{n_i}\sum_{k=1}^{n}(u_{ik})^m \left\|x_k - v_i\right\|_A^2\right\}} \quad (1.46)$$

$$SC2(U) = \frac{\sum_{i=1}^{c-1}\sum_{j=i+1}^{c}\left\{\frac{1}{n_{ij}}\sum_{k=1}^{n}\left[\min(u_{ik},u_{jk})\right]^2\right\}}{\frac{1}{n_{\cup}}\sum_{k=1}^{n}\max_{1\le i\le c}(u_{ik})^2} \quad (1.47)$$

Où :

$n_i = \sum_{k=1}^{n} u_{ik}$ est le cardinal flou de la i^{ème} classe ($1 \le i \le c$)

$n_{ij} = \sum_{k=1}^{n} \min(u_{ik}, u_{jk})$ est le cardinal flou de l'intersection floue des classes *i* et *j*

$n_{\cup} = \sum_{k=1}^{n} \max_{1\le i\le c}(u_{ik})$ est le cardinal flou de l'union floue des *c* classes

$\bar{v} = \frac{1}{n}\sum_{k=1}^{n} x_k$ est la grande moyenne des données

Conclusion

Au cours de ce chapitre consacré à une présentation des éléments de base d'apprentissage non supervisé, nous avons présenté des notions nécessaires à la compréhension de l'apprentissage flou non supervisé. Nous avons ainsi présenté les traits communs aux techniques d'apprentissage flou non supervisé les plus connues, et qui portent sur le prétraitement des données, le problème d'initialisation et le problème du nombre de classes. Nous avons souligné que certaines de ces techniques exigent la connaissance a priori du nombre de classes alors que d'autres détectent automatiquement ce nombre lors du partitionnement. Nous avons également présenté les indices de validité les plus usuels permettant l'évaluation des techniques d'apprentissage mentionnées. Certains de ces critères de validité sont basés uniquement sur la matrice d'appartenance associée à la partition floue découverte, alors que d'autres prennent en compte également la structure géométriques des objets.

Les méthodes présentées dans ce chapitre se basent sur des règles de décision pour construire les classes. A cet effet, un aperçu a été introduit sur les règles de décision les plus usuelles. Ces méthodes partagent plusieurs traits communs, mais sont également accompagnées de

quelques inconvénients qui portent notamment aux paramètres d'entrée, à la robustesse aux valeurs aberrantes et l'indépendance de l'ordre des données [Chelloug, 2006].

Par ailleurs, les techniques d'apprentissage flou non supervisé reposent généralement sur des mesures de similarité. Celle-ci est un sujet très important, mais sous-estimé. C'est l'objet du chapitre suivant.

CHAPITRE 2

LES MESURES DE SIMILARITE EN APPRENTISSAGE FLOU NON SUPERVISE

Introduction

La notion de similarité est essentielle à l'apprentissage non supervisé dans le sens que c'est elle qui permet de former des classes [Guerif, 2006]. En effet, le critère de formation des classes consiste à maximiser la mesure de similarité intra-classes et à minimiser la mesure de similarité inter-classes. De ce fait, le choix d'une mesure appropriée de similarité peut parfois être plus important que le choix de l'algorithme de l'apprentissage non supervisé lui-même [Halkidi, 2001].

Une façon classique de quantifier la similarité entre deux éléments est de définir une distance numérique. Souvent, le choix de cette distance est arbitraire et sensible à la représentation des objets. En plus, la plupart des distances considèrent les attributs des objets de la même manière. Or, certains attributs ont plus d'importance dans la construction de certaines classes [Candillier, 2006]. Dans ce sens, Cosmin Lazar affirme que « *toutes les distances entre les paires de points d'un ensemble quelconque sont presque identiques et donc la notion du plus proche voisin ou de similarité devient instable* » [Lazar, 2008]. Ceci est appelé le phénomène de concentration [Doherty, 2004] [Lazar, 2008]. Pour pallier ce problème, certains auteurs proposent l'utilisation des distances fractionnaires [Doherty, 2004] [Lazar, 2008].

Dans ce chapitre, nous allons introduire la notion de mesure de similarité ainsi que sa relation étroite avec les mesures de distances, en effectuant également une étude comparative des distances les plus utilisées. Nous présenterons également les distances fractionnaires pour quantifier le degré de dissemblance entre les paires d'objets en cherchant à déterminer le lien qui existe entre les données et la valeur du coefficient *r* de la distance fractionnaire. On va soulever aussi l'intérêt de la normalisation ou l'utilisation des distances pondérées dans les tâches d'apprentissage non supervisé en définissant de nouvelles distances normalisées.

2.1 Distances et mesures de similarités

Plusieurs mesures de similarité ont été proposées dans la littérature au cours des dernières décennies pour mesurer le degré de ressemblance entre les objets représentés par un ensemble d'attributs mesurables [Sneath, 1973] [Clifford, 1975] [West, 2006]. Une façon classique de quantifier la similarité entre deux éléments est de définir une distance numérique. Ces distances se différent par leurs formules mathématiques mais respectent toutes un certain nombre de règles.

2.1.1 Similarité

Une mesure de similarité est une fonction mathématique qui permet de regrouper des objets en classes [Lazar, 2008]. Généralement, c'est une fonction qui quantifie le degré de ressemblance entre deux objets [Haidar, 2005]. Ainsi, la similarité entre deux objets considérés est maximale lorsque ces deux objets sont identiques et minimale lorsqu'ils sont complètement différents.

Mathématiquement, une mesure de similarité s sur un ensemble de points $X=\{x_1, x_2, \ldots, x_n\} \subset \mathcal{R}^p$ est une fonction $s: X \times X \rightarrow R^+$ tel que $s(x, y)$ entre deux points x et y doit satisfaire les conditions suivantes:

$$0 \leq s(x,y) \leq 1 \quad \forall\ (x,y) \tag{2.1}$$

$$s(x,x) = 1 \quad \forall\ x \tag{2.2}$$

$$s(x,y) = s(y,x) \quad \forall\ (x,y) \tag{2.3}$$

Une similarité s peut être calculée à partir d'une distance d selon une des équations suivantes [Bezdek, 1981] [Bandemer, 1992] :

$$s(x,y) = \frac{1}{1+d(x,y)} \tag{2.4}$$

$$s(x,y) = \frac{1}{1+d^*(x,y)} \tag{2.5}$$

$$s(x,y) = 1-d^*(x,y) \tag{2.6}$$

$$s(x,y) = e^{-d^*(x,y)} \tag{2.7}$$

Où d^* est une distance normalisée et dérivée de d dans [0, 1].

Le choix d'une mesure de similarité est très important pour la qualité des résultats obtenus, mais il n'est pas trivial et dépend du contexte. Plusieurs études menées dans ce sens l'affirment. Toutefois, elles recommandent de tenir compte des éléments suivants: ses propriétés mathématiques, la nature des données (numérique, symbolique), l'utilisation qui sera faite de la matrice de similarité (calcul de matrice de covariance) [Gower et Legendre, 1986].

Le plus souvent, on mesure la similarité entre les prototypes et les données considérées. Toutefois, on partitionne parfois selon la densité où l'on estime la densité des données en fonction de leur similarité. Ainsi, un regroupement de données similaires peut être défini comme une région dense. Les zones de faible densité définissent les limites des classes [Cabanes, 2010].

2.1.2 Distances usuelles

Une fonction de distance *d* sur un ensemble de points *E* est définie par *d: E* x $E \rightarrow R^{+}$ et la distance *d(x, y)* entre deux points *x* et *y* vérifie les propriétés suivantes :

$$d(x,y) = 0 \Leftrightarrow x = y \quad \forall\,(x,y) \tag{2.8}$$

$$d(x,y) \;=\; d(y,x) \quad \forall\;(x,y) \tag{2.9}$$

$$d(x,z) \;\leq\; d(x,y) + d(y,z) \quad \forall\;(x,y,z) \tag{2.10}$$

Différentes formules ont été proposées dans la littérature pour définir une mesure de distance. Le choix de la mesure de distance impose souvent la forme des classes à découvrir [Guerif 2006]. Ainsi par exemple, l'algorithme des K-moyennes a tendance à former des groupes hyper-sphériques par son utilisation d'une distance euclidienne, alors que la matrice de covariance diagonale de l'algorithme de Gustafson et Kessel [Gustafson, 1978] impose la formation de classes hyper-ellipsoïdales.

La distance la plus usuelle est la distance L_r définie par :

$$d(x_i, x_j) = \left(\sum_{k=1}^{p} \left| x_{ik} - x_{jk} \right|^{r} \right)^{\frac{1}{r}} \tag{2.11}$$

La mesure de distance L_r est une métrique pour $r \geq 1$ mais ne l'est pas pour $r < 1$ parce que la propriété de l'inégalité triangulaire n'est pas satisfaite pour $r < 1$ [François, 2007]. Cependant, cette propriété n'est pas nécessaire pour les tâches d'apprentissage non supervisé.

La métrique L_r dépend du paramètre r ($r \in \Re^+$) appelé exposant de la métrique, et couvre la distance de Minkowski ($r >= 1$) et les métriques fractionnaires ($r < 1$).

Distance de Minkowski

Elle est efficace pour des données qui représentent des classes «compactes » ou « séparées » [Jain et al, 1999]. Son inconvénient principal est la tendance de certaines données à dominer les autres. La normalisation des données est ainsi nécessaire pour remédier à ce problème.

Il est facile de voir d'après l'équation (2.11) que la distance euclidienne, L_2, la distance de Manhattan, L_1, et la distance de Chebyshev, L_∞, sont des cas particuliers de la distance de Minkowski. Diverses approches ont employé, adapté ou étendu cette métrique [Agrawal,1993] [Yi,2000] [Gionis,1999] [Goldin,1995] [Keogh,2001] [Kahveci,2001] [Chan,1999] [Faloutsos,1994] [Rafiei,1997] [Shahabi,2000] [Singh,1998] [Haidar, 2005].

Distance euclidienne

Elle est probablement la distance la plus couramment utilisée. Elle est calculée comme suit:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} \qquad (2.12)$$

La distance euclidienne est souvent utilisée dans des espaces à deux ou trois dimensions. Elle est fortement influencée par les grandes unités de mesure et varie en fonction de l'échelle de chaque variable. En effet, elle favorise les attributs d'un ordre d'échelle plus significatif et la contribution des attributs moins significatifs est ignorée [Gretton, 2005] [Guaus, 2009]. C'est pourquoi la distance euclidienne est souvent calculée après centrage, réduction ou normalisation de variables [Cha, 2006] [Lazar, 2008] [Dietterich, 2009].

Distance de Manhattan

La distance de Manhattan ($r=1$) entre deux vecteurs est calculée en additionnant la valeur absolue de la différence des attributs correspondant à chaque dimension. Elle est aussi appelée «city block distance» ou L_1 métrique. Elle est moins sensible au bruit et définie par:

$$d(x_i, x_j) = \sum_{k=1}^{p} \left| x_{ik} - x_{jk} \right| \tag{2.13}$$

Schématiquement, elle consiste à déterminer la distance qui serait parcourue pour aller d'un point à l'autre si un chemin en forme de grille est suivi. Son coût du calcul est plus faible que celui de la distance euclidienne qui nécessite de calculer des carrés.

Distance de Spearman

La distance de Spearman est le carré de la distance euclidienne entre deux vecteurs. Elle permet de « surpondérer » les objets éloignés. Le temps de son calcul est moins long que celui de la distance euclidienne, puisqu'elle n'exige pas la racine carré.

$$d(x_i, x_j) = \sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \tag{2.14}$$

Distance de Chebyshev

La distance de Chebyshev, ou L_∞ métrique, est aussi appelée « chess board » distance. Elle calcule la distance maximale entre les attributs de deux objets. Elle est aussi appelée L_∞ métrique.

$$d(x_i, x_j) = \max_{1 \le k \le p} \left| x_{ik} - x_{jk} \right| \tag{2.15}$$

Distances fractionnaires

Généralement, la métrique fractionnaire ($r < 1$) a rarement été utilisée dans les tâches de l'apprentissage non supervisé, et l'impact de son utilisation dans ce contexte soulève de nos jours plusieurs controverses. En effet, dans [Doherty, 2004], l'auteur affirme que l'utilisation d'une norme fractionnaire permet de réduire l'impact des différences d'attributs individuels extrêmes. D'autre part, dans [Aggarwal, 2001], les auteurs montrent que des valeurs ($r < 1$) atténuent le phénomène de concentration et ainsi les métriques fractionnaires peuvent fournir

de meilleurs résultats. Hathaway [12] a proposé de généraliser l'utilisation des distances L_r et a donné des exemples avec ($0,5 \leq r < 1$).

Toutefois, dans [François, 2007], les auteurs montrent qu'il existe des données pour lesquelles les métriques d'ordre supérieur sont moins concentrées que les métriques fractionnaires. Ils montrent que les résultats obtenus en [Aggarwal, 2001] ne sont pas toujours valables. Or, les résultats théoriques de Hainneburg et al [Hinneburg, 1999] montrent que pour toutes les métriques « L_r » avec ($r \geq 3$), la recherche du plus proche voisin n'a pas de sens car la valeur maximale de la distance entre deux objets converge vers la valeur minimale quand la dimension des données augmente. Ceci s'explique comme si la métrique avait perdu ses capacités discriminatoires entre la notion de « près » et de « loin ».

Ainsi, lutter contre la concentration est tout à fait justifié, et l'impact des normes sur les résultats de l'apprentissage est très grand.

2.1.3 Normalisation et distances pondérées

Méthodes de normalisation des données

En apprentissage non supervisé, le choix de la similarité peut affecter le résultat final. De ce fait, certains proposent de définir la mesure de similarité en fonction des données elles-mêmes, en particulier dans un contexte non supervisé ou aucune pré-information sur la structure des données n'est disponible [Bouroumi, 2000] [El Imrani, 2000]. Toutefois, certaines caractéristiques des objets considérés peuvent présenter des caractéristiques pénalisantes [Blansche, 2006], notamment en ce qui concerne l'ordre de grandeur, où le choix de l'unité peut avoir une influence majeure et négliger la contribution des autres caractéristiques. Pour pallier ce problème, plusieurs méthodes dont la pondération ont été proposées. La pondération est la recherche d'un ou plusieurs vecteurs de poids qui modifient le degré d'utilisation des attributs. Plusieurs recherches ont montré que la pondération de la distance par un facteur défini en fonction des données peut améliorer les résultats de l'apprentissage non supervisé [Doherty, 2004].

Les données nécessitent parfois d'être traitées avant leur analyse [Lazar, 2008]. En effet, certains attributs peuvent avoir des ordres de grandeur énormes et dépassent celles d'autres attributs qui se trouvent négligeables sans normalisation des données [Karthikeyani et

Visalakshi, 2009]. Celle-ci empêche les attributs de grande portée comme «salaire» de dominer des attributs de petite portée comme l'âge, ou lorsque par exemple la taille est en « cm » et le poids en « kg ». Elle donne ainsi la même importance à toutes les variables. D'où son importance spécialement dans les cas d'utilisation des distances sensibles aux différences des échelles des attributs.

La normalisation est une transformation linéaire des données dans la plage d'un intervalle, généralement [0, 1][Doherty, 2004]. Il existe de nombreuses méthodes pour la normalisation des données dans la littérature [Lazar, 2008][Karthikeyani et Visalakshi, 2009], telles la normalisation par centrage, la normalisation Min-Max, la normalisation du Z-score et la normalisation par échelle décimale.

La normalisation par centrage : elle consiste à soustraire de chaque attribut sa moyenne [Lazar, 2008]:

$$x_{ij}^{*} = x_{ij} - m_j \tag{2.16}$$

Où $m_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$

La normalisation par mise à l'échelle [Lazar, 2008] ou normalisation Min-Max effectue une transformation linéaire sur les données d'origine. Supposons que min_x_i et max_x_i soient les valeurs minimales et maximales pour l'attribut x_i. La normalisation Min-Max est calculée par :

$$x_{ij}^{*} = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \tag{2.17}$$

$$ou \quad x_{ij}^{*} = \frac{x_i}{max(x_i) - min(x_i)} \tag{2.18}$$

L'inconvénient de cette approche est sa sensibilité aux valeurs aberrantes: la distance maximale peut correspondre à un point aberrant et être très importante, perturbant ainsi la normalisation [Lesot, 2009].

La normalisation Z-score ou par centrage et mise à l'échelle : Dans cette normalisation, les valeurs d'un attribut x_i sont normalisées sur la base de la moyenne et de l'écart type de x_i. Une valeur de x_i est normalisée en calculant:

$$x_{ij}^{*} = \frac{x_{ij} - m_j}{\sigma_{x_i}} \tag{2.19}$$

Où $\sigma_{x_i} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - m_j\right)^2}$ est l'écart-type de l'attribut x_i.

Cette méthode de normalisation est utile lorsque le minimum et le maximum de l'attribut x_i sont inconnus ou lorsqu'il existe des valeurs aberrantes qui dominent la normalisation *min-max.*

La normalisation par échelle décimale s'effectue en déplaçant le point décimal des valeurs de l'attribut x_i. Le nombre de décimales déplacées dépend de la valeur absolue maximale de x_i. Une valeur de x_i est normalisée en calculant:

$$x_i^* = \frac{x_i}{10^j} \tag{2.20}$$

Où j est le plus petit entier tel que $Max(\left|x_i^*\right|) < 1$.

Généralement, la normalisation améliore les résultats. Toutefois, on trouve des cas exceptionnels où elle pourrait dégrader la performance. C'est le cas, par exemple, d'un ensemble de classes sphériques avec des centres le long d'une ligne. La normalisation transformerait les classes sphériques en elliptiques, ce qui pourrait être problématique pour certains algorithmes d'apprentissage non supervisé basés sur la distance euclidienne.

Exemple de distances pondérées

Certaines distances sont par définition normalisées. Les plus connues sont la distance de Canberra et la distance de Bray Curtis.

Distance de Canberra

Cette distance a été introduite en 1966 (Lance et Williams 1966). Elle examine la somme de la série d'une fraction des différences entre les coordonnées de deux objets. Elle est très sensible pour les variations proches de zéro et souvent utilisée pour les données éparpillées autour d'une origine et principalement pour des valeurs non négatives. Elle est définie par:

$$d(x_i, x_j) = \sum_{k=1}^{p} \frac{\left|x_{ik} - x_{jk}\right|}{\left|x_{ik}\right| + \left|x_{jk}\right|} \tag{2.21}$$

Distance de Bray Curtis

La distance de Bray-Curtis ou Sorensen est aussi appelée la distance écologique. Elle est considérée comme une méthode de normalisation couramment utilisée dans plusieurs domaines (botanique, écologie, sciences de l'environnement). Toutefois, cette mesure ne satisfait pas l'axiome de l'inégalité triangulaire, ce qui fait d'elle une fausse distance.

$$d(x_i, x_j) = \frac{\sum_{k=1}^{p} |x_{ik} - x_{jk}|}{\sum_{k=1}^{p} (x_{ik} + x_{jk})} \tag{2.22}$$

Gower et Legendre [Gower et Legendre, 1986] ont, de leur part, défini trois autres distances pondérées pour les données positives seulement. Elles sont appelées D_5, D_9 et D_{10} définies respectivement par :

$$D_5(x_i, x_j) = \frac{1}{p} \sum_{k=1}^{p} \frac{(x_{ik} - x_{jk})^2}{(x_{ik} + x_{jk})^2} \tag{2.23}$$

$$D_9(x_i, x_j) = \frac{\sum_{k=1}^{p} |x_{ik} - x_{jk}|}{\sum_{k=1}^{p} \max(x_{ik}, x_{jk})} \tag{2.24}$$

$$D_{10}(x_i, x_j) = \frac{1}{p} \sum_{k=1}^{p} \left(1 - \frac{\min(x_{ik}, x_{jk})}{\max(x_{ik}, x_{jk})}\right) \tag{2.25}$$

2.2 Résultats des tests de distances pondérées

Hattaway avait montré que dans certains cas, l'apprentissage non supervisé peut être amélioré pour des valeurs de $r \geq 0{,}5$. Nous allons présenter dans le premier paragraphe des cas où l'apprentissage est amélioré également pour des valeurs de $r \leq 0{,}5$. Dans le second paragraphe, nous présenterons un nouveau vecteur de pondération pour améliorer également les résultats de l'apprentissage non supervisé.

Les données utilisées pour tester nos approches sont issues des bases réelles mises à disposition par l'Université de Californie à Irvine (UCI machine learning repository) [Blake, 1998] [Asuncion, 2007]. Ces bases sont supervisées, mais aucune information sur les classes n'est donnée à l'algorithme. Ainsi, il est possible de déterminer après l'apprentissage, le

nombre d'objets qui n'appartiennent pas à leur classe correspondante, et ensuite le taux de reconnaissance, calculé par :

$$Tauxde\ reconnaissance = 100 * \frac{Nombre\ d'objets\ correctement\ identifiés}{Nombre\ total\ des\ objets} \quad (2.26)$$

2.2.1 Distances fractionnaires

Il n'existe pas de distance particulière qui satisfait toute sorte d'applications mais certaines mesures sont plus appropriées à certaines situations. Dans ce sens, nous nous attacherons à examiner les distances fractionnaires et déterminer dans quels cas elles peuvent améliorer l'apprentissage.

A cette fin, et pour généraliser les travaux de Hattaway, nous avons mené des expériences avec l'algorithme FCM sur certains ensembles de données avec des valeurs du paramètre *r* inférieures à 0,5.

Descriptif des données

Les ensembles de données utilisés dans ces tests sur les distances fractionnaires sont les suivants: Iris, Wine, BCW, Spect-Heart, Breast-Tissu et Indian.

L'ensemble de données « Iris » qui contient 150 vecteurs d'objets représente trois classes différentes de fleurs (Sesota, versicolour et Virginie). Chaque objet est formé de 4 attributs et chaque classe contient 50 éléments.

L'ensemble de données « Wine » est le résultat d'une analyse chimique des vins de trois cultivateurs différents. Il y a 13 attributs et 178 échantillons provenant de trois classes correspondant à trois cultivateurs différents avec respectivement 59, 79, et 48 échantillons par variété.

L'ensemble de données « BCW » a trait au cancer du sein. Il y a 699 échantillons dont chacun est de dimension 9. Ces échantillons proviennent de la classe maligne (cancéreuse) ou bénigne (non cancéreuse). Les deux classes contiennent respectivement 458 et 241 points.

L'ensemble de données « Spect-heart » décrit le diagnostic des images concernant le « *cardiac Single Proton Emission Computed Tomography (SPECT)* ». Ce sont des images de 267 patients, correspondants à dix images 2D par patient (cinq images pour le repos et cinq

pour l'étude du stress). L'objectif consiste à classer chaque patient dans la catégorie des patients normaux ou anormaux.

L'ensemble de données « Breast-Tissu » est le résultat de mesure du tissu mammaire obtenu par spectroscopie d'impédance électrique. Le nombre d'attributs est 9, celui des échantillons est 106, et celui des classes est six.

L'ensemble de données « Indian » est composé de 583 objets avec 10 attributs. Il y a deux classes: la première avec 416 objets et la seconde contient 167 objets.

Bases de données	***Instances***	***Attributs***	***Classes***
Iris	150	4	3
BCW	699	9	2
Wine	178	13	3
Heart	267	22	2
Breast-Tissu	106	9	6
Indian	583	10	2

Table 2.1 Description des données utilisées

Résultats et discussion

L'algorithme FCM nécessite d'avoir les valeurs de certains paramètres à l'avance. Dans nos tests, ces valeurs sont les suivantes:

- le paramètre m=2.
- Le maximum des itérations= 500.
- $\varepsilon = 10^{-5}$.

Nos expériences consistent à répéter l'algorithme FCM en utilisant différentes valeurs de *r* entre deux limites choisies arbitrairement: 0,01 et 30. Cependant, les meilleurs résultats sont toujours obtenus pour r entre 0,01 et 11 comme le montre la table 2.2.

r	*Iris*	*BCW*	*Heart*	*Wine*	*Breast-Tissu*	*Indian*
0.01	84.67%	91.02%	64.8%	91.02%	45.29%	**55.58%**
0.02	84.67%	91.02%	64.8%	91.02%	43.40%	**55.58%**
0.03	84.67%	91.02%	64.8%	91.02%	44.34%	**55.58%**
0.04	84.67%	91.02%	64.8%	91.02%	44.34%	**55.58%**
0.05	84.67%	91.02%	64.8%	91.02%	42.46%	**55.58%**
0.06	84.67%	91.02%	64.8%	91.02%	43.40%	**55.58%**
0.07	85.34%	91.02%	64.42%	91.02%	40.57%	**55.58%**
0.08	85.34%	91.58%	64.05%	91.58%	44.34%	**55.58%**
0.09	85.34%	92.14%	64.05%	92.14%	**46.23%**	**55.58%**
0.1	85.34%	91.58%	63.68%	91.58%	**46.23%**	55.07%
0.2	86.00%	**93.83%**	63.68%	**93.83%**	33.02%	54.89%
0.3	86.67%	87.65%	63.3%	87.65%	45.29%	50.26%
0.4	86.67%	77.53%	62.18%	77.53%	**46.23%**	55.07%
0.5	88.00%	74.72%	60.3%	74.72%	37.74%	55.41%
0.6	88.67%	71.35%	60.3%	71.35%	38.68%	32.25%
0.7	88.67%	70.23%	59.56%	70.23%	30.19%	31.39%
0.8	88.67%	69.67%	57.18%	69.67%	29.25%	31.05%
0.9	88.67%	69.11%	57.31%	69.11%	30.19%	30.37%
1	88.67%	73.04%	56.18%	73.04%	27.36%	30.37%
2	89.34%	69.67%	58.06%	69.67%	30.19%	30.37%
3	88.67%	69.67%	82.03%	69.67%	31.14%	28.99%
4	88.67%	69.67%	83.15%	69.67%	31.14%	28.99%
5	**89.34%**	69.67%	**84.27%**	69.67%	27.36%	28.99%
6	**89.34%**	69.67%	**84.27%**	69.67%	31.14%	28.99%
7	**89.34%**	69.67%	**84.27%**	69.67%	26.42%	28.99%
8	**89.34%**	69.67%	**84.27%**	69.67%	26.42%	28.99%
9	88.67%	69.67%	70.04%	69.67%	31.14%	28.99%
10	88.67%	69.67%	57.31%	69.67%	32.08%	28.99%
11	88.67%	69.67%	50.57%	69.67%	39.63%	28.99%
Chybechev	88.67%	69.67%	52.06%	69.67%	27.36%	28.99%
Canberra	94.67%	94.95%	*	94.95%	52.84%	*
BrayCurtis	88%	71.92%	64.05%	71.92%	46.23%	*

Table 2.2 Taux de reconnaissance des distances usuelles et la métrique L_r avec différentes valeurs de *r*.

Nous avons pensé à chercher s'il existe une relation entre la valeur $\boldsymbol{r_{opt}}$ et la dimension des données. Les résultats obtenus montrent qu'il n'existe pas de relation dans ce sens. Par exemple, Breast-Tissu et BCW ont le même nombre des attributs (9 attributs) mais les meilleurs résultats sont obtenus pour Breast-Tissu avec $r_{opt} = 0.1$ et $r_{opt} = 0.4$, et pour BCW avec $r_{opt} = \infty$ (la distance de Chebyshev ou L_∞).

Nous avons également remarqué qu'il n'y a pas d'influence du nombre de classes et de la corrélation des variables sur r_{opt}. En effet, les meilleurs résultats sont comme suit :

- Iris et Wine ont tous les deux 3 classes, mais pour l'ensemble des données Iris, r_{opt} est égale à 5, 6, 7 et 8, alors que pour l'ensemble des données Wine, $r_{opt} = 0.2$.

- Les deux ensembles de données Indian et Breast-Tissu ont de meilleurs résultats pour $r < 1$, mais la corrélation est moyenne pour le premier jeu de données et grande pour le deuxième ensemble de données.

Notons par ailleurs, que les ensembles Wine, Indian et Breast-Tissu contiennent des valeurs aberrantes, alors que les ensembles Iris, BCW et Heart ne les contiennent pas (Figure 2.1).

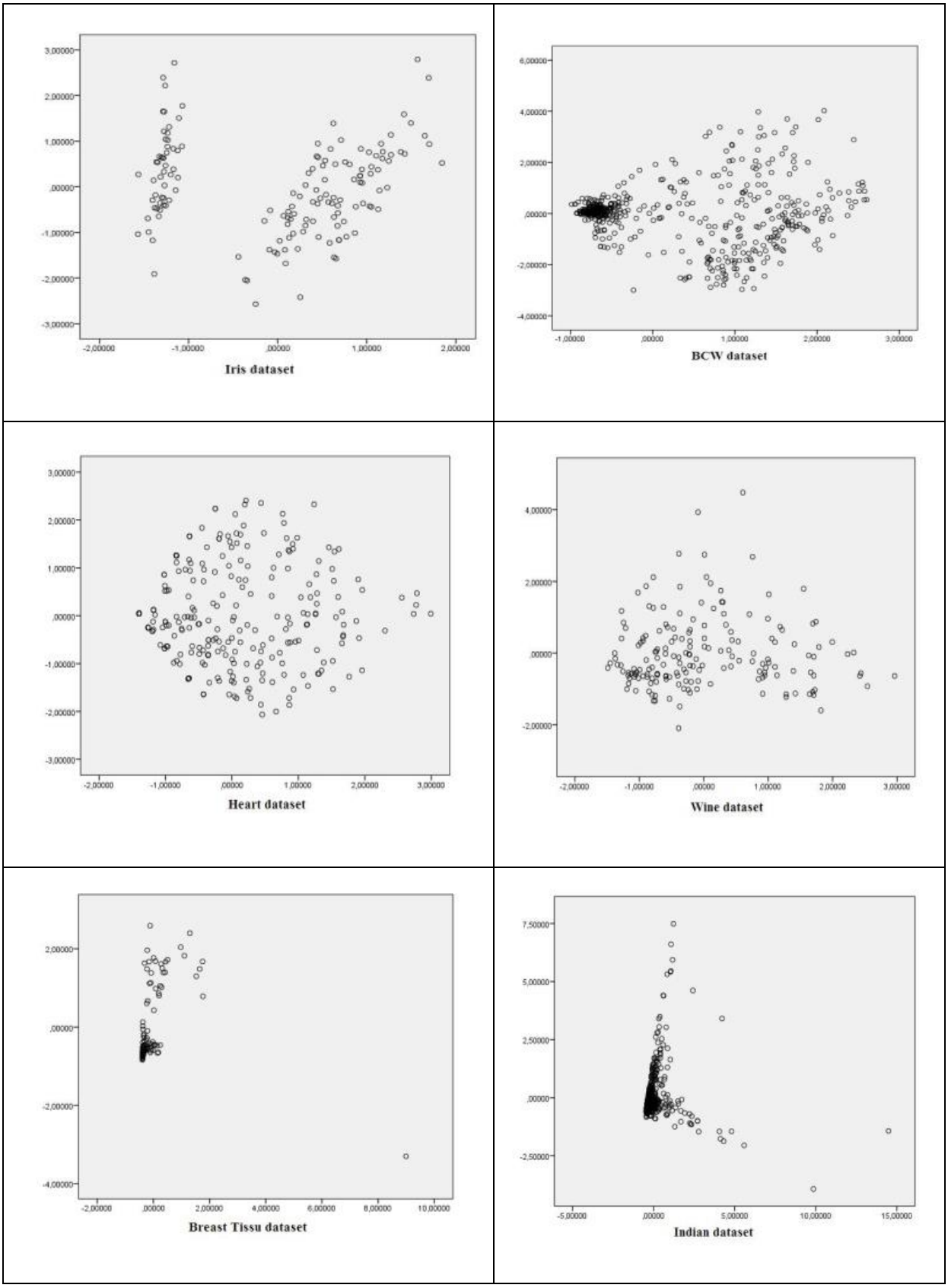


Figure 2.1 Representation des données de diférentes bases avec la méthode d'analyse en composante principale (ACP)

La table 2.2 qui représente le taux de reconnaissance des objets montre que pour $r \geqslant 1$, les bons résultats ont été obtenus pour les ensembles de données ne contenant pas de valeurs aberrantes, alors que les valeurs de $r < 1$ donnent de bons résultats quand il y a des valeurs aberrantes. Cela étend les résultats précédents [Doherty, 2004] au contexte de l'apprentissage non supervisé flou.

Nos résultats confirment également le constat sur la limite de la distance euclidienne pour résoudre les problèmes de l'apprentissage non supervisé [Rand, 1971]. En effet, pour les six ensembles de données considérés, la distance euclidienne n'a donné de bons résultats que pour l'ensemble des Iris. Cet ensemble est caractérisé par ses trois classes sphériques. Pour les cinq autres ensembles de données considérés, les résultats de l'algorithme FCM utilisant la distance euclidienne sont très faibles par rapport aux autres distances. En plus, ce meilleur taux de reconnaissance obtenu pour la distance euclidienne dans le cas des Iris, et qui est 89,34%, est le même pour les valeurs de r égale à 5, 6, 7 et 8. La table 2.3 présente une synthèse des résultats obtenus, en mentionnant la valeur de r_{opt}.

Bases de données	***Meilleur taux de reconnaissance (r_{opt})***	***Faible taux de reconnaissance (r_{opt})***	***Manhattan***	***Euclidian***	***Chybechev***
Iris	89.34% $r_{opt} \in \{2, 5, 6, 7, 8\}$	84.67% $r_{opt} \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$	88.67%	89.34%	88.67%
BCW	96.71% $r_{opt} = 11$	88.7% $r_{opt} \in \{0.03, 0.04\}$	94.14%	95.43%	**97%**
Heart	**84.27%** $r_{opt} \in \{5, 6, 7, 8\}$	50.19% $r_{opt} = 12$	56.18%	58.06%	52.06%
Wine	**93.83**% $r_{opt} = 0.2$	69.11% $r_{opt} = 0.9$	73.04%	69.67%	69.67%
Breast-Tissu	**46.23**% $r_{opt} \in \{0.09, 0.1, 0.2, 0.3, 0.4\}$	22.65% $r_{opt} = 14$	27.36%	30.19%	27.36%
Indian	**55.58%** $r_{opt} \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$	28.99% $r_{opt} \in \{3,4,5,6,7,8,9,10,11,12, 13,14,15\}$	30.37%	30.37%	28.99%

Table 2.3 Taux de reconnaissance du FCM utilisant L_r métrique sur les données utilisées pour les tests.

Les résultats présentés ci-dessus montrent que les valeurs du paramètre *r* inférieures à 1 peuvent améliorer considérablement les performances de l'apprentissage, en particulier lorsque l'ensemble de données contient des valeurs aberrantes (Breast-tissu, Indian et Wine).

2.2.2 Distances pondérées proposées

Définition

La mesure proposée est basée sur une pondération des dimensions de l'espace de représentation par le biais d'un facteur *v*. L'idée sous-jacente est de minimiser les effets dus à une dimension qui discrimine certains objets. Dans ce sens, et inspiré d'autres travaux [Bouroumi, 2000] [Bandemer, 1992] [Digby, 1987], nous avons proposé de pondérer les distances usuelles en introduisant le facteur suivant :

$$v_j = \max_{1 \le i \le n}(x_{ij}) - \min_{1 \le i \le n}(x_{ij}) \tag{2.27}$$

Comme la distance de Minkowski peut être calculée grâce à la forme quadratique suivante:

$$d(x_k, x_i) = \sqrt{(x_k - x_i)^T A (x_k - x_i)} \tag{2.28}$$

Où *A* est une matrice *pxp* définie positive, la nouvelle distance, « d_w », est définie alors par :

$$d_w(x_k, x_i) = \sqrt{(x_k - x_i)^T A * Q \ \ (x_k - x_i)} \tag{2.29}$$

Où *Q* est une matrice *pxp* définie positive, et définie par :

$$Q_{ij} = \begin{cases} (v_j)^{-2}, & i = j \\ 0, & \text{sinon} \end{cases} \tag{2.30}$$

Selon les équations (2.11) et (2.29), la version pondérée de la distance est calculée comme suit:

$$d_w(x, y) = \left(\sum_{j=1}^{p} \left| \frac{x_j - y_j}{v_j} \right|^N \right)^{\frac{1}{N}} \tag{2.31}$$

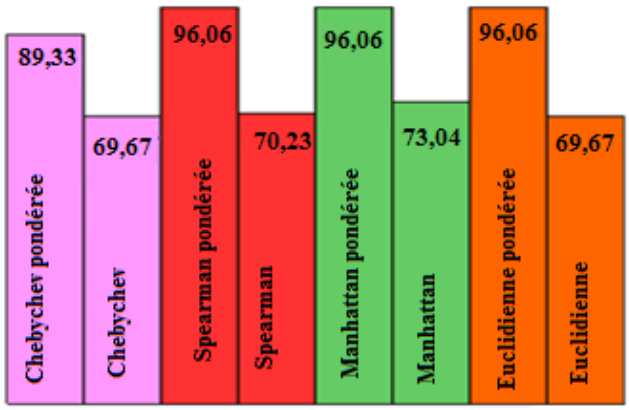
Résultats et discussion

Pour évaluer la performance des distances pondérées proposées, nous avons effectué des expériences sur trois ensembles de données du monde réel disponibles à partir de l'Université de Californie à Irvine (UCI): Wine, Spect-heart et Breast-Tissu.

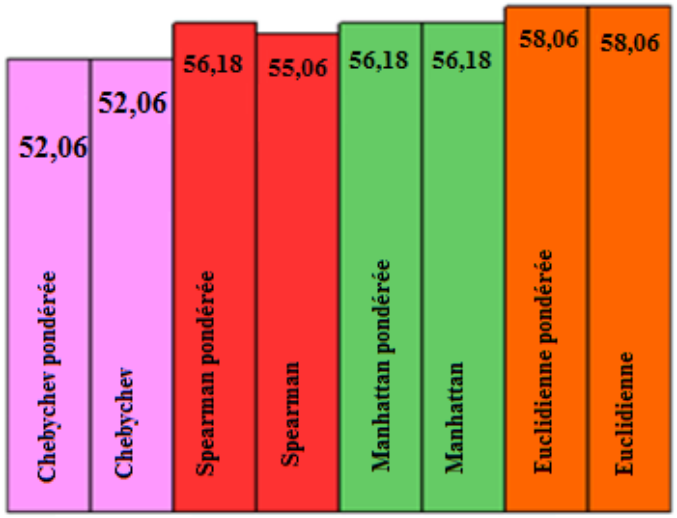
Nous avons comparé les distances pondérées et non pondérées via l'algorithme FCM. Les résultats de l'apprentissage non supervisé, pour les deux distances pondérées et non pondérées, sont présentés dans la table 2.4 et schématisés dans la figure 2.2.

Données	***Distances***							
	Euclidienne	***Euclidienne pondérée***	***Manhattan***	***Manhattan pondérée***	***Spearman***	***Spearman pondérée***	***Chebyshev***	***Chebyshev pondérée***
Wine	69.67%	**96.06%**	73.04%	**96.06%**	70.23%	**96.06%**	69.67%	**89.33%**
Heart	58.06%	58.06%	56.18%	56.18%	55.06%	**56.18%**	52.06%	52.06%
Breast Tissue	30.13%	**31.14%**	24.53%	**63.21%**	29.25%	**39.63%**	27.36%	**49.06%**

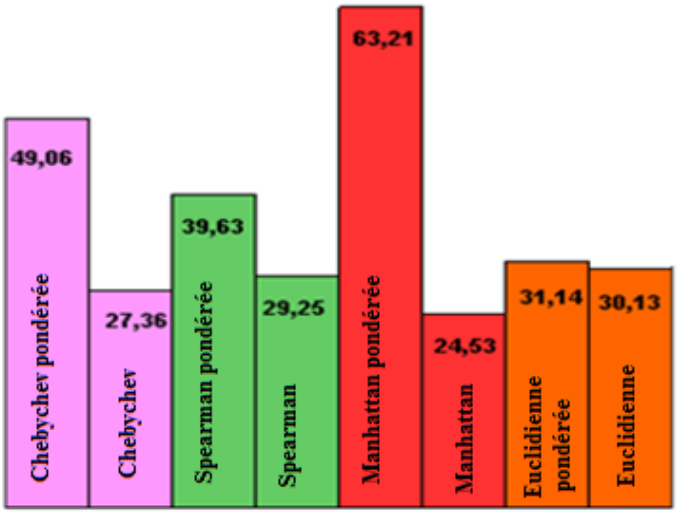
Table 2.4 Taux de reconnaissance des distances pondérées et non pondérées.



a) Taux de reconnaissance pour la base de données Wine



b) Taux de reconnaissance pour la base de données Spect-heart



c) Taux de reconnaissance pour la base de données Breast Tissue

Figure 2.2 Taux de reconnaissance pour les ensembles de données Wine (a), Heart(b) et Breast –Tissu(c).

Les résultats ont montré une amélioration de l'apprentissage non supervisé avec les distances pondérées pour les bases de données Wine et Breast-Tissu. Cependant, il n'y avait aucune amélioration significative pour l'ensemble de données Spect-Heart. Pour ce dernier ensemble de données, une analyse comparative des attributs a montré que chaque attribut prend une valeur de la paire {0, 1}. Les données ne présentent pas donc de variations importantes et il n'y a pas de différence d'échelles. Les résultats de l'apprentissage n'ont pas été alors améliorés.

A signaler que la normalisation n'améliore pas toujours les résultats de l'apprentissage. Elle pourrait même dégrader les performances. C'est le cas par exemple d'un ensemble formé de classes sphériques dont les prototypes sont le long d'une ligne. La normalisation transforme les classes sphériques en elliptiques, ce qui pourrait poser problème pour certains algorithmes d'apprentissage non supervisé flou, notamment l'algorithme FCM.

En résumé, les résultats expérimentaux montrent que cette pondération améliore les résultats de l'apprentissage parce que la pondération compense la différence du poids entre les variables et le choix des unités de mesure des caractéristiques. Elle permet également de minimiser les effets des valeurs aberrantes qui faussent les résultats de l'apprentissage.

Conclusion

La distance a un impact crucial sur les résultats de l'apprentissage non spervisé. Ceci s'explique par le fait que chaque distance implique une vision différente des données en raison de sa géométrie. Le choix de la distance ne doit pas être alors arbitraire, et doit être défini en fonction des besoins réels. En plus, l'utilisation de la métrique L_r peut améliorer les résultats de l'apprentissage si le paramètre *r* est convenablement ajusté pour lutter contre le phénomène de concentration. Par ailleurs, pondérer les distances peut améliorer les résultats, et doit être alors considéré lors des résolutions des problèmes de l'apprentissage.

L'étude effectuée a donné des résultats encourageants, bien qu'ils ne soient pas parfaits, et a confirmé le constat sur les mesures de similarités qui est l'inexistence de méthode pour choisir la métrique optimale dans les problèmes de l'apprentissage non supervisé. Nous conseillons lors de telle tâche de normaliser les données ou de normaliser les distances, et de tester les algorithmes d'apprentissage avec plusieurs distances avant de décider du choix de la métrique optimale.

CHAPITRE 3

NOUVELLE APPROCHE POUR LA REDUCTION DE L'IMPACT DU CHEVAUCHEMENT DES CLASSES

Introduction

Dans le monde réel, on se trouve souvent confronté à des situations où les informations disponibles ne sont pas toujours précises. Ces situations d'imprécisions sont courantes en apprentissage non supervisé flou qui permet l'appartenance graduelle pour pouvoir tenir compte de ces imprécisions et prendre de décision. Toutefois, il s'avère parfois difficile de prendre de décision, notamment lorsqu'un objet ressemble autant aux objets d'une classe qu'aux objets d'une autre et se trouve ainsi à la frontière de deux ou plusieurs classes, ce qui réduit parfois l'efficacité de l'apprentissage. Cette réduction est proportionnelle au degré de chevauchement entre les classes. Pour pallier ce problème, on préconise d'utiliser des connaissances lors du processus d'apprentissage. Ces connaissances sont utilisées pour améliorer les résultats en les rendant plus précis, plus robustes et en fournissant à l'expert une information plus riche et plus pertinente [Forestier, 2010]. L'intégration des connaissances peut être dirigée telle dans les approches *semi-supervisées* ou les approches dites *avec contraintes*. Or, ce type de connaissances n'est pas facilement disponible. De ce fait, intégrer dans le processus d'apprentissage flou non supervisé, au fur et à mesure, des connaissances produites par l'algorithme à chaque itération, semble bien fondé et peut réduire l'impact du chevauchement des classes sur les résultats. C'est le but de la nouvelle approche dynamique d'apprentissage non supervisé flou, appelée IUFL (*Improved unsupervised fuzzy learning*), proposée dans ce chapitre.

3.1 L'approche proposée

En classification, les données considérées proviennent d'au moins deux classes différentes. Partant de ce constat, le processus d'apprentissage est lancé en créant deux classes autour des objets les moins similaires. Pour cela, les points représentatifs ou prototypes des classes

créées, v_1 *et* v_2 sont initialisés à l'aide de ces deux objets après les avoir échangés avec les deux premiers objets x_1 et x_2.

Les (*n - 2*) objets restants sont ensuite successivement examinés de manière séquentielle en analysant leurs similarités avec les points ou prototypes qui représentent les classes découvertes auparavant. Une nouvelle règle d'apprentissage est utilisée pour les objets x_i restants ($3 \leq i \leq n$). Son principe est le suivant :

(1) créer une nouvelle classe autour de cet objet,

(2) ou utiliser les informations acheminées par l'objet pour l'affecter à une classe existante et mettre à jour les points représentatifs des classes,

(3) ou différer et reporter l'examen de cet objet jusqu'à ce que l'une des deux décisions précédentes puisse être prise avec suffisamment de confiance. La méthode est dynamique en ce que la règle de décision dépend du nombre de classes *c*, qui varie au cours du processus d'apprentissage.

Le choix de la meilleure décision à prendre est basé sur deux critères quantitatifs:

- *Le premier critère* est la valeur maximale des similarités calculées entre x_i et les *c* prototypes existants:

$$Max_1(x_i) = \underbrace{Max}_{1 \leq k \leq c}(Sim(i,k)) \tag{3.1}$$

Notons P_i le prototype correspondant.

- *Le deuxième critère* est la différence entre $Max_1(x_i)$ et le degré de similarité de x_i au second prototype le plus proche:

$$Max_1(x_i) - Max_2(x_i) \tag{3.2}$$

avec :

$$Max_2(x_i) = \underset{\substack{1 \leq k \leq c \\ v_k \neq P_i}}{Max}(Sim(i,k)) \tag{3.3}$$

La détermination de ces deux critères permet de prendre l'une des décisions suivantes :

- La première décision consiste à créer une nouvelle classe et initialiser son prototype par l'objet x_i. Cette décision est prise lorsque $Max_1(x_i)$ est inférieur à un seuil ξ, défini par l'utilisateur, et dont les valeurs peuvent théoriquement varier entre les limites :

$$\xi_{\min} = \underbrace{Min}_{\substack{1 \le i,j \le n \\ i \ne j}} (Sim(i,j)) \tag{3.4}$$

$$\xi_{max} = \underbrace{Max}_{\substack{1 \le i,j \le n \\ i \ne j}} (Sim(i,j)) \tag{3.5}$$

Cette décision signifie que x_i ne ressemble pas assez aux prototypes des classes précédemment détectées afin de considérer qu'il provient d'une de ces classes. Par conséquent, une nouvelle classe est créée autour de x_i.

– La seconde décision consiste à reporter l'exploration de x_i jusqu'à ce que d'autres objets soient examinés. Cette décision est prise si les deux conditions suivantes sont respectées :

$$Max_1(x_i) \ge \xi \tag{3.6}$$

$$Max_1(x_i) - Max_2(x_i) < \frac{1}{c+1} \tag{3.7}$$

La condition $Max_1(x_i) \ge \xi$ signifie que le degré de similarité que l'objet x_i présente avec les prototypes existants est suffisante pour considérer qu'il provient de l'un des c classes déjà découvertes.

La deuxième condition reflète cependant, le fait qu'il existe une grande ambiguïté en ce qui concerne la classe d'où peut provenir x_i . Cette ambiguïté dépend du nombre c des classes existantes et s'accroit lorsque x_i présente le même niveau de similarité avec tous les prototypes des classes détectées jusqu'à lors.

$\frac{1}{c+1}$ est le seuil que devrait dépasser la différence entre les degrés d'appartenance de l'objet x_i aux classes correspondant à ses deux prototypes les plus similaires, pour que le processus d'apprentissage prenne en compte l'information portée par cet objet. Dans le cas contraire, l'exploration de cet objet est reportée jusqu'à ce que les objets suivants, le cas échéant, soient explorés, ce qui peut aider à réduire l'ambiguïté observée. Notons que la mesure de similarité $Sim(x_i,v_k)$ entre un objet x_i et un prototype v_k peut également être interprétée comme le degré d'appartenance de x_i à la classe représentée par v_k [Bouroumi, 2000].

– La troisième décision consiste à exploiter l'information portée par x_i afin de mettre à jour les prototypes des classes existantes selon la règle d'apprentissage donnée par l'équation suivante :

$$v_k(i) = v_k(i-1) + \frac{Sim(x_i, v_k)}{n_k(i)}\left[x_i - v_k(i-1)\right] \quad 1 \le k \le c, \quad c \ge 2 \tag{3.8}$$

où $v_k(i)$, $v_k(i\text{-}1)$ sont respectivement les prototypes de la $k^{éme}$ classe avant et après l'ajout de x_i, et $n_i(k)$ désigne le cardinal flou de la $k^{ème}$ classe après l'ajout de x_i, défini par:

$$n_i(k) = \sum_{j=1}^{k} Sim(x_i, v_k) \qquad 1 \le k \le c, \;\; i \le n \tag{3.9}$$

Cette troisième décision est prise quand $Max_1(x_i) - Max_2(x_i)$ est plus grande ou égale au seuil $\frac{1}{c+1}$, et la valeur maximale des similarités calculées est supérieure ou égale au seuil ξ.

Théoriquement parlant, un objet peut ne pas porter assez d'informations et ainsi sera rejeté tant que la condition requise pour qu'il soit traité n'est pas respectée. Pour cela, l'utilisateur a la possibilité de fixer le nombre de fois que les objets seront rejetés.

Enfin, en répétant ce procédé pour les mêmes données d'entrée en utilisant différentes valeurs du seuil de similarité ξ, on peut obtenir différentes solutions plus ou moins acceptables. Pour sélectionner le meilleur résultat parmi toutes ces solutions candidates, deux critères de validité sont utilisés [Bouroumi, 2000] [Benrabh, 2005] [Rezaee, 1998] [Halkidi, 2001], à savoir l'entropie [ ] et le coefficient de partition [ ].

L'entropie est définie par:

$$PE(U) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c}\left[u_{ik}\log_a(u_{ik})\right] \tag{3.10}$$

et le coefficient de partition modifié par Dave et défini par:

$$PC(U) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c}(u_{ik})^2 \tag{3.11}$$

La partition la plus proche de la réalité correspond à une entropie minimale et à un coefficient de partition maximal.

En termes de pseudo-code, la méthode proposée est décrite par l'algorithme 3.1.

Algorithme 3.1 : Algorithme IUFL

Entrées: Ensemble de données non étiquetées $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$;

Sorties : Nombre estimé de classes c^*, matrice de prototypes $V= (v_1,\ldots,v_{c^*})$.

Début

Trouver les deux objets les moins similaires, x_i et x_j
Echanger(x_1, x_i)
Echanger(x_2 , x_j)
Calculer: $\xi_{min} \leftarrow \underset{\substack{1\le i,j\le n \\ i\neq j}}{Min}(Sim(i,j))$, $\xi_{max} \leftarrow \underset{\substack{1\le i,j\le n \\ i\neq j}}{Max}(Sim(i,j))$, $\Delta\xi \leftarrow 0{,}1$

Pour ξ allant de $\xi_{\min}$ à $\xi_{\max}$ avec un pas $\Delta\xi$ ***faire***

c ←2
$v_1 \leftarrow x_1$
$v_2 \leftarrow x_2$
Objects_Traités ← 2 // Nombre des objets traités
Pour *chaque* objet x_i non traité ***faire***
Si l'objet x_i n'est pas rejeté 2 fois ***alors***
Calculer les 2 similarités maximales de x_i avec les prototypes existants

$$Max_1(x_i) \leftarrow \underbrace{Max}_{1\le k\le c}(Sim(i,k)),\ Max_2(x_i) \leftarrow \underset{\substack{1\le k\le c \\ v_k\neq P_i}}{Max}(Sim(i,k))$$

Si ($Max_1 < \xi$) ***alors***
Créer une classe
$c \leftarrow c+1$
$v_c \leftarrow x_i$ // le prototype de la nouvelle classe
Marquer l'objet x_i traité
Incrémenter le nombre des objets traités
Sinon
Si $(Max_1(x_i) - Max_2(x_i)) \geq \frac{1}{c+1}$ ***alors***
Mettre à jour les prototypes selon (Eq.3.8)
Marquer l'objet x_i traité
Incrémenter le nombre des objets traités
Sinon *Rejeter* x_i
Finsi
Finsi
Sinon
Mettre à jour les prototypes selon (Eq.3.8)
Marquer l'objet x_i traité
Incrémenter le nombre des objets traités.
Finsi
Finpour
Calculer PE(U)
Calculer PC(U)
Fin pour

Fin

3.2 Résultats et discussions

3.2.1 Présentation des données

Pour évaluer les performances de l'algorithme proposé, des expériences ont été menées sur quatre ensembles de données disponibles sur le site UCI [Blake, 1998]: Wine, Breast Cancer, Balance scale and Haberman's Survival (voir la table 3.1). Une segmentation de deux images IRM du cerveau est également présentée.

Données numériques

Le premier et deuxième ensemble de données Wine et Breast Cancer ont été présentés dans le chapitre 2. Le troisième ensemble de données est « Balance scale ». Il contient 625 objets de données. Chaque objet dispose de 4 attributs: poids gauche, la distance gauche, le poids à droite, et la distance droite. Ces objets forment 3 classes caractérisées comme suit: 49 échantillons équilibrés (B), 288 gauches (L) et 288 droites (R).

Le dernier exemple est « Haberman's Survival » qui est le résultat d'une mesure de 306 cas de survie de patients ayant subi une chirurgie du cancer du sein. Les objets ont 3 attributs : l'âge du patient au moment de l'opération, l'année de son opération et le nombre de nœuds axillaires positifs détectés. Ces données sont distribuées en deux classes: la première classe est celle des patients ayant survécu 5 ans ou plus, la seconde est celle des patients décédés dans les 5 ans de l'opération.

La table 3.1 résume les données considérées et donne des informations sur les attributs, la taille et le nombre de classes.

	BCW	***Wine***	***Balance***	***Haberman's Survival***
Nombre des données	699	178	625	306
Nombre des attributs	9	13	4	3
Nombre des classes	2	3	3	2

Table 3.1 Description des données.

Images IRM

L'imagerie par résonance magnétique (I.R.M.) est une technique d'imagerie médicale utilisée pour faire un diagnostic qui se fonde sur les principes de la résonance magnétique nucléaire. En analysant ces images, le médecin peut identifier et suivre la progression des pathologies. Cependant, l'analyse « *manuelle* » des images est longue et pénible. Ainsi, de nombreuses recherches ont été menées afin d'automatiser l'analyse de ces images, en l'occurrence la segmentation [Naveen, 2018]. Celle-ci permet d'analyser ces images et de prendre une décision finale en procédant à une séparation des différentes zones homogènes d'une image.

Il existe de nombreuses méthodes de segmentation et chacune d'elles est fortement liée à un domaine d'application [Nadernejad, 2011]. Dans ce travail, on s'intéresse à la segmentation basée sur le clustering. Nous considérons deux images IRM (figure 3.1) et pour chacune d'elles, nous procédons à segmenter l'encéphale formé de trois tissus cérébraux: la matière blanche (MB), la matière grise (MG) et le liquide céphalo-rachidien(LCR).

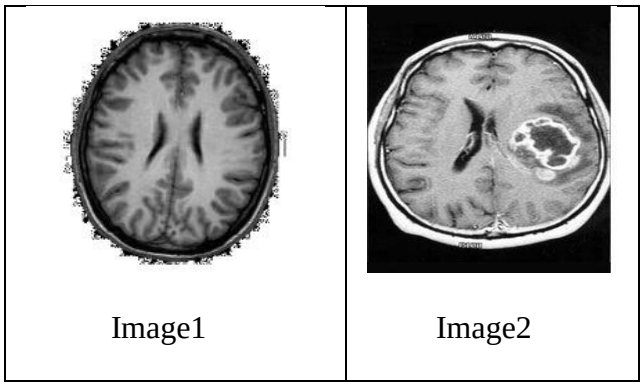


Figure 3.1 Images IRM du cerveau

Mais avant d'entamer la segmentation, on a procédé à une extraction des données de ces images. Ainsi, chaque image est représentée par une matrice où chaque ligne représente un pixel.

Après avoir effectué l'apprentissage non supervisé de ces deux images par les méthodes considérées et obtenu pour chaque pixel la classe à laquelle il appartient, on a procédé à la reconstruction de l'image en donnant une même couleur à tous les pixels de la même classe.

3.2.2 Détection du nombre de classes

Tout d'abord, la procédure de détection des classes est exécutée pour les valeurs de ξ comprises entre $\xi_{\min} = \underset{\substack{1\le i,j\le n \\ i\neq j}}{Min}(Sim(i,j))$ et $\xi_{\max} = \underset{\substack{1\le i,j\le n \\ i\neq j}}{Max}(Sim(i,j))$, avec un pas de $\Delta\xi = 0{,}1$.

En faisant varier le seuil ξ, différentes partitions de *c* classes peuvent être détectées. Le rôle de cette première exploration est de montrer comment le nombre de classes détectées varie avec ξ. La table 3.2 montre les valeurs de *c* obtenues avec un pas $\Delta\xi = 0{,}1$ pour les données de BCW.

ξ	***c***
0,2131	2
0,3131	10

Table 3.2 - Variation de *c* avec un pas $\Delta\xi = 0{,}1$ pour les données de BCW

On remarque que dans le cas de la base de données BCW, certaines valeurs de c ne sont pas obtenues avec le pas considéré ($\Delta\xi = 0{,}1$). Pour obtenir ces valeurs, l'ensemble de données considéré a été exploré avec un pas $\Delta\xi = 0{,}001$. En fait, plus le pas du seuil est petit, plus la durée d'exploration est élevée, alors que s'il est grand, la structuration des classes est plus rapide, mais certaines valeurs de c ne seront pas obtenues. La table 3.3 résume les résultats de l'exploration effectuée.

ξ	***c***
[0,1231 – 0,2161]	2
0,21 71	3
[0,2181 – 0,2311]	4
[0,2321 – 0,2411]	5
[0,2421 – 0,2521]	6

Table 3.3 Variation de *c* avec le pas $\Delta\xi = 0{,}001$ pour les données BCW

Toute valeur de l'intervalle permet d'avoir le nombre adéquat de classe. On choisit d'utiliser la limite inférieure. On constate que les plus petites valeurs de seuil ξ qui ont conduit à un

certain nombre de classes comprises entre 2 et 6 pour chaque ensemble de données. Le signe * signifie que l'on n'a pas détecté les nombres *c=2* ou *c=6* malgré que l'on a varié le pas d'exploration en lui donnant plusieurs valeurs.

c	***BCW***	***Wine***	***Balance***	***Haberman's Survival***
2	0.1231	0.2219	*	0.3983
3	0.2171	0.2269	0.3333	0.4383
4	0.2181	0.2299	0.3353	0.4483
5	0.2321	0.2339	0.3473	0.4783
6	0.2421	0.2359	*	0.4883

Table 3.4. Valeur minimale du seuil ξ pour détecter les nombres de classes c ($2 \leq c \leq 6$).

Les deux indices de validité, l'entropie et le coefficient de partition, sont utilisés pour choisir la meilleure partition parmi l'ensemble de partitions. Nous pouvons voir que la meilleure solution correspond toujours au nombre réel des classes pour les quatre exemples considérés. Les valeurs optimales sont affichées en gras comme le montre la table 3.5.

c		***2***	***3***	***4***	***5***	***6***
BCW	*E*	**0.385**	0.511	0.536	0.575	0.627
	PC	**0.832**	0.694	0.627	0.562	0.492
Wine	*E*	0.971	**0.604**	0.647	0.770	0.809
	PC	0.519	**0.556**	0.525	0.371	0.305
Balance	*E*	*	**0.889**	0.926	0.940	*
	PC	*	**0.419**	0.306	0.239	*
Haberman's Survival	*E*	**0.965**	0.983	0.989	0.992	0.996
	PC	**0.523**	0.346	0.258	0.205	0.169

Table 3.5 Valeurs des indices de validité pour chaque classe c détectée ($2 \leq c \leq 6$).

Les deux algorithmes AFNS et IUFL déterminent le nombre de classes *c* présentes dans l'ensemble de données. Nous avons exécuté les deux algorithmes 30 fois après avoir changé de manière aléatoire l'ordre des éléments. La valeur optimale de *c* correspond au minimum de l'entropie *PE(U)* et au maximum du coefficient de partition *PC(U)*. La table suivante présente les résultats de cette exploration:

Algorithme		***BCW***	***Wine***	***Balance***	***Haberman***
AFNS	Nombre de fois que c est correct	10	4	6	14
	Nombre de fois que c est incorrect	20	26	24	16
IUFL	Nombre de fois que c est correct	24	23	25	23
	Nombre de fois que c est incorrect	6	7	5	7

Table 3.6 Comparaison de *c* détecté par AFNS et IUFL.

La table 3.6 montre que le nombre de fois où l'algorithme a déduit la bonne valeur de *c* est bien plus grand que dans le cas de l'algorithme AFNS. Ceci est dû principalement à la possibilité de l'algorithme IUFL à réduire l'impact de l'ordre des éléments, et ce grâce à sa faculté de « rejeter » les « mauvais » éléments.

3.2.3 Résultats de l'apprentissage

Les bases de données considérées sont supervisées, mais aucune information sur les classes n'est fournie à l'algorithme IUFL. Ainsi, il est possible de déterminer à la fois le nombre d'objets mal classés et le taux de reconnaissance.

Les expériences ci-dessous visent à illustrer l'efficacité de l'algorithme IUFL en comparaison avec les algorithmes suivants: K-means, FCM, PCM, ISODATA et AFNS. Ces algorithmes attribuent les objets aux classes par le principe d'attribution de distance minimale. Ce principe consiste à attribuer un nouvel objet x_i à la classe pour laquelle il a le plus grand degré d'appartenance.

Nous avons implémenté ces algorithmes pour tester leur efficacité. L'algorithme FCM est utilisé avec les normes habituelles pour partitionner les données. Il s'agit des normes suivantes: euclidienne, Spearman, Manhattan et Tchebychev, qui sont des cas particuliers de la distance Minkowski.

Pour évaluer les résultats obtenus, nous avons procédé à une première comparaison basée sur le taux de reconnaissance. Pour les bases de données médicales ayant deux classes, nous avons réalisé 3 autres comparaisons: la précision, la sensibilité et la spécificité, définis respectivement par :

$$\text{Pr}écision = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.12}$$

$$Sensibilité = \frac{VP}{VP + FN} \tag{3.13}$$

$$Specificité = \frac{VN}{VN + FP} \tag{3.14}$$

Où:

VP (Vrais positifs) est le nombre d'objets correctement identifiés ;

FP (Faux positifs) est le nombre d'objets non correctement identifiés ;

VN (Vrais négatifs) est le nombre d'objets correctement rejetés ;

FN (Faux négatifs) est le nombre d'objets non correctement rejetés.

La sensibilité est la capacité d'identifier correctement les patients malades. Elle correspond à la proportion de vrais positifs par rapport à l'ensemble des structures qui devraient être apprises. Elle tend vers 1 (respectivement 0) s'il y a peu, respectivement beaucoup, de faux négatifs.

La spécificité est la capacité d'identifier correctement les patients non malades. Elle correspond à la proportion de vrais négatifs par rapport à l'ensemble des structures qui ne devraient pas être apprises. La spécificité tend vers 1 (respectivement 0) s'il y a peu (respectivement beaucoup) de faux positifs.

Les autres algorithmes ont été appliqués aux données et leurs performances ont été comparées. La table 3.7 résume les résultats obtenus.

Données	*FCM*	*K-means*	*ISODATA*	*PCM*	*UFL*	*I UFL*
BCW	65,96%	**66,1%**	65,52%	61.52%	64,95%	**66,1%**
Wine	69.67%	69.67%	66.30%	66.86%	69,67%	**71.92%**
Balance	53.92%	48.16%	50.4%	50.72%	48,96%	**66.4%**
Haberman's Survival	49.02%	49.02%	49.02%	50.33%	52,95%	**53.93%**

Table 3.7 Taux de reconnaissance des algorithmes étudiés appliqués aux données étudiées.

La table 3.7 montre que l'algorithme IUFL peut considérablement améliorer les performances de l'apprentissage. Il peut également augmenter la précision de la découverte des classes comme le montre la table 3.8.

Ainsi, la précision de l'algorithme proposé dans le cas de la base de données BCW est de 0,93 alors qu'elle n'est que 0,66 pour FCM.

Dans le cas de la base de données Haberman's Survival, l'algorithme IUFFL présente une légère amélioration de l'apprentissage par rapport aux autres algorithmes considérés, sauf pour l'algorithme K-means qui présente une bonne précision que celle obtenue par l'algorithme proposé.

Données	*FCM*	*K-means*	*ISODATA*	*PCM*	*UFL*	*I UFL*
BCW	0,66	0,66	0,82	0,65	0,66	**0,93**
Haberman's Survival	0,49	**0,73**	0,53	0,49	0,53	0,54

Table 3.8 Les valeurs de la précision calculée pour les différentes données.

En ce qui concerne les performances sur le plan de la sensibilité, l'algorithme IUFL permet d'obtenir de bons résultats en comparaison avec les autres algorithmes. Il présente une sensibilité de 0,98 (Table 3.9) qui est très proche de 1 dans le cas de l'ensemble de données BCW, c'est-à-dire que l'algorithme IUFL produit peu de faux négatifs.

Données	*FCM*	*K-means*	*ISODATA*	*PCM*	*UFL*	*I UFL*
BCW	0,71	0,75	0,96	0,43	0,79	**0,98**
Haberman's Survival	0,28	0,73	0,23	0,22	0,26	**0,75**

Table 3.9 Les valeurs de la sensibilité calculée pour les différents algorithmes.

Sur le plan de la spécificité, IUFL améliore les résultats pour l'ensemble BCW, mais reste faible pour la base de données Haberman's Survival comme le montre la table 3.10.

Données	*FCM*	*K-means*	*ISODATA*	*PCM*	*UFL*	*I UFL*
BCW	0,02	0,03	0,50	0,05	0,30	**0,81**
Haberman's Survival	0,57	**0,76**	0,36	0,37	0,43	0,42

Table 3.10 Les valeurs de la spécificité calculée pour les différentes données.

Les tables suivantes (table 3.11, 3.12, 3.13 et 3.14) donnent les centres réels calculés à partir des données étiquetées à l'origine, et les prototypes détectés lors de l'apprentissage par les algorithmes étudiés. On constate que l'algorithme IUFL fournit des prototypes qui sont très proches des centres réels. Ce constat est également confirmé par les tables 3.15 et 3.16 qui présentent les distances calculées entres les centres réels et les prototypes détectés. Cette distance est minimale lorsqu'un prototype détecté est proche du centre réel.

BCW					
Centres réels	***IUFL***	***UFL***	***FCM***	***PCM***	***Isodata***
2,956 7,195	3,005 7,239	3,904 7,482	1,987 7,820	3,081 3,081	3,873 6,114
1,325 6,572	1,294 7,035	2,454 7,988	1,941 6,124	1,351 1,352	2,510 5.100
1,443 6,56	1,424 6,961	2,527 7,826	1,943 5.754	1,452 1,452	2,592 5,238
1,364 5,547	1,325 5,971	2,242 6,87	1,939 6,409	1,350 1,350	2,007 5,417
2,12 5,298	2,077 5,54	2,783 6,226	1,955 5,718	2,108 2,108	2,693 5,779
1,305 7,564	1,301 7,986	2,664 8,064	1,947 8,555	1,371 1,371	2,138 9,664
2,1 5,979	2,098 6,255	2,943 6,835	1,966 5,247	2,170 2,170	2,885 4,570
1,29 5,863	1,241 6,291	2,245 7,113	1,936 4,911	1,290 1,291	2,490 3,384
1,063 2,589	1,087 2,627	1,379 3,228	1,936 2,045	1,076 1,076	1,430 1,584

Table 3.11 Les centres réels et les centres obtenus par les différents algorithmes pour les données BCW.

Balance					
Centres réels	***IUFL***	***UFL***	***FCM***	***PCM***	***Isodata***
2,938 3,611 2,399	2,643 4,029 2,259	2,909 3,327 4,497	0,067 2,747 2,448	3,075 3,075 3,075	2,015 1,658 1,638
2,938 3,611 2,399	1,925 3,585 3,587	2,975 3,151 2,943	0,063 1,838 4,166	3,150 3,140 3,100	2,015 1,658 1,638
2,938 2,399 3,611	3,109 2,029 3,892	3,001 3,009 2,994	0,062 2,937 3,130	2,928 2,910 2,918	1,702 4,083 4,098
2,938 2,399 3,611	1,989 3,172 3,89	3,003 3,012 3,012	0,057 3,465 2,678	2,848 2,848 2,848	1,481 2,999 3,937

Table 3.12 Les centres réels et les centres obtenus par les différents algorithmes pour les données Balance.

Wine															
Centres réels			***IUFL***			***UFL***			***FCM***			***Isodata***		
13,744	12,278	13,153	13,680	12,290	13,145	13,041	12,622	13,030	12,509	12,962	13,817	13,804	12,829	12,947
2,010	1,932	3,333	1,886	1,792	3,504	2,308	2,473	3,338	2,454	2,512	1,890	1,883	2,634	2,530
2,455	2,244	2,437	2,431	2,243	2,422	2,369	2,318	2,430	2,288	2,398	2,440	2,426	2,362	2,437
17,037	20,238	21,416	17,222	20,258	21,323	19,367	20,693	21,671	20,775	19,773	16,890	17,023	20,083	19,966
106,338	94,549	99,312	106,365	92,358	98,310	99,895	95,990	97,571	92,318	103,763	105,206	105,511	99,451	107,529
2,840	2,2588	1,678	2,884	2,151	1,666	2,326	2,019	1,783	2,074	2,135	2,866	2,867	2,032	2,162
2,9823	2,0808	0,781	3,025	1,957	0,792	2,081	1,553	1,018	1,787	1,612	3,026	3,014	1,488	1,766
0,29	0,363	0,447	0,285	0,368	0,458	0,356	0,397	0,440	0,387	0,388	0,287	0,285	0,404	0,365
1,899	1,630	1,153	1,962	1,519	1,137	1,614	1,440	1,245	1,457	1,521	1,915	1,910	1,446	1,532
5,528	3,086	7,396	5,581	3,001	7,301	5,055	4,739	6,739	4,080	5,679	5,788	5,703	5,406	5,486
1,062	1,056	0,682	1,065	1,066	0,691	0,965	0,912	0,703	0,945	0,886	1,078	1,078	0,888	0,870
3,157	2,785	1,683	3,144	2,779	1,727	2,652	2,357	1,869	2,498	2,388	3,093	3,114	2,283	2,539
1115,711	519,507	629,895	1107,404	504,865	629,585	766,658	561,499	599,230	454,006	736,870	1214,622	1195,149	639,033	804,127

Table 3.13 Les centres réels et les centres obtenus par les différents algorithmes pour les données Wine.

Haberman's Survival																	
Centres réels		***IUFL***		***UFL***		***FCM***		***PCM***		***Isodata***							
52,017	53,679	52,501	54,330	52,000	55,912	62,684	43,705	52,008	52,007	50,469	54,561						
62,860	62,827	62,867	63,103	62,788	63,032	63,112	62,640	62,746	62,746	60,357	64,067						
2,790	7,457	3,965	5,966	3,661	3,708	3,023	3,759	2,563	2,563	1,643	20,680						

Table 3.14 Les centres réels et les centres obtenus par les différents algorithmes pour les données Haberman's Survival.

Base de données	***IUFL***		***UFL***		***FCM***		***PCM***		***ISODATA***	
BCW	0,100	1,027	2,851	3,0422	1,880	2,209	0,1617	13,195	2,681	4,345
Haberman's Survival	1,271	1,650	0,874	4,368	10,672	10,639	0,254	5,172	3,159	13,310

Table 3.15 Distance entre les centres réels et les centres obtenus par les algorithmes étudiés pour les données BCW et Haberman's Survival.

Base de données	***IUFL***			***UFL***			***FCM***			***PCM***			***ISODATA***		
Wine	8,31	14,807	1,07	349,13	42,063	30,724	98,918	65,553	107,10	483,74	112,62	4,003	79,442	119,65	174,45
Balance	1,429	0,953	1,260	0,102	1,02	2,332	5,751	2,305	2,056	0,268	0,981	1,418	2,314	3,29	1,225

Table 3.16 Distance entre les centres réels et les centres obtenus par les algorithmes étudiés pour les données Wine et Balance.

Les tables 3.15 et 3.16 montrent que les prototypes détectés par l'algorithme IUFL sont plus proches des centres réels que ceux détectés par les autres algorithmes étudiés dans les cas des données BCW et Wine. Cependant, dans le cas de la base de données Haberman's Survival, PCM a détecté un prototype plus proche à l'un des deux centres réels, et dans le cas des données Balance, les algorithmes UFL et Isodata ont chacun détecté un prototype proche à un des centres réels.

Nous avons également appliqué l'algorithme IUFL pour segmenter les deux images IRM du cerveau : image1 et image2 (Figure 3.1). Dans chaque image considérée, le vecteur x_j d'un pixel *j* est formé des niveaux de gris de ce pixel.

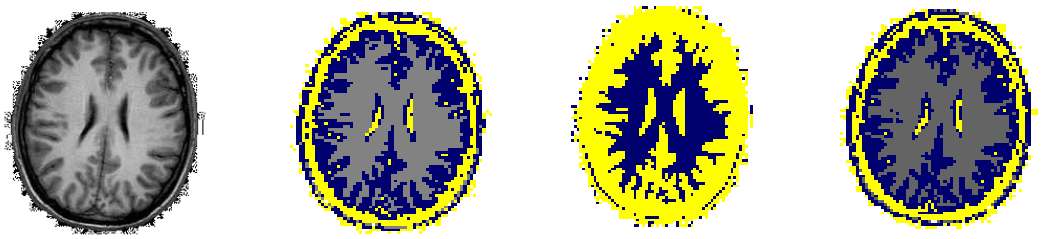
Pour l'image1, la segmentation de l'encéphale impose de fixer à 3 le nombre de classe à identifier ($c = 3$), correspondant aux trois tissus cérébraux présents dans l'encéphale à savoir la matière blanche (MB), la matière grise (MG) et le liquide céphalo-rachidien(LCR) [Philipps, 1995].

La table 3.17 présente les valeurs minimales de ξ correspondant à chaque nombre détecté de classes ($2 \leq c \leq 6$) produites par IUFL et les indices de validité de la segmentation de l'Image1. On constate que la meilleure partition correspond à $c = 3$.

ξ	***c***	***PE***	***PC***
0,20	2	0.666	0.516
0,60	2	0.556	0.530
0,70	**3**	**0.520**	**0.542**
0,71	4	0.540	0.482
0,72	5	0.543	0.473
0,75	6	0.531	0.459

Table 3.17 Nombre des classes détectées par IUFL et les valeurs des indices de validité pour l'Image1.

L'algorithme permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance à une classe donnée. La classe à laquelle est associé un pixel est celle dont le degré d'appartenance sera le plus élevé. Le résultat visuel de la segmentation est présenté par la Figure 3.2. On constate que l'algorithme Isodata qui détecte automatiquement le nombre de classes existantes n'a détecté dans cet exemple que 2 classes. Comparés à une segmentation manuelle, les résultats de l'algorithme IUFL ont montré une bonne adéquation avec la réalité.



(a) (b) (c) (d)

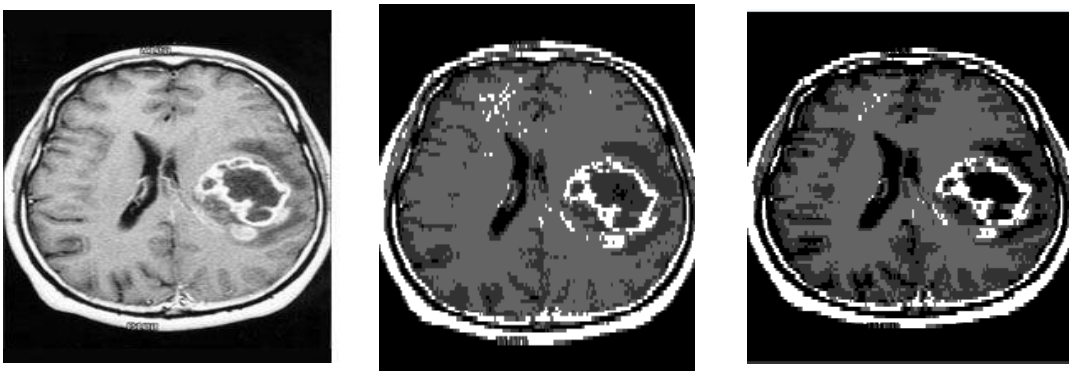
Figure 3.2 Segmentation de l'image1
(a) image Originale (b) FCM, c =3 (c) Isodata, c =2 (d) IUFL, c = 3

Quant à l'image2, la présence de la tumeur induit à ce que c soit égale à 4. Les indices de validité calculés montrent qu'effectivement la meilleure partition est obtenue pour c = 4. La table 3.18 présente les différentes valeurs de classes détectées ainsi que les indices de validité produits par IUFL.

ξ	***c***	***PE***	***PC***
[0,10 – 0,50]	*3*	*0.407*	*0.690*
[0,60- 0,70]	***4***	***0.353***	***0.713***
0,75	*5*	*0.503*	*0.564*
0,80	*8*	*0.503*	*0.495*

Table 3.18 Nombre de classes produites par IUFL
et indices de validité pour l'image2.

Les prototypes appris par IUFL sont utilisés pour initialiser FCM qui donne une très bonne initialisation des centres des classes par rapport à FCM (figure 3.3).



(a) Image originale (b) FCM avec $c = 4$ (c) IUFL + FCM avec $c = 4$

Figure 3.3 Segmentation de l'image 2.

Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode basée sur une procédure d'apprentissage non supervisé qui explore la base de données X pour (1) découvrir le nombre de classes et (2) fournir un prototype pour chaque classe détectée. À cette fin, une nouvelle règle d'apprentissage a été proposée. Elle consiste à reporter le traitement immédiatement de certains objets qui ne peuvent être facilement reconnus et qui impactent la partition finale. Ces objets sont reportés jusqu'à ce d'autres objets nouvellement rencontrés soient examinés dissipant ainsi le flou ou réduisant la confusion. En d'autres termes, les informations fournies par les objets examinés au cours du processus itératif contribuent à minimiser le flou de la partition finale.

Nos expériences ont montré que l'incorporation de la règle proposée dans le processus améliore la précision de l'apprentissage. Lors de nos tests sur les six ensembles de données, une amélioration a été perçue par rapport à FCM. En plus, l'approche proposée est simple et habile à s'auto-organiser, bien qu'elle nécessite légèrement plus de temps puisqu'elle ré-explore des objets non marqués. Elle peut également être utilisée comme une initialisation pour d'autres algorithmes.

CHAPITRE 4

NOUVELLE APPROCHE POUR L'APPRENTISSAGE DES DONNEES BRUITEES

Introduction

Dans la réalité, les limites entre les classes sont souvent ambiguës et incertaines. Cette incertitude s'exprime par le fait qu'un objet possède des caractéristiques qui permettent de l'assigner à plus qu'une classe. Cette incertitude est d'autant plus accentuée par la présence des valeurs aberrantes ou points isolés. Ces derniers déséquilibrent l'analyse des données en se voyant accorder une importance plus grande qu'ils n'ont [Ben-David, 2014]. Ainsi, des algorithmes d'apprentissage non supervisé peuvent générer des partitions erronées. Par conséquent, la détection des valeurs aberrantes est importante [Ramaswamy 2000] [Tang, 2012]. Cependant, ces valeurs aberrantes ne sont pas nécessairement erronées et peuvent contenir des informations significatives [Rehm, 2007]. Tel est le cas lors de la détection d'une fraude [Bolton, 2002] ou d'une intrusion sur un réseau informatique [Lane, 1999]. Par conséquent, les valeurs aberrantes ne doivent pas être systématiquement rejetées [Berthold, 1999].

Certaines méthodes ont été proposées pour détecter les valeurs aberrantes [Davé, 1991] [Davé, 1997 – a] [Davé, 1997 – b] [Ohashi, 1984]. Toutefois, ces méthodes nécessitent certains paramètres difficiles à estimer.

Dans ce chapitre, nous présentons deux approches qui permettent d'effectuer simultanément un apprentissage non supervisé des données et un isolement des valeurs aberrantes pour réduire leur impact et améliorer les résultats.

4.1. Préliminaires sur les valeurs aberrantes

L'apprentissage non supervisé tente de trouver une structure de données dans les données considérées selon leurs caractéristiques similaires, ce qui est difficile dans le contexte non supervisé où aucune information préalable sur les données n'est fournie. Cette difficulté s'accentue lorsque ces données contiennent des valeurs aberrantes, qu'on appelle en littérature scientifique les « *outliers* ».

Une valeur aberrante est en général une valeur erronée correspondant à une mauvaise mesure, une erreur de calcul, une erreur de saisie ou une fausse déclaration. C'est un élément considérablement dissemblable du reste des données [Han, 2006].

La détection des valeurs aberrantes est importante dans de nombreux domaines, particulièrement dans le traitement d'image [Al-Zoubi, 2006], la détection des fraudes [Bolton, 2002] et l'identification des intrusions sur le réseau informatique [Lane, 1999]. Plusieurs approches ont été proposées dans la littérature scientifique pour les détecter [Knorr, 1998] [Ramaswamy, 2000]. Ces approches peuvent être classées en des approches fondées sur la distribution, et d'autres basées sur la proximité.

Dans les approches fondées sur la distribution, les valeurs aberrantes sont définies en fonction de la probabilité de distribution [Hawkins, 1980] [Barnett, 1994]. Ces approches développent des modèles statistiques de sorte que les points qui ont une faible probabilité d'appartenance à ce modèle statistique sont déclarés comme des valeurs aberrantes [Al-Zoubi, 2010]. Toutefois, ces approches exigent une connaissance préalable de la distribution des données, ce qui rend ces approches difficiles à utiliser dans des applications pratiques [Al-Zoubi, 2010].

Dans les approches basées sur la proximité, une valeur aberrante est un point isolé qui est loin du reste des données. Cette modélisation contient spécifiquement trois méthodes: approches fondées sur la densité, celles basées sur une classification et celles fondées sur la distance.

Les approches basées sur la densité calculent la densité des régions et considèrent les points dans les régions à faibles densité comme des valeurs aberrantes [Breunig, 2000]. Ces approches attribuent un degré d'aberrance pour chaque point de données. Le plus connu de ces degrés est le facteur d'aberrance local (Local Outlier Factor (LOF)). Il est dit local dans la mesure où seul un voisinage restreint de chaque objet est pris en compte.

Les approches fondées sur la distance distinguent les valeurs aberrantes des autres en fonction du nombre d'objets de leur voisinage. Ainsi, un point x est une valeur aberrante, si on ne trouve que M points de données à une distance d de x [Knorr, 1998]. Cette approche ne nécessite pas la connaissance à priori de distribution des données. Cependant, comme les valeurs de M et d sont choisies par l'utilisateur, il est difficile de déterminer leurs valeurs [Knorr, 1998] [Ramaswamy, 2000]. Pour surmonter cette limite, un autre algorithme a été proposé [Angiulli, 2002].

Les approches fondées sur une classification utilisent la taille des classes obtenues pour indiquer la présence des valeurs aberrantes. Ces approches considèrent que les valeurs aberrantes forment de petites classes tandis que les objets normaux appartiennent aux classes denses [Loureiro, 2004].

Résoudre à la fois le clustering et la détection des valeurs aberrantes est fortement souhaité [Ott, 2014] [Knorr, 1998] [Loureiro, 2004]. Plusieurs algorithmes ont été proposés dans ce sens et parmi eux certains prennent en compte toutes les données, mais minimisent l'influence des valeurs aberrantes [Davé, 1991] [Davé, 1997]. L'algorithme le plus connu est le FCM robuste [Davé, 1991]. Dans cet algorithme, la notion de classe des « *outliers* » est introduite. Cette classe est caractérisée par un prototype fictif qui est à une constante distance δ aux autres objets. D'où l'importance de déterminer la distance δ qui est un paramètre critique de l'algorithme [Cimino 2007].

Dans ce qui suit, nous proposons deux approches qui permettent de fournir les valeurs aberrantes qui existent, et de partitionner les données en détectant automatiquement les classes qu'elles forment.

4.2. Variante de FCM proposée pour le traitement de données bruitées

L'algorithme FCM optimise la fonction objective J_m définie par :

$$J_m(U,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d^2(x_k, v_i) \qquad (4.1)$$

J_m dépend des distances des objets aux centres des classes pondérées par les degrés d'appartenance. Ainsi, une valeur aberrante influence l'estimation de la moyenne des classes [Jolion, 1989] et les centres des classes peuvent alors être éloignés des centres réels. L'algorithme FCM n'est pas robuste donc aux valeurs aberrantes.

Dans ce paragraphe, nous présentons une variante de FCM pour pallier le problème des valeurs aberrantes. Cette approche est constituée de deux étapes.

La première étape permet de détecter les valeurs susceptibles d'être aberrantes et que nous appelons *possible outliers*, à l'aide d'un nouvel concept appelé *degré de proximité* des objets. Ce degré reflète la proximité d'un objet aux autres objets considérés.

La seconde étape permet de partitionner les données contenant des valeurs aberrantes par une variante adaptée de l'algorithme FCM, et ceci en introduisant le concept de classes aberrantes. Il s'agit de considérer chaque valeur aberrante comme centre d'une classe aberrante au lieu de considérer une seule classe de bruit contenant toutes les valeurs aberrantes, comme proposé dans l'algorithme FCM robuste [Davé, 1991].

4.2.1 Les valeurs aberrantes possibles

La première phase de notre approche consiste à détecter les objets susceptibles d'être aberrants [Dik, 2014]. Elle ne nécessite pas le partitionnement des données en classes, mais elle informe juste si un élément de données est susceptible d'être aberrant.

Le principe de cette phase repose sur la notion de degré de proximité d'un point par rapport aux autres points considérés, et émane du fait qu'un point normal a plus de voisins avec lesquels il partage des caractéristiques similaires. Cette notion reflète la proximité d'un objet aux autres objets et peut être considérée comme l'opposé du degré d'isolement qui caractérise les valeurs aberrantes. Cependant, au lieu d'attribuer un facteur d'isolement à un objet en fonction de sa distance à son voisinage local [Breunig, 2000], le degré de proximité proposé dépend de toutes les données, ainsi le problème de détermination du voisinage local ne se pose pas. Il consiste à calculer la somme des similarités d'un objet avec tous les autres points, et non seulement ses voisins. Ainsi, le degré de proximité d'un objet x_i est défini par:

$$D(x_i) = \left(\sum_{\substack{j=0 \\ j \neq i}}^{n} Sim(x_i, x_j) \right) \tag{4.2}$$

Avec:

$$Sim(x_i, x_k) = 1 - \frac{\|x_i - x_k\|_A^2}{p} \tag{4.3}$$

Où :

Sim (x_i, x_k) est la mesure de similarité entre les objets x_i et x_k.

p est la dimension des objets $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \Re^p$.

A, utilisée par la similarité *Sim*, est la matrice *pxp* définie par [Bouroumi, 2000] :

$$A_{jt} = \begin{cases} (r_j)^{-2}, & j = t \\ 0, \ sinon \end{cases} \tag{4.4}$$

Où le facteur r_j représente la différence entre les valeurs maximales et minimales d'un attribut. Il est défini par :

$$r_j = \max_{1 \le i \le n} \{x_{ij}\} - \min_{1 \le i \le n} \{x_{ij}\} \qquad 1 \le j \le p \tag{4.5}$$

Le processus de détermination des valeurs aberrantes possibles consiste à calculer pour chaque objet x_i, $1 \le i \le n$, la valeur $D(x_i)$, et déterminer les L valeurs minimales. Pour cela, définissons d'abord les paramètres suivants:

$D^1_{min}, D^2_{min}, \ldots, D^L_{min}$ les L minimales valeurs de degré de proximité, tels que : $D^1_{min} \le D^2_{min} \le \ldots \le D^L_{min}$, avec $D^1_{min} = \min_{1 \le i \le n}(D(x_i))$;

D_{range} la différence entre les degrés supérieur et inférieur de proximité: $D_{range} = \max_{1 \le i \le n}(D(x_i)) - \min_{1 \le i \le n}(D(x_i))$;

M est le nombre des valeurs aberrantes possibles.

Le processus se résume en l'exécution de la boucle suivante :

Pour k allant de 1 à L-1 ***faire***

Si ($D^k_{min} << D^{k+1}_{min}$) ***alors***

Les objets correspondant à $D^k_{min}, D^{k-1}_{min}, \ldots D^1_{min}$ sont des valeurs aberrantes possibles.

$M \leftarrow k$.

Fin si

Fin pour

Notre approche ne nécessite ni la distance minimale *d* que l'utilisateur devrait définir dans les approches basées sur la distance, ni de procéder à une classification des données. Toutefois, pour estimer facilement l'équation : $D^k_{min} << D^{k+1}_{min}$ on procède à leur pondération par la valeur de D_{range}.

4.2.2 Phase d'apprentissage

Dans cette étape, l'ensemble de données est partitionné en utilisant les M valeurs aberrantes possibles détectées lors de la première étape, en tant que centres en plus des c centres choisis aléatoirement. Ainsi, l'algorithme proposé POFCM (*Possible outliers FCM*) est exécuté avec

(M + c) centres. À la fin du traitement, nous vérifions si certaines classes aberrantes contiennent des objets autres que les valeurs aberrantes initiales. Si tel est le cas, et N $(N \leq M)$ étant leur nombre, alors les N possibles valeurs aberrantes correspondant à ces classes aberrantes ne sont pas des valeurs aberrantes et l'algorithme POFCM est exécuté à nouveau avec les ($c + M - N$) centres.

Algorithme 4.1 : Algorithme POFCM

Entrées: Ensemble de données non étiquetées $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$;
Le nombre de classes c $(1< c < n)$
Le paramètre m $(m >1)$; t_{max} (nombre maximal des itérations); ε (seuil de tolérance);

Sorties : Nombre estimé de classe c^*, matrice de prototypes $V= (v_1,\ldots,v_{c^*})$, valeurs aberrantes

Début

Déterminer les M possibles valeurs aberrantes notées y_i

faire

$M_{init} \leftarrow M$ (nombre initial des valeurs aberrantes)

Initialiser prototypes $V_0= (y_1,y_2, \ldots y_M, v_{1,0}, v_{2,0}, \ldots, v_{c,0}) \in \Re^{(c+M)\,x\,p}$

faire

Calculer U_t en utilisant V_{t-1} et u_{ik} (Eq.1.24)

Calculer V_t en utilisant U_t et v_i (Eq.1.25)

Tant que ($\|V_t - V_{t-1}\|_{err} > \varepsilon$) et ($t < t_{max}$))

Pour chaque classe de valeur aberrante C_i ***faire***

Si (card (C_i) > 1) ***alors***

$M \leftarrow M-1$; (Ce n'est pas une vraie valeur aberrante)

Sinon

Marquer l'objet O_i comme valeur aberrante

Fin si

Fin pour

Tant que ($M \neq M_{init}$)

$U^* \leftarrow U_t$

$V^* \leftarrow V_t$

Fin

4.2.3 Résultats et discussions

Pour évaluer les performances de notre méthode, des expériences sont menées sur un ensemble X_1 de données artificielles, et sur quatre ensembles de données réelles mises à

disposition par l'Université de Californie à Irvine (UCI machine learning repository) [Blake, 1998] [Asuncion, 2007]: Wine, BCW, Breast-Tissu et Spect-Heart (Table 4.1).

Base de données	*Nombre d'objets*	*Nombre d'attributs*	*Nombre de classes*
X_1	42	1	2
Wine	178	13	3
BCW	699	9	2
SPECT Heart	267	22	2
Breast -Tissu	106	9	6

Table 4.1 Description des données étudiées.

L'ensemble de données X_1 est un exemple artificiel dérivé de [Bouroumi 2000]. Il contient deux classes bien séparées dans le plan et deux valeurs aberrantes (Figure 4.1).

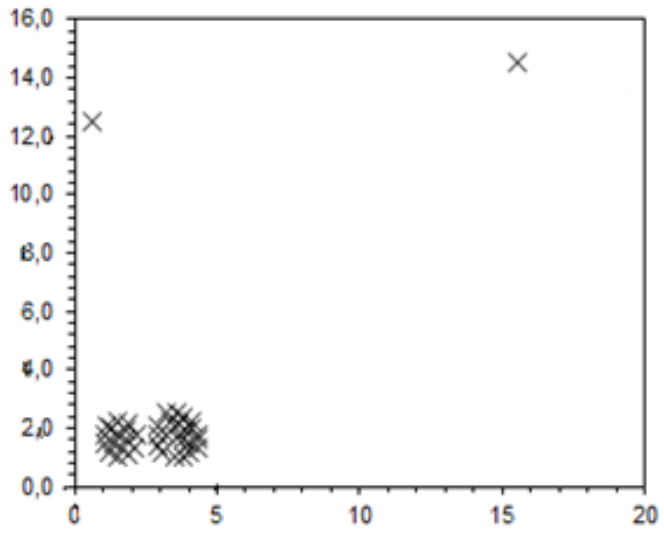


Figure 4.1 Présentation des données artificielles de X_1.

Dans un premier temps, nous cherchons s'il existe des valeurs aberrantes possibles dans les ensembles de données considérés. Pour cela, nous fixons L = 4, nous calculons le degré de proximité des objets et nous recherchons les quatre petites valeurs.

Nous constatons que :

Pour l'ensemble X_1 : $\frac{D^1_{min}}{D_{range}}$ = 0.12 et $\frac{D^2_{min}}{D_{range}}$ = 0.51 alors que les valeurs de $\frac{D^3_{min}}{D_{range}}$ et $\frac{D^4_{min}}{D_{range}}$ sont relativement grandes (respectivement 1.082, et 1.083).

Pour l'ensemble Wine, $\frac{D^1_{min}}{D_{range}}$ = 3.93 alors que les valeurs calculées de $\frac{D^2_{min}}{D_{range}}$, $\frac{D^3_{min}}{D_{range}}$ et $\frac{D^4_{min}}{D_{range}}$ sont relativement grandes (respectivement 4.07, 4.11 et 4.14).

Pour l'ensemble Breast-Tissu, $\frac{D^1_{min}}{D_{range}}$ = 0.44 alors que les valeurs de $\frac{D^2_{min}}{D_{range}}$, $\frac{D^3_{min}}{D_{range}}$ et $\frac{D^4_{min}}{D_{range}}$ sont presque égales (respectivement 0.91, 0.93 et 0.95).

Les résultats de la table 4.2 montrent qu'il existe des valeurs aberrantes possibles pour X_1, Wine et Breast-tissu.

Base de données	D^1_{min}	D^2_{min}	D^3_{min}	D^4_{min}	D_{range}	$\frac{D^1_{min}}{D_{range}}$	$\frac{D^2_{min}}{D_{range}}$	$\frac{D^3_{min}}{D_{range}}$	$\frac{D^4_{min}}{D_{range}}$
X1	4,19	17,17	35 ,96	35 ,99	33,23	0,12	0,51	1,082	1,083
BCW	165,73	166,20	172,03	174,66	352,07	0,47	0,472	0,48	0,49
Wine	109,67	113,80	114,88	115,60	27,9	3,93	4,07	4,11	4,14
Heart	51,52	53,12	56,15	57,04	79,4	0,65	0,67	0,71	0,72
Breast Tissu	27,00	54,91	55,99	57,53	60,04	0,44	0,91	0,93	0,95

Table 4.2 Résultats de détection des valeurs aberrantes.

On déduit de ces résultats que :
L'ensemble X_1 contient deux valeurs aberrantes possibles telles schématisées par la figure.4.2, les ensembles de données Wine et Breast-Tissu contiennent des valeurs aberrantes possibles, et que les ensembles de données BCW et Spect-Heart ne contiennent pas de valeurs aberrantes.

La table 4.3 présente les indices des éléments susceptibles d'être des valeurs aberrantes dans chaque ensemble de données.

Base de données	*Indice de l'objet 1*	*Indice de l'objet 2*	*Indice de l'objet 3*	*Indice de l'objet 4*
X1	0 (*)	1 (*)	10	11
Wine	121 (*)	158	146	59
Breast-Tissu	102 (*)	86	97	105

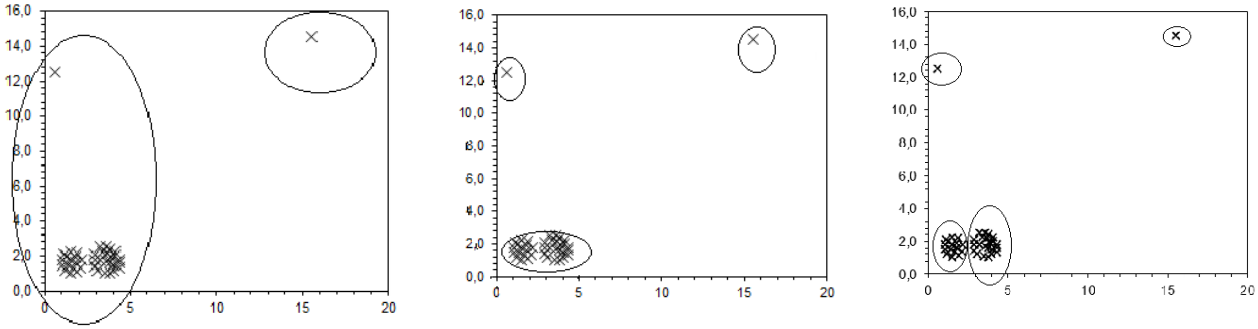
Table 4.3 Indices des valeurs aberrantes possibles**.**
* indique une valeur aberrante vraie.

Une fois que les possibles valeurs aberrantes sont déterminées, l'algorithme POFCM est exécuté. Les résultats de cet algorithme sont présentés dans la table 4.4.

Base de données	*c*	*M*	*FCM*	*FCM sans valeur aberrante possible*	*POFCM*
X1	2	2	61.91%	100%	**100%**
Wine	3	1	69.67%	77.97%	69.67%
Breast -Tissu	6	1	30.13%	**31.43**	**32.08%**

Taux de reconnaissance des algorithmes POFCM,
FCM avec et sans les valeurs aberrantes possibles.

Pour des raisons démonstratives, nous présentons dans la figure 4.2 les résultats du partitionnement de l'ensemble X_1 par l'algorithme FCM avec différentes valeurs de *c*, et dans la figure 4.3 le partitionnement de X_1 par l'algorithme POFCM. Nous constatons que l'algorithme FCM n'a détecté les valeurs aberrantes que pour *c=4* alors que POFCM a détecté les classes réelles et les valeurs aberrantes (figure 4.3).



(a) (b) (d)

Figure 4.2 Représentation des résultats de FCM sur X_1 avec $c = 2$ (a), $c=3$ (b) et $c= 4$ (d)

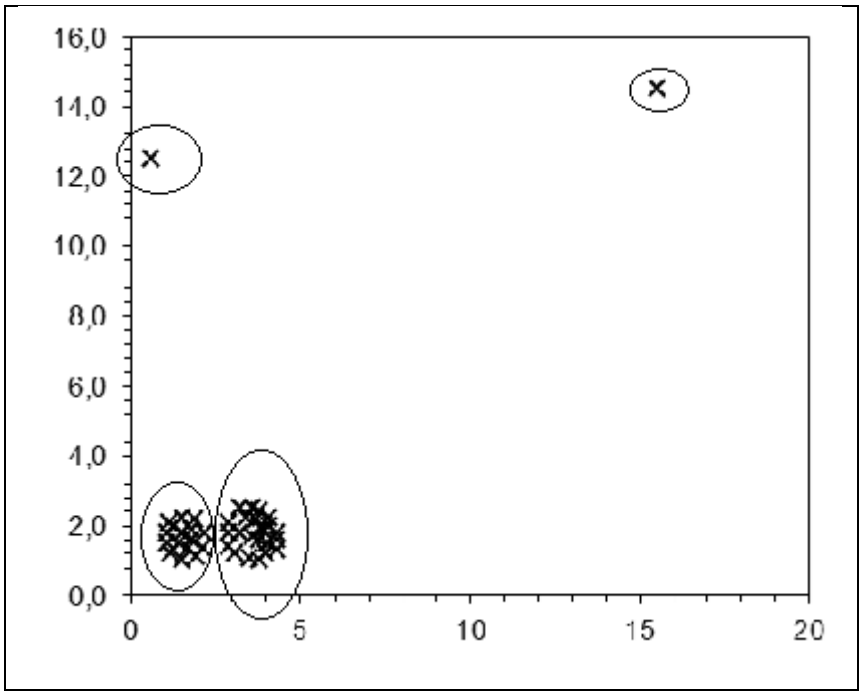


Figure 4.3 Représentation des résultats de POFCM sur X_1 avec $c=2$ et $M=2$.

Les résultats de la table 4.4 montrent que l'apprentissage des données des bases X_1 et Breast-tissu a été amélioré par l'algorithme proposé. Cela est dû à l'existence de valeurs aberrantes dans les données.

Nous avons présenté dans ce paragraphe une variante adaptée de l'algorithme FCM pour détecter des éventuelles valeurs aberrantes et partitionner l'ensemble des données. Les résultats expérimentaux présentés montrent que cette approche améliore l'apprentissage des données contenant des valeurs aberrantes.

4.3 Algorithme flou robuste pour l'apprentissage non supervisé

Dans ce paragraphe, nous présentons un algorithme d'apprentissage flou non supervisé robuste et dynamique (RDUFL) qui vise à partitionner un ensemble de données en détectant le nombre de classes existantes et les éventuelles valeurs aberrantes. Il se compose de trois étapes principales :

– La première étape est une méthode de prétraitement dans laquelle les valeurs aberrantes possibles sont déterminées et mises en quarantaine en utilisant le concept de degré de proximité présenté précédemment.

– La deuxième étape est une méthode d'apprentissage qui consiste à détecter automatiquement le nombre de classes avec leurs prototypes moyennant un seuil dynamique. Ce seuil est automatiquement déterminé en fonction de la similarité entre les prototypes détectés, et qui sont mis à jour lors de l'exploration de nouvelles données.

– La dernière étape traite les objets mis en quarantaine et qui étaient détectés lors de la première étape, afin de déterminer s'ils appartiennent à une des classes détectées dans la deuxième phase.

4.3.1 Valeurs aberrantes possibles

Dans cette étape, on détermine les M valeurs aberrantes possibles telles présentées dans le paragraphe précédent, en se basant sur la mesure de similarité définie par:

$$Sim(x_i, x_k) = 1 - \frac{\|x_i - x_k\|_A}{\sqrt{p}} \tag{4.6}$$

Où A est la matrice pxp définie par l'équation (Eq.4.4).

Le choix de cette mesure de similarité est motivé par les propriétés suivantes [El Imrani, 2000]:

(i) $Sim(x_i , x_k) \in [0,1]$; $\forall x_i , x_k \in \Re^p$ puisque :

$$(x_{ij} - x_{kj}) \le r_j \quad \forall 1 \le j \le p \tag{4.7}$$

$$\|x_i - x_k\|_A = \sqrt{\sum_{j=1}^{p} \frac{(x_{ij} - x_{kj})^2}{r_j^2}} \le \sqrt{p} \tag{4.8}$$

(ii) $Sim(x_i\ x_k) = 1$ pour $x_i = x_k$

(iii) $Sim(x_i\ x_k)$ tend vers 0 lorsque $(x_{ij} - x_{kj})$ tend vers $r_j \quad \forall\ 1 \leq j \leq p$, ce qui signifie que les objets présentent un maximum de différence pour chacune de leurs *p* composantes.

Une fois ces objets détectés et mis en quarantaine, on partitionne l'ensemble des autres données lors de l'étape d'apprentissage sans tenir compte de ces *M* valeurs aberrantes possibles.

4.3.2 Phase d'apprentissage

En supposant que les vecteurs d'objets qui forment la base de données d'apprentissage *X* appartiennent au moins à deux classes distinctes, et étant donné une mesure de similarité inter-points, la méthode proposée commence par la création de deux classes autour des deux premiers objets x_1 et x_2 [Benrabah, 2005].

L'algorithme proposé explore séquentiellement tous les *n-2-M* objets de la base d'apprentissage X et analyse leurs ressemblances en utilisant la mesure de similarité donnée par l'équation (4.6). Un seuil dynamique ξ est utilisé pour détecter quand un nouvel objet n'est pas reconnu et dissemblable à tous les prototypes existants. Lorsque ce seuil n'est pas atteint, une nouvelle classe est créée et son prototype est initialisé avec l'objet courant.

Le seuil ξ représente le minimum de similarité que doit avoir chaque objet avec son prototype le plus proche. Dans ce travail, ξ est dynamique et dépend de l'objet courant. Il est calculé automatiquement à chaque itération comme suit:

Si x_i est l'objet courant et v_k son plus proche prototype, le seuil ξ est défini par:

$$\xi = \underset{\substack{1 \leq j \leq c \\ j \neq k}}{Min}\left[\frac{Sim(v_j, v_k)}{2}\right] \tag{4.9}$$

L'algorithme utilise la mesure de similarité et son seuil associé pour construire des classes. A chaque itération, la similarité de l'objet courant avec les prototypes existants est calculée. Selon le maximum de cette similarité, deux décisions seront concevables:

Une nouvelle classe est créée, lorsque:

$$\underset{1\le k\le c}{Max}\,(Sim(i,k)) \;<\xi \tag{4.10}$$

Cela signifie que l'élément actuel x_i n'est pas suffisamment similaire avec les prototypes des classes détectées précédemment. Il est censé provenir d'une classe qui n'est pas encore découverte et par conséquent doit représenter une nouvelle classe [Bouroumi, 2000].

Les prototypes sont mis à jour, lorsque :

$$\underset{1\le k\le c}{Max}\,(Sim(i,k)) \;\ge \xi \tag{4.11}$$

L'objet courant x_i est considéré comme ayant la similarité minimale requise avec les prototypes des classes précédemment détectées. Donc, nous ne devons pas créer de nouvelle classe.

Le fait de ne pas prendre en compte les possibles valeurs aberrantes au cours de cette phase permet une meilleure recherche des prototypes et une non-distorsion lors de la détection automatique du nombre de classes.

4.3.3 Affectation des valeurs aberrantes possibles

Dans cette phase, nous traitons les M possibles valeurs aberrantes O_i ($1 \le i \le M$) qui ont été détectées et mises en quarantaine au cours de la première phase. Pour chaque objet O_i, on cherche son plus proche prototype v_k et sa classe correspondante notée C_k. Soit x_l l'élément de C_k le plus éloigné du prototype v_k , déterminé par :

$$x_1 = \underset{x_j \in C_k}{Max}\left(d(x_j, v_k)\right) \tag{4.12}$$

Si le point x_l est plus similaire à l'objet O_i qu'à son plus proche prototype v_k, c'est que l'objet O_i a des voisins qui sont les voisins de x_l. Cet objet O_i n'est pas alors une valeur aberrante.

Ainsi deux cas se présentent :

- Si $Sim(O_i, x_l) \;< Sim(x_l, v_k)$: O_i est une valeur aberrante.
- Si $Sim(O_i, x_l) \;\ge Sim(x_l, v_k)$: O_i n'est pas une valeur aberrante.

Algorithme 4.2 : Algorithme RDUFL

Entrées: Ensemble de données non étiquetées $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$;

Sorties : Nombre estimé de classe c^*, matrice de prototypes $V= (v_1,\ldots,v_{c^*})$

Etape 1: Déterminer les M valeurs aberrantes possibles O_i.

Pour i allant de 1 à n **faire**

Calculer $D(xi)$

Calculer $D^1_{min}, D^2_{min}, \ldots, D^L_{min}$ tels que : $D^1_{min} \leq D^2_{min} \leq \ldots \leq D^L_{min}$

Calculer D_{range}

Pour k allant de 1 à L-1 ***faire***

Si $\left(\frac{D^k_{min}}{D_{range}} << \frac{D^{k+1}_{min}}{D_{range}}\right)$ ***alors***

Marquer les objets correspondants à D^k_{min}, D^{k-1}_{min} ,… D^1_{min} comme valeurs aberrantes possibles

$M \leftarrow k$

Finsi

Finpour

Finpour

Mettre en quarantaine les valeurs aberrantes possibles Oi

Etape 2 : Partitionner l'ensemble X après avoir enlevé les objets O_i

c ←2

$v_1 \leftarrow x_1$

$v_2 \leftarrow x_2$

Pour i allant de 2 à n-2-M ***faire***

Déterminer le plus proche prototype v_k à l'objet x_i

Si $\left(\underset{1\leq j\leq c}{Max}\ (Sim(i,j)) < \underset{\substack{1\leq j\leq c \\ j\neq k}}{Min}\left[\frac{Sim(v_j, v_k)}{2}\right]\right)$ ***alors***

$c \leftarrow c+1$

$v_i \leftarrow x_i$

Sinon

Mettre à jour les prototypes v_j, $1 \leq j \leq c$ (Eq. 4.5)

Finsi

Finpour

Etape 3 : Traitement des valeurs aberrantes possibles

Pour *i* allant de *1* à M **faire**
Chercher le prototype le plus proche v_k à l'objet O_i (C_k la classe correspondante)
Chercher le point x_l de la classe C_k le plus éloigné du prototype v_k
Si $(Sim(O_i, x_l) < Sim(x_l, v_k))$ ***alors***
Marquer O_i est comme valeur aberrante
Sinon
Réaffecter O_i à la classe C_k
Finsi
Finpour
Fin

4.3.4 - Résultats et discussions

Pour évaluer la performance de notre approche, des expériences sont menées sur un ensemble de données artificielles X_2 et huit ensembles de données mises à la disposition par UCI [Blake 1,998] [Asuncion, 2007]. En outre, l'efficacité de cette méthode est évaluée en la comparant à d'autres méthodes d'apprentissage non supervisées connues, à savoir : les c-moyennes floues (FCM), les c-moyennes possibilistes (PCM) et le robuste FCM appelé aussi « Noise clustering (NC) ».

Ensemble de données artificielles

L'ensemble de données X_2 est un exemple artificiel à deux dimensions dérivé de [Bouroumi, 2000]. Il est formé de trois classes de 58 points dans le plan et 7 valeurs aberrantes (figure 4.4). Cet ensemble de données bidimensionnel est important vu la possibilité qu'il offre en termes de visualisation.

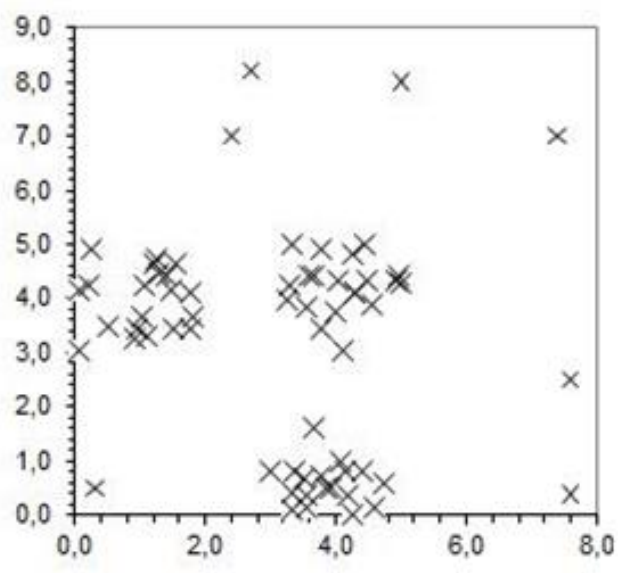


Figure 4. 4 Représentation des données de l'ensemble X_2.

Pour l'ensemble des données X_2, l'approche proposée détecte 7 valeurs aberrantes possibles. Les phases d'apprentissage et traitement des ces valeurs aberrantes possibles montrent qu'elles sont des vraies valeurs aberrantes. Le nombre de classes détectées est trois et le taux de reconnaissance est de 100% (Figure 4.5).

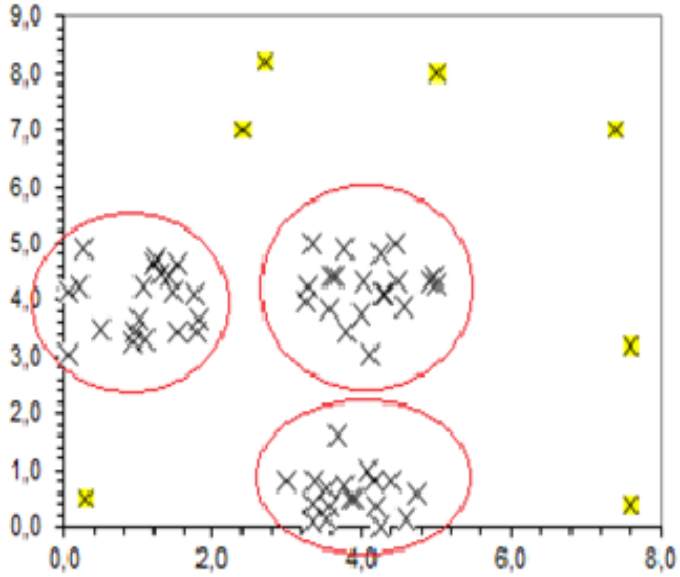


Figure 4. 5 Représentation des classes obtenues par l'algorithme RDUFL appliqué à X_2. Les valeurs aberrantes sont représentées en jaune.

En ce qui concerne les résultats de l'algorithme FCM, celui-ci n'a pas réussi à détecter les valeurs aberrantes pour c = 3 (Figure (4.6.a)). Il a seulement détecté deux valeurs aberrantes pour c = 4 (Figure (4.6.b)) et trois valeurs aberrantes pour c = 5 (Figure (4.6.d)). Ce n'est que pour c = 6 que FCM a finalement détecté les 7 valeurs aberrantes (Figure (4.6.e)) en les considérant comme des points appartenant à deux classes ayant un cardinal faible.

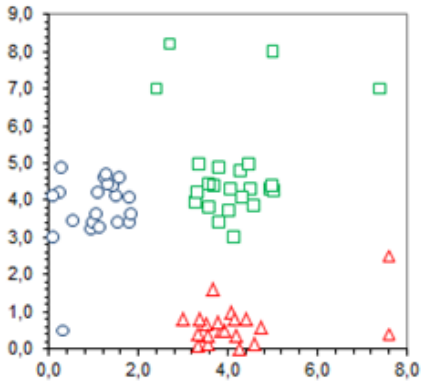


Figure (a) c = 3

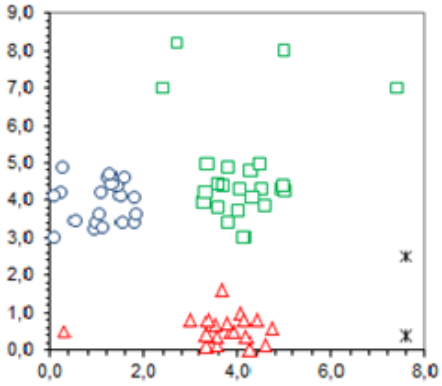


Figure (b) c = 4

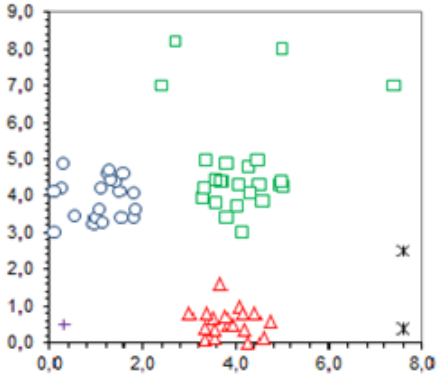


Figure (d) c = 5

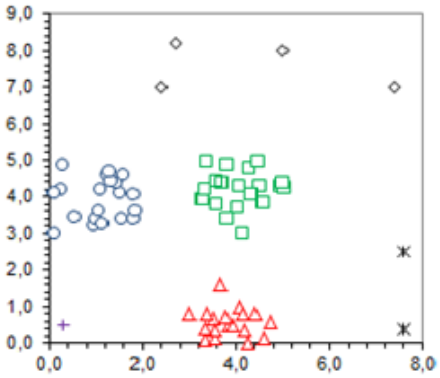


Figure (e) c = 6

Figure 4.6 Apprentissage non supervisé de X_2 par FCM pour différentes valeurs de c.

L'algorithme robuste FCM n'a détecté que deux valeurs aberrantes (figure 4.7). De plus, le taux de reconnaissance obtenu par l'algorithme robuste FCM est de 92,31%. Cet algorithme dépend de la valeur de λ. Pour les tests que nous avons effectués, la valeur de λ est choisie comprise entre 0,01 et 0,9, et le meilleur résultat est obtenu pour λ= 0.7.

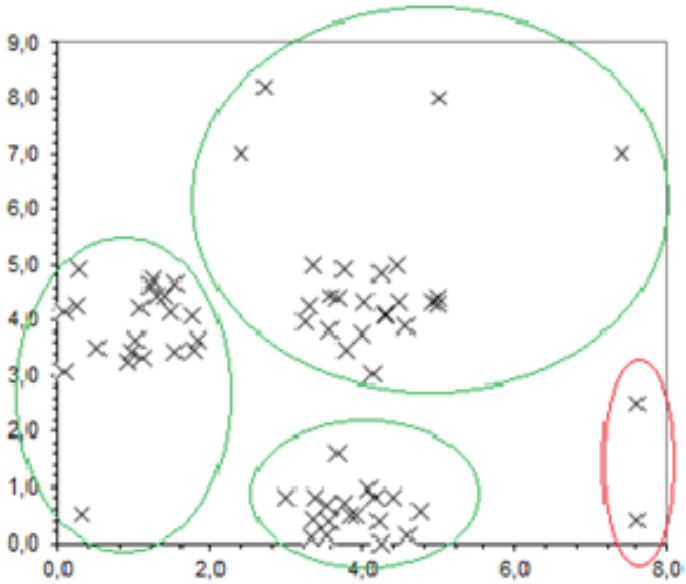


Figure 4.7 Résultats de Robust-FCM sur X_2
La classe des valeurs aberrantes est en rouge.

Les taux de reconnaissance de l'apprentissage par les algorithmes considérés en comparaison avec l'algorithme proposé sont présentés par la table 4.5 comme suit :

FCM	*PCM*	*Robust-FCM*	*RDUFL*
89.23%	**40%**	**92.31%**	**100%**

Table 4. 5 Taux de reconnaissance pour l'ensemble des données X_2.

Ces résultats montrent la sensibilité de ces algorithmes aux valeurs aberrantes et leurs difficultés à extraire correctement les classes [Jain 2009, Jolion 1989].

Ensemble de données réelles

Les expériences de test ont été réalisées avec huit ensembles de données: Lymphography, Diabetes, Indian, Haberman's Survival, BCW, Post-operative Patient, Parkinsons et EEG Eyes State. Les bases de données Indian, Haberman's Survival et BCW ont été présentées dans les chapitres précédents. Nous présentons dans ce paragraphe les cinq autres ensembles de données utilisées.

En premier lieu, l'ensemble de données considéré *« Lymphographie »* dispose de 148 objets avec 18 attributs. Ce sont des observations qui ont été obtenues à partir de patients souffrant d'un cancer dans le système lymphatique. Ce dernier est l'ensemble de tissus et d'organes qui fabriquent et entreposent les cellules qui combattent les infections et les maladies. La base de données considérée est composée de quatre classes: normale (2 objets), métastases (81 objets), lymphes malignes (61 objets) et fibrose (4 objets).

La base de données « *Diabetes* » est composé de 768 objets avec 8 attributs. Cet ensemble de données comprend uniquement les femmes appartenant au patrimoine indien Pima vivant près de Phoenix, en Arizona. Cet ensemble de données est extrait d'une base de données plus vaste appartenant à l'origine à l'Institut national du diabète, des maladies digestives et rénales. Les données forment deux classes : la classe 0 avec 500 instances et la classe1 interprétée comme "testé positif pour diabète " avec 268 instances.

L'ensemble de données « *Parkinsons Disease* » est composée d'un ensemble de mesures vocales biomédicales. Il existe 22 attributs et 195 échantillons issus de deux classes correspondant à des personnes en bonne santé et à celles atteintes de la maladie de Parkinson. Les deux classes contiennent respectivement 48 et 147 points.

L'ensemble de données « *Postoperative Patient* » vise à déterminer où envoyer les patients dans une zone de recouvrement postopératoire. Le nombre d'instances est de 90 réparties en trois classes: classe I (patient envoyé à l'unité de soins intensifs) avec 2 éléments, classe S (patient prêt à rentrer chez lui) avec 24 éléments et classe A (patient envoyé à l'hôpital général) avec 64 éléments.

L'ensemble de données « *EEG Eyes* » se compose de valeurs EEG prises avec un « *Neuroheadset Emotiv EEG* », et d'une valeur indiquant l'état de l'œil. Cet état oculaire a été détecté via une caméra lors de la mesure EEG et ajouté manuellement au fichier plus tard. L'état 0 indique que l'œil est ouvert et l'état 1 indique qu'il est fermé. L'ensemble des données est formé de 14980 échantillons ayant 14 attributs. Les deux classes contiennent respectivement 8257 et 6723 éléments.

La table 4.6 décrit les données utilisées et fournit des informations sur leurs attributs, leurs tailles et le nombre de classes.

Base de données	*Nombre d'éléments*	*Nombre d'attributs*	*Nombre de classes*
Lymphography	148	18	4
Diabetes	768	8	2
Indian	583	10	2
Haberman's Survival	306	3	2
BCW	699	9	2
Post-operative Patient	90	8	3
Parkinsons	197	23	2
EEG Eyes State	14980	14	2

Table 4.6 Description des ensembles de données utilisées pour nos tests**.**

Dans un premier temps, nous cherchons s'il y a des possibles valeurs aberrantes dans les ensembles de données considérés. Pour cela, nous calculons le degré de proximité pour les objets et recherchons les petites valeurs [Dik, 2014].

Une fois les possibles valeurs aberrantes déterminées et isolées, on exécute l'algorithme proposé RDUFL sur les autres données en examinant un objet à chaque itération. A la fin de la phase d'apprentissage, on obtient les classes détectées avec leurs prototypes. L'approche proposée permet de détecter le nombre exact des classes pour tous les exemples de données considérés.

La phase de traitement des possibles valeurs aberrantes O_i permet d'obtenir les résultats décrits dans la table 4.7.

Base de données	*C détectée*	*Valeurs aberrantes possibles*	*Valeurs aberrantes vrais*
Lymphography	2	15	11
Diabetes	2	6	3
Indian	2	32	14
Haberman's Survival	2	12	6
BCW	2	18	0
Post-operative Patient	2	2	2
Parkinsons	2	9	3
EEG Eyes State	2	4	2

Table 4. 7 Nombre de classes détectées par RDUFL, et les valeurs aberrantes possibles et vraies.

Pour la base de données « *Lymphography* », il existe réellement quatre classes dont deux classes sont dites rares (classe 1 et 4) vu leur petite taille [Zengyou He, 2003]. La classe 1 contient deux éléments et la classe 4 en contient 4. L'algorithme RDUFL détecte 11 valeurs aberrantes comprenant les 6 valeurs aberrantes présentées dans [Zengyou He, 2003]. Robust-FCM en donne 5 dont seuls 2 éléments sont des vraies valeurs aberrantes et appartiennent aux classes rares. PCM ne reconnaît aucune classe rare de cet ensemble.

Pour la base de données « *Diabete*s », le nombre des valeurs aberrantes détectées par les méthodes basées sur le clustering est 6, et par les méthodes basées sur la densité est 8 [Mandhare, 2017]. L'algorithme RDUFL détecte 3 vraies valeurs aberrantes.

Pour la base de données BCW, le degré de proximité permet de détecter 18 possibles valeurs aberrantes qui ont été isolées et non prises en compte dans la phase d'apprentissage. L'algorithme détecte l'existence de deux classes. La comparaison de ces 18 possibles valeurs aberrantes avec les prototypes des classes détectées montre que ces éléments présentent

suffisamment de caractéristiques communes avec ces prototypes. Par conséquent, elles ne sont pas de véritables valeurs aberrantes.

Nous recréons une distribution très déséquilibrée des données BCW en choisissant un élément maligne sur six tel suggéré dans [He, 2005]. L'ensemble de données obtenu comprend 39 éléments malignes (8%) et 444 enregistrements bénins (92%). Pour montrer l'efficacité de l'algorithme RDUFL, on ne met pas en quarantaine toutes les valeurs aberrantes possibles d'un seul coup, mais on met en quarantaine certains de ces valeurs aberrantes possibles seulement. Pour évaluer l'apprentissage effectué, nous mesurons la sensibilité et la spécificité. La sensibilité permet de connaitre la capacité d'identifier correctement les patients malades, alors que la spécificité permet de déterminer la capacité d'identifier correctement les patients sains.

RDUFL détecte les anomalies de cet ensemble de données et identifie les valeurs aberrantes (qui forment des petites classes) pour chaque nombre choisi de possibles valeurs aberrantes. Nous rapportons les résultats dans la table 4.8

Valeurs aberrantes Possible	***Nombre des anomalies***	***% des anomalies***	***Sensibilité***	***Spécificité***
5	39	100	100%	80%
10	39	100	100%	79%
15	39	100	100%	78%
20	39	100	100%	77%
25	39	100	100%	76%
30	39	100	100%	72%
35	39	100	100%	70%
39	39	100	100%	67%

Table 4.8 Taux de sensibilité et de spécificité de RDUFL pour la base de données BCW modifiée.

La sensibilité est de 100%, c'est-à-dire égale à 1, ce qui signifie qu'il n'y a pas de faux négatifs. Autrement dit, l'algorithme a détecté tous les patients.

La spécificité diffère en fonction des possibles valeurs aberrantes laissées dans l'ensemble des données et non mises en quarantaine.

De plus, RDUFL peut améliorer considérablement les performances de l'apprentissage et permet d'avoir une amélioration de 30,61% dans la base de données BCW. Ces résultats montrent que l'approche adoptée améliore également la découverte de classes, même en absence de valeurs aberrantes. En effet, RDFUFL améliore l'apprentissage et donne un bon taux de reconnaissance, comme indiqué dans la table 4.9.

Bases de données	***FCM***	***PCM***	***Robust-FCM***	***RDUFL***
Lymphography	50.05%	64.87%	58.11%	**72.30%**
Diabetes	66.02%	37.11%	58.47%	**67. 89%**
Indian	30.37%	66.03%	46.14%	**70.65%**
Haberman's Survival	49.02%	50.33%	42.81%	**60.13%**
BCW	65.96%	61.52%	41.35%	**96.57%**
Post-operative Patient	61.11%	*	72.22%	**73.33%**
Parkinsons	74.36%	54.87%	69.04%	**75.90%**
EEG Eyes State	55.28%	49.71%	50.82%	**57.61%**

Table 4. 9 Taux de reconnaissance obtenus pour les données utilisées.
L'étoile * signifie que l'algorithme PCM n'a pas détecté les trois classes.

Nous constatons que l'algorithme RDUL a un taux de reconnaissance élevé dans tous les cas considérés. Ce taux arrive à 96,5% pour la base de données BCW. Pour les autres algorithmes, FCM atteint un taux de reconnaissance égale 74,36% pour la base de données Parkinsons et Robust-FCM atteint 72,22% pour la base des données « Post-operative Patient ».

Le pourcentage de précision est également calculé pour chaque algorithme, selon l'équation (3.13).

Les résultats des algorithmes considérés sont présentés dans la table 4.10. Selon cette table, on constate que RDUFL a un pourcentage de précision plus élevé que ceux obtenus par les algorithmes étudiés.

Base de données	***FCM***	***PCM***	***Robust-FCM***	***RDUFL***
Lymphography	*	*	*	**70%**
Diabetes	**66 %**	44 %	61%	**66%**
Indian	69%	66%	45%	**76%**
Haberman's Survival	48%	49%	42%	**59%**
BCW	**95%**	65%	76%	**95%**
Post-operative Patient	*	*	*	**70%**
Parkinsons	74%	55%	73%	**75%**
EEG Eyes State	55%	50%	51%	**58%**

Table 4.10 Taux de précision des algorithmes étudiés pour les données considérées**.**

Conclusion

Nous avons présenté dans ce chapitre deux approches pour détecter des valeurs aberrantes et partitionner l'ensemble des données. La première méthode proposée est une variante adaptée de l'algorithme FCM qui comprend deux étapes principales. La première étape est une méthode intuitive de prétraitement basée sur le concept de degré de proximité qui identifie les M objets susceptibles d'être des «possibles» valeurs aberrantes. La deuxième étape est considérée comme étape d'apprentissage où le nouvel algorithme POFCM est exécuté utilisant les M possibles valeurs aberrantes détectées lors de la première phase comme centres en plus des c centres choisis aléatoirement. L'algorithme POFCM vérifie s'il existe des objets dans les classes dites aberrantes autres que les valeurs aberrantes et décide si, dans l'affirmative, elles ne sont pas de «véritables» valeurs aberrantes. Les résultats expérimentaux montrent que notre approche améliore l'apprentissage non supervisé des données contenant des valeurs aberrantes.

La seconde approche, robuste et dynamique, est constituée de trois étapes. La première étape identifie les objets qui peuvent être considérés comme des "possibles" valeurs aberrantes. La seconde étape est un algorithme d'apprentissage flou non supervisé qui détecte les classes existantes formées par les données sans les possibles valeurs aberrantes. L'algorithme dans cette étape fournit également les prototypes de ces classes détectées et les degrés d'appartenance de chaque objet à ces classes. La création des classes se fait selon un seuil dynamique recalculé à chaque itération de l'algorithme. Ce seuil se base sur la similarité entre les prototypes des classes détectées et il est mis à jour à l'exploration de tout nouvel objet. Quant à la dernière étape, elle consiste à traiter ces valeurs aberrantes possibles en se basant sur la similarité de chacune d'elles au plus loin objet de la classe correspondant à son plus proche prototype. Les résultats expérimentaux ont montré l'efficacité de l'approche proposée, surtout pour les ensembles de données qui contiennent des valeurs aberrantes.

Conclusion générale

Les travaux que nous avons exposés dans ce mémoire se situent dans le domaine de l'apprentissage flou non supervisé. Ce dernier est souvent confronté à deux principaux problèmes. Le premier concerne le chevauchement des classes de données, lorsque les limites entre les classes d'une partition sont fortement ambiguës et males définies, et où l'incertitude et la difficulté à prendre une décision sont grandes, réduisant par-là l'efficacité de la méthode d'apprentissage. Le second problème concerne la présence des valeurs aberrantes dans les données. Ces dernières déséquilibrent l'apprentissage en se voyant accorder une importance plus grande qu'elles n'ont.

Plusieurs techniques d'apprentissage flou non supervisé ont été proposées dans la littérature pour pallier ce type de problèmes. Néanmoins, la plupart de ces techniques nécessitent le choix de certains paramètres qui affectent fortement l'apprentissage et peuvent souvent induire des résultats erronés ou inacceptables. En plus, l'absence de méthode ou de règle qui assure le choix de la meilleure technique d'apprentissage flou non supervisé laisse le problème ouvert et difficile.

Nos travaux de recherche avaient pour objectif de contribuer à réduire l'impact de ces deux problèmes et proposer de nouveaux algorithmes dynamiques qui dépendent des similarités des données. Ainsi, nous nous sommes penchés d'abord sur la notion de similarité et son utilité pour représenter la similitude et la proximité des d'objets et de ce fait son rôle crucial pour former les classes qui existent. Ensuite nous avons contribué à la recherche de différents algorithmes dans le contexte d'apprentissage flou non supervisé, en proposant de nouvelles solutions permettant de réduire l'impact de ces deux problèmes.

La première solution, que nous avons appelée IUFL, introduit une nouvelle règle d'apprentissage qui a trait au cas du chevauchement aigu des classes, où il s'avère difficile d'émettre une décision sur l'appartenance d'un objet à une classe particulière si cet objet n'apporte pas suffisamment d'informations. Ceci a permis de réduire l'impact de certains objets à analyser et que l'on peut considérer comme des « *mauvais éléments* » en déférant leur apprentissage tant qu'une condition n'a pas été respectée. Cette règle d'apprentissage confère

à l'algorithme un aspect dynamique en ce qu'elle dépend du nombre de classes qui varie au cours du processus d'apprentissage.

La deuxième solution consiste à considérer les valeurs aberrantes lors de l'apprentissage, en introduisant la notion de "*possibles*" valeurs aberrantes basée sur le concept de degré de proximité. Cette solution nous a permis de concevoir deux robustes algorithmes d'apprentissage flou non supervisé. Le premier algorithme est une variante du FCM qu'on a baptisé POCFCM. Le second algorithme, appelé RDUFL, se base sur un seuil dynamique recalculé automatiquement à chaque itération de l'algorithme, et qui dépend de la similarité des prototypes. De ce fait, la création des classes se fait selon ce seuil dynamique.

Ces approches ont été appliquées à des problèmes différents qui constituent des références de comparaison des différentes approches d'apprentissage flou non supervisé. Conséquemment, les résultats obtenus par les approches proposées ont montré leurs performances et leurs efficacités par rapport aux autres approches usuelles. Ces dits résultats montrent clairement que le taux de reconnaissance et le taux de précision évalués selon nos approches sont généralement supérieurs à ceux des autres approches étudiées dans ce travail.

A titre d'exemple, la comparaison des taux de précisions calculés correspondant à IUFL et FCM qui sont respectivement 0.93 et 0.66, pour la même base de données BCW confirme davantage la validité des résultats de l'algorithme IUFL. En plus, la qualité de la partition générée par IUFL est généralement plus bonne que celles générées par d'autres techniques. En effet, IUFL fournit des prototypes qui sont très proches des centres réels. A cet égard, les distances entres les centres réels et les prototypes détectés par IUFL sont inférieures à celles qui existent entre les centres réels et les prototypes détectés par les autres algorithmes.

De même, l'application de RDUFL à la base de données BCW donne un taux de reconnaissance égale à 96.57% alors que Robust-FCM en donne 41.35% et PCM donne 61.52%. En plus, RDUL détecte les valeurs aberrantes dans plusieurs bases de données, alors que Robust-FCM ne les détecte pas toutes, notamment dans la base de données « *Lymphography* » où Robust-FCM ne détecte que deux valeurs aberrantes.

Finalement, nous avons appliqué les algorithmes conçus dans le cadre de cette thèse aux données du monde médical, notamment la segmentation des images IRM, et nous avons abouti à des résultats très satisfaisants. En plus, la combinaison de l'algorithme IUFL et FCM,

et celle de RDUFL et FCM, ont donné de bons résultats concernant la segmentation de ces images. En effet, les deux algorithmes proposés ont permis de générer des prototypes qui ont constitué des bons centres initiaux pour appliquer FCM.

Plusieurs perspectives d'amélioration et de développement de ce travail sont envisagées. Le premier travail futur sera proposé de manière à considérer simultanément les deux algorithmes IUFL et RDUFL, et proposer un algorithme qui intègre leurs différents apports pour procéder à un apprentissage robuste et dynamique et sans paramétrage. Le second travail futur introduira la notion « *d'informatique granulaire* » pour quantifier l'imprécision et la tolérance de l'incertitude. En effet, la granularité permet la simplification, la clarté, et la tolérance à l'incertitude [Lingras, 2016]. Cela «*sous-tend la remarquable capacité humaine à prendre des décisions rationnelles dans un environnement imprécis*» [Lingras, 2016]. De ce fait, un nouvel robuste algorithme d'apprentissage flou non supervisé utilisant des techniques informatiques granulaires sera développé.

REFERENCES BIBLIOGRAPHIQUES

[Abouelala, 2002] O.Abouelala, «Elaboration d'un ensemble d'algorithmes flous de classification non supervisée de données multidimensionnelles : Application à la segmentation d'images», Thèse de Doctorat, Faculté des Sciences, Rabat, Université Mohamed V, Agdal, Décembre 2002.

[Aggarwal,2001] C.C. Aggarwal, A. Hinneburg and D.A. Keim. «On the surprising behavior of distance metrics in high dimensional space». *Lecture Notes in Computer Science*, 1973:420–434, 2001.

[Al-Zoubi, 2006] M. B. Al-Zoubi, A. M. Kamel and M. J. Radhy. «Techniques for image enhancement and noise removal in the spatial domain», *WSEAS Transactions on Computer*s, Vol. 5, No. 5, pp. 1047-1052, 2006.

[Al-Zoubi, 2010] M. B.Al-Zoubi, A. Al-Dahoud, A.A. Yahya. «Fuzzy clustering-based approach for outlier detection», *Proceeding ACE'10 Proceedings of the 9th WSEAS international conference on Applications of computer engineering*, pp. 192-197, Penang, Malaysia. 2010.

[Angiulli, 2006] F. Angiulli, S. Basta and C. Pizzuti. «Distance-based detection and prediction of outliers». *IEEE Transactions on Knowledge and Data Engineering*, Vol 18, pp.145-160, 2006.

[Antonelli, 2016] M. Antonelli, P. Ducange B. Lazzerini and F. Marcelloni. «Multi-objective evolutionary multiplicative aggregation in group decision making design of granular rule-based classifiers», *Granular Computing* 1 (1), pp. 37-58, 2016.

[Arenas-Garcia, 2007] J. Arenas-Garcia, E. Parrado-Hernandez, A.Meng, L. K.Hansen and J.Larsen. «Discovering music structure via similarity fusion». *In Music, Brain and Cognition Workshop*, NIPS'07, Whistler, Canada. 2007.

[Asuncion, 2007] A. Asuncion and D.J. Newman. «UCI machine learning repository», Irvine, CA: University of California, School of Information and Computer Science, 2007. Available: http://archive.ics.uci.edu/ml/datasets.html.

[Ball, 1966] G. H. Ball and D. J. Hall. «ISODATA An iterative method of multivariate analysis and pattern classification», *the Int. Commun. Conf*., Philadelphia, Pa. 1966.

[Bandemer, 1992] H. Bandemer and W. Näther. «Fuzzy data analysis». *Kluwer Academic*

Publishers, Dordrecht, Boston, London, 1992.

[Barnett, 1994] V. Barnett and T. Lewis. «Outliers in statistical data». *John Wiley*. Chichester. 1994.

[Ben-David, 2014] S. Ben-David and N. Haghtalab. «Clustering in the presence of background noise». In *Proceedings of the 31st International Conference on Machine Learning*, Vol 32, pp. 280-288, Beijing, China. 2014.

[Benrabh , 2005] M. Benrabh, A. Bouroumi and A. Hamdoun. «A fuzzy validity-guided procedure for cluster detection». *Malaysian Journal of Computer Science*, Vol 18, No 1, pp.31-39, 2005.

[Bensaid , 1999] A. M. Bensaid, L. O. Hall, J. C. Bezdek, L.P. Clarke, and M. L. Silbeger, «Validity-guided (re)clustering with applications to image segmentation», *IEEE Trans. Fuzzy Systems*, vol. 4, no. 2, pp. 112-123, May 1999.

[Berthold, 1999] M. Berthold. «Fuzzy models and potential outliers». *In proceedings 18th International Conference of the North American Fuzzy Information Processing Society, NAFIPS*, pp. 532-535. IEEE Press. New York, U.S.A. 1999.

[Bezdek, 1981] J. C. Bezdek. «Pattern recognition with fuzzy objective function algorithms». *Plenum Press, New York*, 1981.

[Bezdek, 1984] J. C. Bezdek. «FCM: The fuzzy c-means clustering algorithm», *Computers & Geoscien*ces, Vol. 10, No. 2-3, pp. 191-203, 1984.

[Blake, 1998] C. L. Blake, C. J. Merz. «UCI repository of machine learning databases», http://www.ics.uci.edu/mlearn/MLRepository.html, University of California, Irvine, Department of Information and Computer Sciences, 1998.

[BoGao, 2004] X. BoGao, «Fuzzy cluster analysis and its applications», *Xian Electronic Technology University Press*, 2004.

[Bouroumi, 2000] A. Bouroumi, M. Limouri and A. Essaïd. «Unsupervised fuzzy learning and cluster seeking», Intelligent Data Analysis, Vol. 4, No. 3-4, pp. 241-253, December 2000.

[Bolton, 2002] R. J. Bolton and D. J. Hand. «Statistical fraud detection: a review». *Statistical Science* 17, pp. 235-255, 2002.

[Bradley, 1998] P. S. Bradley and U. M. Fayyad. «Rening initial points for k-means clustering». *In: Proc. 15th International Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA*, pp. 91-99. 1998.

URL: citeseer.ist.psu.edu

/bradley98refining.html.

[Breunig 2000] M.Breunig , H. Kriegel, R. Ng and J. Sander. «Lof: identifying density-based local outliers». *In Proceedings of 2000 ACM SIGMOD International Conference on Management of Data*. ACM Press, pp.93–104, 2000.

[Cabanes, 2010] G. CABANES, «Classification non supervisée à deux niveaux guidée par le voisinage et la densité», Thèse de doctorat sous la direction du Pr. Younès Bennani, Université Paris13, 2010.

[Candillier, 2006] L. Candillier. «Contextualisation, visualisation et évaluation en apprentissage non supervise», thèse de Doctorat en Informatique, Université Charles de Gaules – Lille 3, Septembre 2006.

[Cha, 2006] S-H. Cha, C.Tappert, and S.Yoon. «Enhancing binary feature vector similarity measures», *Journal of Pattern Recognition Research* 1, pp.63-77, 2006.

[Chelloug, 2006] Chelloug Samia. «Calcul amorphe pour la classification des données par GNG: application à la segmentation d'images et à la bioinformatique», Thèse de Magister, Faculté des sciences de l'ingénieur, Université Mentouri de Constantine, Algérie. 2006.

[Chen, 2009 - a] S. M. Chen, N. Y. Wang and J. S. Pan. «Forecasting enrollments using automatic clustering techniques and fuzzy logical relationships», *Expert Systems with Applications*, vol 36, no. 8, pp.11070-11076, 2009.

[Chen, 2009 - b] S. M. Chen and J. H. Chen. «Fuzzy risk analysis based on similarity measures between interval-valued fuzzy numbers and interval-valued fuzzy number arithmetic operators», *Expert Systems with Applications*, vol 36, no. 3, pp. 6309-6317, 2009.

[Chen, 2011] S. M. Chen and C. Y. Chien. «Parallelized genetic ant colony systems for solving the traveling salesman problem». *Expert Systems with Applications*, vol 38, no. 4, pp.3873-3883, 2011.

[Chen, 2013] S. M. Chen and P. Y. Kao. «TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines». *Information Sciences*, vol 247, pp.62-71, 2013.

[Chenglong, 2011] T. Chenglong. «Clustering of steel strip sectional profiles based on robust adaptive fuzzy clustering algorithm», *Computing and Informatics*, Vol 30,

pp.357–380, 2011.

[Cimino 2007] M. G. C. A. Cimino, G. Frosini, B. Lazzerini and F. Marcelloni. «On the noise distance in robust fuzzy c-means», *International Journal of Computer, Information, Systems and Control Engineering*, Vol.1, No.1, 2007.

[Ciucci, 2016] D. Ciucci. «Orthopairs and granular computing», *Granular Computing* 1 (3), pp.159-170, 2016.

[Clifford, 1975] D. H. T. Clifford and W. Stephenson. «An introduction to numerical classification». *New York: Academic*, 1975.

[Cristani, 2007] M. Cristani, M.Bicego, and V.Murino. «Audio-visual event recognition in surveillance video sequences». *IEEE Transactions on Multi-media*, 9(2). pp. 257–267, 2007.

[Davé 1991] R. N. Davé. «Characterization and detection of noise in clustering», *Pattern Recognition Letters*, vol. 12, no. 11, pp. 657-664, 1991.

[Davé 1997 - a] R. N. Davé, R. Krishnapuram. «Robust clustering methods: A unified view», *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 2, pp. 270-293, 1997.

[Davé 1997 - b] R. N. Davé, S. Sen. «Noise clustering algorithm revisited», *NAFIPS'97*, 21-24, pp. 199-204, New York, U.S.A. September 1997.

[Dasarathy , 1990] B. V. Dasarathy, «Nearest neighbor (NN) norms: NN», *Pattern Classification Techniques*. 1990.

[D'Hondt , 2007] F. D'Hondt et B. El Khayati. «Etude de méthodes de clustering pour la segmentation d'images en couleurs», Faculté Polytechnique de Mons, 5ème Electricité, Certificat Applicatifs Multimédia. Novembre 2007, 5p.

[Deneshkumar, 2014] V. Deneshkumar, K. Senthamaraikannan, M. Manikandan. «Identification of outliers in medical diagnostic system using data mining techniques», *International Journal of Statistics and Applications*, 4(6), pp.241-248. 2014

[Devijver, 1980] Devijver PA, Kittler J. «On the edited nearest neighbor rule». *In: Proc 5th Int Conf on Pattern Recognition. Los Alamitos, CA: IEEE Computer Society Press*; pp. 72-80. 1980.

[Digby, 1987] P.G.N. Digby and Kempton R.A. «Multivariate analysis of ecological communities». *Chapman and Hall, Population and Community Biology Series*, London. 1-205, 1987.

[Dietterich, 2009] T. Dietterich. «Machine learning for sequential data: A review». *Proceedings of*

the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, p.15-30, Windsor, Ontario, Canada. 2002.

[Dik, 2009] A. Dik, A. El moujahid, A. Bouroumi, A. Ettouhami. «Etude comparative de mesures de similarités», RNCJP6, Faculté des sciences Ben'Msik, Casablanca, 2009.

[Dik, 2012] A. Dik, A. Bouroumi, A. Ettouhami. «Improving fuzzy clustering by r-metric», MNOTSI 2012, ENSA – Kenitra.

[Dik, 2014 a] A. Dik, A. El moujahid, A. Bouroumi, A. Ettouhami. «Weighted distances for fuzzy clustering», *in Applied Mathematical Sciences*, vol 8, N°4, pp.147- 156, 2014.

[Dik, 2014 b] A. Dik, K. Jebari, A. Bouroumi, A. Ettouhami. «Fractional metrics for fuzzy c-means», *International Journal of Computer and Information Technology*, Volume 03 – Issue 06, pp.1490 – 1495. November 2014.

[Dik, 2014 c] A. Dik, K. Jebari, A. Bouroumi, A. Ettouhami. «A new fuzzy clustering by outliers», *Journal of Engineering and Applied Sciences*, Vol 9 (10-12), pp. 372-377, 2014.

[Dik, 2015] A. Dik, A. El moujahid, A. Ettouhami. «A new dynamic algorithm for unsupervised learning», *International Journal of Innovative Computing, Information and Control*, Vol 1,1 No 4, pp.1325-1339, 2015.

[Dik, 2018] A. Dik, K. Jebari, A. Ettouhami, «An improved robust fuzzy algorithm for unsupervised learning», *Journal of Intelligent System*, 2018.

https://doi.org/10.1515/jisys-2018-0030.

[Doherty, 2004] K.A.J.Doherty, R.G.Adams and N.Davey. «Non-euclidean norms and data normalisation», *ESANN'2004 proceedings - European Symposium on Artificial Neural Networks, Bruges (Belgium)*, pp. 181-186, 28-30, April 2004.

[Damodar, 2012] R. Damodar and K. J. Prasanta, «Initialization for K-means clustering using Voronoi diagram», *Procedia Technology* 4, *Elsevier*, 395 – 400, 2012.

[Dubois, 2016] D. Dubois and H. Prade. «Bridging gaps between several forms of granular computing», *Granular Computing* 1 (2), pp.115-126, 2016.

[Duda, 1973] R. O. Duda and P. E. Hart. «Pattern Classification and Scence analysis». Wiley-Interscience, New York, 2000

[Duda, 2000] R. O. Duda, P. E. Hart, and D. G. Stork. «Pattern Classification». Wiley, New

York, 1973.

[Dunn , 1973] J. C Dunn. «A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters», *J. Cybernetics*, vol. 3, No. 3, pp. 32-57, 1973.

[El Imrani, 2000] A. El Imrani, A. Bouroumi, M. Limouri and A. Essaid. «A coevolutionary genetic algorithm using fuzzy clustering». *International Journal of Intelligent Data Analysis*, Vol 4, pp.183-193, 2000.

[Eltibi, 2011] M. F. Eltibi and Wesam M. Ashour. «Initializing K-means clustering algorithm using statistical information», *International Journal of Computer Applications* (0975 – 8887), Volume 29– No.7, pp: 51 –55, September 2011.

[El Ferchichi, 2013] S. El Ferchichi, «Sélection et extraction d'attributs pour les problèmes de classification». Thèse de doctorat sous la direction du S.Maouche, Université des Sciences et Techniques, Lille1. 2013.

[Everitt, 1993] B. S. Everitt, «Cluster analysis», Edward Arnold, London, third edition, 1993.

[Forestier, 2010] G.Forestier, «Connaissances et clustering collaboratif d'objets complexes multisources». Thèse de Doctorat, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur, Université de Strasbourg, Septembre 2010.

[Forgy, 1965] E. Forgy. «Cluster analysis of multivariate data: Efficiency vs. interpretability of classification». *Biometrics* 21, Vol.3, pp.768-769. 1965.

[Francois, 2007] D. Francois, V. Wertz, and M. Verleysen. «The concentration of fractional distances». *IEEE Transactions on Knowledge and Data Engineering*, 17(7), pp.873–886, 2007

[Fukuyama, 1989] Y. Fukuyama and M. Sugeno, «A new method of choosing the number of clusters for the fuzzy c-means method», in *Proc. 5th Fuzzy Syst. Symp*., pp. 247-250, 1989.

[Garcia-Garcia, 2010] D. Garcia-Garcia, E.Parrado-Hernandez, J.Arenas-Garcia and F.Diaz-de-Maria, «Music genre classification using the temporal structure of songs». *In IEEE International Workshop on Machine Learn-ing for Signal Processing*. Kittilä, Finland. 2010.

[Gosaina, 2016] A. Gosaina and S. Dahiya. «Performance analysis of various fuzzy clustering algorithms: A review». *Procedia Computer Science*, Vol 79, pp.100 – 111, 2016.

[Gower,1986] J. C. Gower and P. Legendre. «Metric and euclidean properties of dissimilarity coefficients». *Journal of Classification*, 3, pp.5–48, 1986.

[Gretton, 2005] A.Gretton, R.Herbrich, A.Smola, O.Bousquet, and B.Scholkopf. «Kernel methods for measuring independence». *Journal of Machine Learning Research (JMLR)*. 6, pp. 2075–2129, 2005.

[Guaus, 2009] E. Guaus, «Audio content processing for automatic music genre classification: descriptors, databases, and classifiers». PhD thesis, Universitat Pompeu Fabra. Barcelone, Espagne.2009.

[Guérif, 2006] S.GUÉRIF, «Réduction de dimension en apprentissage numérique non supervisé», Thèse de doctorat sous la direction du Pr. Y. Bennani, Université Paris13, *2006*.

[Gustafson, 1978] Gustafson D.E., Kessel W.C. «Fuzzy clustering with a fuzzy covariance matrix», *IEEE Conference. On Adaptive Processes*, (17) No. 1, pp.761-766, 1978.

[Halkidi, 2001] M.Halkidi, Y.Batistakis, and M.Vazirgiannis, «On clustering validation techniques». *Journal of Intelligent Information Systems*, 17, pp.107-145, 2001.

[Haidar, 2005] S.Haidar, «Comparaison des documents audiovisuels par matrice de similarité», thèse de doctorat sous la direction de R. André-Obrecht, Université Toulouse III - Paul Sabatier, Toulouse. 2005.

[Han, 2006] J.Han and M. Kamber, «Data mining: concepts and techniques», *Morgan Kaufmann*, 2nd ed. 2006.

[Hawkins, 1980] D. Hawkins. «Identifications of outliers», *Chapman and Hall*, London, 1980.

[He, 2003] Z. He, X. Xu and S. Deng, «Discovering cluster-based local outliers». *Pattern Recognition Letters*, Vol 24, pp.1641–1650, 2003.

[He, 2005] Z. He, S. Deng and X. Xu. «An optimization model for outlier detection in categorical data». *In: Huang DS., Zhang XP., Huang GB. (eds) Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science*, Vol 3644. Springer, Berlin, Heidelberg. 2005.

[Hinneburg,1999] A. Hinneburg, C.C. Aggarwal, and D.A. Keim. «What is the nearest neighbor in high dimensional spaces»? *Lecture Notes in Computer Science*, 1540, pp. 217–235, 1999.

[Horng, 2005] Y. J. Horng, S. M. Chen, Y. C. Chang and C. H. Lee. «A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques», *IEEE Transactions on Fuzzy Systems*, vol 13, no. 2, pp.216-228. 2005.

[Jain 2009] A.K.Jain. «Data clustering: 50 years beyond k-means», *Pattern Recognition Letters*, Vol 31, pp. 651–666, 2009.

[Jansen, 2007] S.M.H. Jansen. «Customer segmentation and customer profiling for a mobile telecommunications company based on usage behavior, a Vodafone case study», Thèse de Magister, Vodafone Maastricht, Nederland.2007.

[Jeongmin, 2012] Y. Jeongmin, L.Sung-Hee and J.Moongu. «An adaptive aco-based fuzzy clustering algorithm for noisy image segmentation», *International Journal of Innovative Computing Information and Control*, Volume 8, Number 6, pp. 3907-3918, June 2012.

[Jollois, 2003] F-X. Jollois. «Contribution de la classification automatique à la fouille de données», Thèse de Doctorat de l'Université de Metz, spécialité Informatique, 2003.

[Jolion 1989] J-M. Jolion and A.Rosenfeld. «Cluster detection in background noise», *Pattern Recognition*, Vol. 22, No. 5, pp. 603 607, 1989.

[Karthikeyani, 2009] N. Karthikeyani Visalakshi, J. Suguna. «K-means clustering using max-min distance measure», *The 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009), Cincinnati, Ohio, USA* - 17, 2009.

[Kaufman, 1990] Kaufman L, P.J. Rousseeuw. «Finding groups in data». *Prentice-Hall, Englewood Cliffs*, NJ, 1990.

[Khodja, 1997] L.Khodja. «Contribution à la classification floue non supervisée», Thèse de doctorat de l'Université de Savoie Mont Blanc, Spécialité: Electronique - Electrotechnique – Automatique, 1997.

[Knorr 1998] E.M. Knorr and R.T. Ng. «Algorithms for mining distance-based outliers in large dataset». *Proceedings of the 24rd International Conference on Very Large Data Bases*, August 24-27, San Francisco, CA., USA., pp:392-403. 1998.

[Krishnapuram, 1993] R. Krishnapuram and J. Keller. «A possibilistic approach to clustering», *IEEE Trans. Fuzzy Syst*., vol.1, no.2, pp.98-110, 1993.

[Lane, 1999] Lane, T. and C. E. Brodley. «Temporal sequence learning and data reduction for anomaly detection», *ACM Transactions on Information and System Security*,

Vol. 2, No. 3, pp. 295-331, 1999.

[Laetitia, 2003] J. Laetitia. «Métaheuristiques pour l'extraction de connaissances: application à la génomique», Thèse de Doctorat, Université des Sciences et Technologies de Lille, 26 Novembre 2003.

[Lazar, 2008] C.Lazar. «Méthodes non supervisées pour l'analyse des données multivariées». Thèse de doctorat sous la direction du D.Nuzillard, Université de Reims Champagne, Ardenne, 2008.

[Legendre, 1984] L. Legendre et P. Legendre, «Ecologie numérique. Tome 2 - La structure des données écologiques». *Masson, Paris. 2ème édition revue et augmentée.* pp.1-344. 1984.

[Legendre, 1998] P. Legendre et L. Legendre. «Numerical ecology». *Second English Edition. Development in Environmental Modelling*, 20. Elsevier. 853 pp. 1998.

[Lesot, 2009] M-J. Lesot, M. Rifqi and H. Benhadda. «Similarity measures for binary and numerical data: a survey», *Int. J. Knowledge Engineering and Soft Data Paradigms*, Vol. 1, No. 1, pp. 63-84. 2009.

[Lingras, 2016] P. Lingras, F. Haider and M. Triff. «Granular meta-clustering based on hierarchical, network, and temporal connections», *Granular Computing* 1 (1), pp.71-92. 2016.

[Livi, 2016] L. Livi and A. Sadeghian. «Granular computing, computational intelligence, and the analysis of non-geometric input spaces», *Granular Computing* 1 (1), pp.13-20. 2016.

[Loureiro 2004] A.Loureiro, L. Torgo and C. Soares. «Outlier detection using clustering methods: a data cleaning application», in *Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector*. Bonn, Germany, 2004.

[MacQueen, 1967] J,B. MacQueen, «Some methods for classification and analysis of multivariate observations», *in Proceeding of the Symposium on Mathematics and Probability, 5th, Berkley, University of California Press*, vol. 1, pp: 281-297, 1967.

[Mandhare, 2017] Harshada C. Mandhare et al, «Comparative analysis with implementation of cluster based, distance based and density based outlier detection techniques using different healthcare datasets». *International Journal of Advanced Research in Computer Science*, 8 (5), pp.1349-1355, May-June 2017.

[Ménard , 1998] Ménard. M, «The fuzzy c+2 means: Solving the extended ambiguity reject in clustering», *in IEEE Transactions on fuzzy systems*, vol.1, N° .2, pp. 195-203, 1998.

[Morsier, 2015] F. Morsier, D. Tuia, M. Borgeaud, V. Gass and J.P. Thiran. «Cluster validity measure and merging system for hierarchical clustering considering outliers». *Pattern Recognition.* Vol 48, No 1, pp. 1478-1489. 2015.

[Nadernejad, 2011] E. Nadernejad and A. Barari. «A novel pixon-based image segmentation process using fuzzy filtering and fuzzy c-mean algorithm», *International Journal of Fuzzy Systems*, Vol. 13, No. 4, pp:350 – 357, December 2011.

[Naveen, 2018] A. Naveen and T. Velmurugan, «Clustering techniques on brain MRI», *Indian Journal of Public Health Research & Development*, Vol.9, No. 2, pp: 430 - 435, February 2018.

[Nazari, 2018] Z.Nazari and D.Kang. «Evaluation of multivariate outlier detection methods with benchmark medical datasets», J*CSNS International Journal of Computer Science and Network Security*, Vol.18, No.4, April 2018.

[Ohashi 1984] Y.Ohashi, «Fuzzy clustering and robust estimation», *Proceedings of 9th Meeting SAS UserGroup Int*, Hollywood Beach, Florida, 1984.

[Ott, 2014] L.Ott, L. Pang, F. Ramos and S. Chawla. «On integrated clustering and outlier detection». *Advances in Neural Information Processing Systems,* 27, pp.1359-1367, 2014.

[Pal, 1995] N.R. Pal, and J.C. Bezdek, «On cluster validity for the fuzzy c-means model», *IEEE Trans. On Fuzzy Systems*, Vol. 3, no 3, pp: 370 – 379, August 1995.

[Pal, 2005] N.R. Pal, K. Pal, J.M. Keller and J.C. Bezdek. «A possibilistic fuzzy c-means clustering algorithm». *IEEE Transactions on Fuzzy Systems,* Vol 13, No 4, pp. 517 – 530. 2005.

[Parizeau, 2004] M. Parizeau, Réseaux de neurones, Université LAVAL, Automne 2004.

[Peters, 2016] G. Peters and R. Weber. «DCC: A framework for dynamic granular clustering», *Granular Computing* 1 (1), 1-11, 2016.

[Philipps, 1995] W. Philipps, R. Velthuizen, S. Phuphanich, L. Hall, L. Clarke, and M. Silbiger. «Application of fuzzy c-means algorithm segmentation technique for tissue differentiation in MR images of a hemorragic glioblastoma multiforme». *Magnetic Resonance Imaging*, vol. 13, pp. 277–290, 1995

[RAMMAL, 2010] A. RAMMAL, «Modélisation multi-agent dans un processus de gestion multi acteur, application au maintien à domicile», Thèse de Doctorat, l'Université Toulouse III - Paul Sabatier, Décembre 2010.

[Ramaswamy, 2000] S.Ramaswamy, Rastogi and R.Kyuseok, «Efficient algorithms for mining outliers from large data sets». *In: Proceddings of SIGMOD'00, Dallas, Texas*, pp. 93-104. 2000.

[Rand, 1971] W. M. Rand, «Objective criteria for the evaluation of clustering methods», *Journal of the American Statistical Association*, 66 (336), pp. 846–850, 1971.

[Rehm, 2007] F. Rehm, F. Klawonn and R. Kruse. «A novel approach to noise clustering for outlier detection». *Soft Computing*. Vol 11, No 5, 489-494, 2007.

[Rezaee, 1998] M. R. Rezaee, B. P. F. Lelieveldt and J. H. C. Reiber, «A new cluster validity index for the fuzzy c-mean», *Pattern Recognition Letters* 19, Elsevier, pp:237–246, 1998.

[Roux, 1985] M. Roux, «Algorithmes de classification», 151 p., Masson, Paris, 1985.

[Ruspini, 1969] E. R. Ruspini, «A new approach to clustering», *Inform. Control*, vol. 15, no. 1, pp. 22-32, July 1969.

[Skowron, 2016] A. Skowron, A. Jankowski and S. Dutta. «Interactive granular computing», *Granular Computing* 1 (2), 95-113, 2016.

[Sneath, 1973] P. H. A. Sneath. and R. R. Sokal, «Numerical taxonomy». *London: Freeman*, 1973.

[Sun, 2004] H. Sun, S. Wang and Q. Jiang, «FCM-based model selection algorithms for determining the number of clusters», *Pattern Recognition* 37, 2027–2037, 2004.

[Suganya, 2012] R. Suganya, R. Shanthi. «Fuzzy c- means algorithm- A review», *International Journal of Scientific and Research Publications*, Vol 2, Issue 11, November 2012.

[Tang, 2012] C. Tang, S. Wang and Y. Chen. «Clustering of steel strip sectional profiles based on robust adaptive fuzzy clustering algorithm». *Comput. Inform.*, 30, 357-380. 2012.

[Tou, 1974] J.T. Tou, R.C. Gonzales. Pattern recognition principles. Reading, MA. Addison-Wesley, 1974.

[Tsai, 2008] P.W. Tsai, J.S. Pan, S.M. Chen, B.Y. Liao and S.P. Hao. «Parallel cat swarm optimization», *In Proceedings of the seventh International Conference on*

Machine Learning and Cybernetics, Kunming, China, Vol 6, pp.3328-3333. 2008.

[Turenne, 2006] N.Turenne, « Méthode des KNN ».Cours assuré à *INRA.* 2006. disponible sur : http://exorciste2.free.fr/Amine/nouveau%20dossier/MOAB/coursDM_KNN.pdf

[Wang, 2009] Q. Wang, S.Kulkarni, and S.Verdu, «Divergence estimation for multidimensional densities via k-nearest-neighbor distances». *IEEE Transactions on Information Theory*, 55, pp: 2392–2405. 2009.

[Wang, 2017] G. Wang, J. Yang and J. Xu, «Granular computing: from granularity optimization to multi-granularity joint problem solving», *Granular Computing* 2 (3), 105-120. 2017.

[Weng, 2007] W. Weng and Y. Zhang, «On fuzzy cluster validity indices», *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2095-2117, October 2007.

[West, 2006] K.West, S.Cox, and P.Lamere, «Incorporating machine learning into music similarity estimation». *In Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, ACM, pp. 89–96. 2006.

[Windham , 1981] M.P. Windham. «Custer validity for fuzzy clustering algorithm». *Fuzzy sets and Systems*, vol.5, pp. 177- 185, 1981.

[Wilson , 1972] Wilson DL. «Asymptotic properties of nearest neighbor rules using edited data». *IEEE Trans Systems Man Cybernet*; SMC-2. pp.408-421. 1972.

[Xie , 1991] X. L. Xie and G. A. Beni, «Validity measure for fuzzy clustering». *IEEE Trans. Pattern Anal. Machine Intell*. Vol. 3 No. 8, pp. 841-846, 1991.

[Xu , 2016] YJ. Xu, L. Chen, RM. Rodríguez, F. Herrera and HM. Wang, «Deriving the priority weights from incomplete hesitant fuzzy preference relations in group decision making», *Knowledge-Based Systems*, 99, pp.71-78. 2016.

[Xu , 2017] YJ. Xu, JF. Cabrerizo and E.Herrera-Viedma, «A consensus model for hesitant fuzzy preference relations and its application in water allocation management», *Applied Soft Computing*, 58, pp.265-284. 2017.

[Xu , 2018] YJ. Xu, CY. Li and XW Wen, «Missing values estimation and consensus building for incomplete hesitant fuzzy preference relations with multiplicative consistency», *International Journal of Computational Intelligence Systems*, vol 11, pp.101-119, 2018.

[Yao, 2016] Y. Yao, «A triarchic theory of granular computing», *Granular Computing* 1 (2), pp.145-157, 2016.

[Yu, 2012] J. Yu, S. H. Lee and M. Jeon, «An adaptive ACO-based fuzzy clustering algorithm for noisy image segmentation». *International Journal of Innovative Computing Information and Control*, Vol 8, No 6, pp.3907-3918. 2012.

[Zadeh, 1965] L. A. Zadeh. «Fuzzy sets». *Information and Control*. Vol 8, No 3, pp.338-353. 1965.

[Zhang, 2016] WC. Zhang, YJ. Xu and HM. Wang. «A consensus reaching model for 2-tuple linguistic multiple attribute group decision making with incomplete weight information», *International Journal of Systems Science*, 47(2), pp.389-405. 2016.

[Zengyou, 2003] Zengyou He, Xiaofei Xu, Shengchun Deng. «Discovering cluster-based local outliers», *Pattern Recognition Letters* 24, 1641–1650, 2003.

Résume (max 200 mots)

Nos travaux de recherche ont pour objectif, d'une part, de réduire l'impact du chevauchement des classes de données, lorsque les limites entre les classes d'une partition sont fortement ambiguës et males définies, et où l'incertitude et la difficulté à prendre une décision sont grandes, et d'autre part, à identifier les valeurs aberrantes qui peuvent déséquilibrer l'apprentissage en se voyant accorder une importance plus grande qu'elles n'ont. Ainsi, on a proposé de nouveaux algorithmes d'apprentissage flou non supervisé à partir de données non étiquetées et en présence d'éventuelles valeurs aberrantes. On s'est intéressé ainsi aux points suivants: 1) les mesures de similarités entre les données et leur rôle crucial pour former les classes, ainsi qu'à la caractérisation de ces classes par des prototypes, 2) la quantification de l'imprécision et la tolérance de l'incertitude dans le cas du chevauchement aigu des classes où il s'avère difficile d'émettre une décision dans un environnement imprécis sans avoir suffisamment d'informations, 3) l'impact des valeurs aberrantes sur l'apprentissage, et les techniques proposées dans la littérature pour pouvoir effectuer un apprentissage des données en présence des valeurs aberrantes. Les expériences menées sur des données du monde réel montrent l'efficacité des algorithmes proposés pour l'apprentissage des données et la gestion de l'incertitude.

Mots-clefs (5) : Apprentissage flou non supervisé, Mesure de Similarité, Clustering, Outliers, Degré de proximité.

Abstract (max 200 mots)

Our research aims to reduce the impact of overlapping data classes, when the boundaries between classes are highly ambiguous and not defined, and where uncertainty and the difficulty of making a decision is great. The second object is to identify outliers that can unbalance learning in the partition. Thus, new unsupervised fuzzy learning algorithms are proposed using unlabeled data and possibly outliers. We are interested to 3 topics: 1) measures of similarities between the data, their crucial role in class formation, and the characterization of these classes by prototypes, 2) the inaccuracy and tolerance of uncertainty in the case of acute class overlap where it is difficult to make a decision in an inaccurate environment without sufficient information; 3) the impact of outliers on learning, and the techniques proposed in the literature to perform data learning in the presence of outliers. Experiments conducted on real-world data show that the proposed algorithms present the effectiveness to learn the structure of data and manage the uncertainty.

Key Words (5) : Outliers, Similarity, Unsupervised learning, Clustering, Outliers, proximity