الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التـعليم العالي والبحث العلمي

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Thèse de Doctorat Science

Présentée à l'Université de Guelma
Faculté des Sciences et de la Technologie
Département d'Électronique et Communication/Télécommunication
Spécialité : Génie Électrique

Présentée par : BOUDOUDA Houria

Thème : Reconnaissance des Formes Incomplètement Définies

Sous la direction de : Prof. SERIDI Hamid

JURY

Prof : BOUKROUCHE Abdelhani	**Université de Guelma**	**Président**
Prof : SERIDI Hamid	**Université de Guelma**	**Rapporteur**
MC : MOUSSAOUI Abdelkrim	**Université de Guelma**	**Examinateur**
MC : KHOLLADI Mohamed-Khireddine	**Université de Constantine**	**Examinateur**
MC : CHAOUI Allaoua	**Université de Constantine**	**Examinateur**
MC : KAZAR Okba	**Université de Biskra**	**Examinateur**

2010

Résumé

Cette thèse se place dans le cadre du clustering flou, dont l'objectif consiste à chercher une partition d'un ensemble de formes incomplètement définies en classes les plus naturelles possible ou clusters sans aucune connaissance a priori. L'approche proposée basée sur la fusion des deux concepts flou et possibiliste et initialisé par la matrice d'appartenance, permet, d'une part, de résoudre simultanément le problème de chevauchement et de la coïncidence, de réduire l'effet du bruit et d'autre part d'accélérer le processus de la classification. Pour valider notre modèle, nous avons effectué des tests avec les FCM (Fuzzy C-Means), les PCM (Possibilistic C-Means) et les FPCM (Fuzzy-Possibilistic C-Means) pour deux cas d'initialisation sur les bases de données : iris, cuisse humaine et image des textures.

Mots clés : Reconnaissance des formes, logique floue, clustering flou, apprentissage non supervisé.

Abstract

This thesis is placed within the framework of the fuzzy clustering, which aims to seek a partition of a set of forms incompletely defined in the most natural possible classes or clusters without any a priori knowledge. The proposed approach, based on the fusion of fuzzy and possibility concepts and initialized by a membership matrix, allows on the one hand to solve simultaneously the problem of overlapping and coincidence, to reduce the noise effect and on the other hand to accelerate the clustering process. The model validation is carried out by the FCM (Fuzzy C-Means), the PCM (Possibilistic C-Means) and the FPCM (Fuzzy-Possibilistic C-Means) for two cases of initialization by using Iris, Textured image and Tight human data basis.

Keywords: Pattern recognition, fuzzy logic, fuzzy clustering, unsupervised learning.

الملخص

هذه الأطروحة تدخل تحت إطار التقسيم غير المراقب الغامض ، الذي يهدف إلى تقسيم مجموعة من الأشكال الغامضة إلى عدت مجموعات طبيعية دون أي معرفة مسبقة. الطريقة المقترحة التي تعتمد على الدمج بين نضرتي الإمكان والغموض ، قادرة من جهة وفي نفس الوقت على حل مشكلتي التشابك و التصادف ،التقليل من اثر التشويش ومن جهة أخرى على تسريع عملية التقسيم. للتحقق من صحة نموذجنا ،أجرينا اختبارات على قواعد البيانات : Iris، Cuisse humaine وImage des textures مع الخوارزميات FCM، PCM و FPCM.

الكلمات المفتاحية :المنطق الغامض ، إعادة معرفة الأشكال ، التقسيم غير المراقب

Remerciements

C'est avec un grand plaisir que je réserve ces mots en signe de reconnaissance à ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Mes remerciements s'adressent tout d'abord à Monsieur le président ***Mr. Abdelhani BOUKROUCHE*** *Professeur à l'université de Guelma.*

Je remercie aussi ***Mr. Abdelkrim MOUSSAOUI*** *Maître de conférences à l'université de GUELMA,* ***MM. Khireddine-Mohamed KHOLLADI*** *et* ***Allaoua CHAOUI****, Maîtres de conférences à l'université de CONSTANTINE et* ***Mr. Okba KAZAR*** *Maître de conférences à l'université de BISKRA, membres du jury, pour l'honneur qu'ils m'ont accordé en acceptant de juger mon travail.*

Je tiens à exprimer toute ma reconnaissance et ma gratitude ***à Mr. SERIDI Hamid****, Professeur à l'université de Guelma, pour avoir accepté de diriger ce travail dès le début, et pour sa contribution décisive dans l'élaboration de ce travail.*

Mes remerciements s'étendent également à ***Mr. Herman AKDAG*** *Professeur à l'université de REIMS, ainsi que à tous les membres du CRestic.*

Merci à tous

Table des matières

Introduction générale

I. Cadre général

Le champ d'étude relative à la reconnaissance de formes, qui est un sujet largement interdisciplinaire, s'est développé de manière significative depuis les années soixante. Il couvre des développements dans le domaine des statistiques, d'ingénierie, d'intelligence artificielle, d'informatique, de psychologie, de physiologie, etc. Le grand nombre d'applications, partant des applications classiques telles que la reconnaissance automatique des caractères et le diagnostic médical arrivant aux récentes applications du *data mining* (tels que le crédit scoring et l'analyse des transactions par carte de crédit), ont attiré de considérables efforts de recherche donnant naissance, ainsi, à de nombreuses méthodes.

Jain et al. dans [78] ont donné la suivante définition à la reconnaissance des formes : "*A form recognition is the act of taking in raw data and taking an action based on the category of the pattern*". À ce stade, la plupart des recherches portent sur les méthodes d'apprentissage supervisé ou non supervisé [78] et [79].

Parmi les méthodes d'apprentissage non supervisées nous distinguons la méthode d'*analyse de clusters* ou *clustering* cherchant à regrouper un ensemble d'observations en sous-ensembles appelés *clusters*. Elle a fait le sujet de plusieurs ouvrages tels que [112], [51], [14], [83], [15], [52], [126]. Par ailleurs, des milliers d'articles touchant à ce domaine ont été publiés en mettant en jeu soit de nouvelles techniques, soit des versions améliorées de méthodes déjà existantes par exemple [35], [127], [4], [8], [92], [101], [37], [93], [19], [65], [69], [5], [125], [87], [118], [21], [45], [56], [28], [75], [124], [30], [117].

Grossièrement, l'objectif du clustering consiste à diviser un ensemble d'objets en un ensemble de classes ou *clusters*, de telle sorte que les objets affectés au même cluster doivent être assez proches que possible, et que deux objets de différents clusters devraient être assez dissemblables que possible. Ainsi, nous pouvons dire que cette technique tente de modéliser la capacité de l'homme pour grouper des objets similaires dans des classes ou catégories homogènes.

Plusieurs motivations sont à l'origine de la construction de classes suivant cette technique [20]. Principalement, le clustering est un outil pour la découverte de la structure précédemment cachée dans un jeu de données désordonnées. Dans ce

cas, on suppose que le « vrai » ou le naturel groupement existe dans l'ensemble de données. Toutefois, l'affectation d'objets aux classes et la description de ces classes ne sont pas préalablement connues. En assemblant les objets similaires en clusters, on essaie de reconstituer le groupement inconnu dans l'attente que chaque découvert cluster représente un véritable type ou catégorie d'objets.

Pour mener à bien leur tâche, les méthodes de clustering doivent s'attaquer à diverses difficultés. L'une des problématiques centrales du clustering est celle de définir la notion de similarité entre les données qui peuvent être de nature numérique ou catégorielle. Il existe trois sortes de similarité en clustering : la similarité à maximiser pour deux objets appartenant au même cluster et à minimiser pour deux objets appartenant à des clusters différents, la similarité entre un objet et un cluster à maximiser si l'objet est associé au cluster pour une bonne *cohésion interne* du cluster; et la similarité entre clusters à minimiser pour une bonne *isolation externe* des clusters. Typiquement, la similarité entre objets est estimée par une fonction calculant la distance entre ces objets. Malheureusement, la plupart des méthodes de clustering ne prennent pas en compte la notion de contexte, d'autant plus qu'elles se basent en général sur une notion de distance entre objets. D'autre part, les méthodes de clustering doivent porter une attention particulière à leur complexité pour qu'elles soient applicables à de larges bases de données. Un dernier problème est celui d'évaluation des sorties des méthodes de clustering. C'est un problème très particulier du fait que l'intérêt même d'un regroupement est difficile à définir, car il est subjectif par nature [27].

Vu le nombre important de techniques de clustering rencontrées dans la littérature, il serait très difficile de fournir une liste exhaustive englobant toutes ces dernières. De même, et vu le nombre de propriétés de chaque technique, il est impossible d'en réaliser un classement complet. Cependant, si on ne considère que des attributs à valeurs numériques, des méthodes de type polythétique et des problèmes sur des bases de données de taille raisonnable, la diversité des méthodes devient assez bien délimitée et peuvent être généralement divisées en méthodes hiérarchiques [78] et méthodes de partition [60], [68], [115].

Les méthodes hiérarchiques produisent une hiérarchie complète qui est une séquence imbriquée de partitions de données d'entrée. Elles peuvent être soit d'agglomérations (ascendantes) ou de division (descendantes). Les méthodes ascendantes génèrent une séquence de partitions imbriquées en partant d'un regroupement trivial dans lequel chaque élément se trouve dans un cluster unique et en terminant par le regroupement trivial où tous les éléments sont dans le même cluster. Une méthode de division, comme son nom l'indique, effectue une procédure de division partant d'un cluster regroupant tous les objets jusqu'à ce qu'un critère d'arrêt soit atteint (généralement jusqu'à l'obtention d'une partition de clusters représentés par des singletons).

Généralement, les méthodes de partition visent à obtenir une seule partition de données d'entrée en un nombre déterminé de clusters. Ces méthodes cherchent,

souvent, une partition qui optimise une fonction objectif adéquate. Elles peuvent être divisées en clustering dur et clustering flou.

Le clustering dur fournit une partition dure où chaque forme de l'ensemble de données est affectée à un et un seul cluster. Dans ces méthodes, on trouve la famille des méthodes de clustering par critère du carré des écarts aux centroïdes (*centres mobiles*, *nuées dynamiques*, *k-moyennes*), le clustering statistique, le clustering stochastique, le clustering basé sur la densité, le clustering basé sur les grilles, Le clustering basé sur les graphes, et finalement, le clustering spectral.

Le clustering flou [14], [17] quand à lui, génère une partition floue fournissant le degré d'appartenance de chaque forme à un cluster donné. Dans des situations réelles, le clustering flou est plus naturel que le clustering dur. Ça se voit clairement dans le cas des objets situés aux frontières de deux ou plusieurs clusters, où le clustering flou ne force pas ces objets d'appartenir complètement à l'un des clusters, mais plutôt il les assigne avec un degré d'appartenance compris entre 0 et 1 indiquant leurs appartenances partielles.

Dans ce travail, nous nous intéressons essentiellement aux techniques de clustering flou destinées à la reconnaissance des formes incomplètement définies. Plusieurs théories ont été développées afin d'offrir un cadre formel au traitement d'informations incomplètement définies ; citons, par exemple, les théories des probabilités [112], des sous-ensembles flous [49], [134], de Dempster-Shafer [114], ou des possibilités [131], [50]. Notons que, les notions de classe et de sous-ensemble flou sont, intuitivement liées dans les mécanismes du raisonnement humain.

L'application en clustering de la théorie des sous-ensembles flous proposée par Zadeh [130] a donné lieu à une nouvelle technique de clustering flou en se basant sur des relations floues et des fonctions objectives. Cela suggère pour certains chercheurs une nouvelle approche de regroupement tirant profit du concept de fonction d'appartenance [111], [54], [16]. Toute fois, le premier chercheur qui a proposé l'utilisation des ensembles flous dans le clustering est Ruspini (1969).

Une grande variété de méthodes de clustering flou a été proposée à travers une plus grande quantité d'articles et d'ouvrages. Elles visent à trouver tous les clusters flous qui peuvent exister dans un certain ensemble d'exemples. Pour pouvoir présenter de telles méthodes, il est fructueux de se concentrer sur leurs idées sous-jacentes et leurs principes suivant lesquelles elles (méthodes) ont été catégorisées. Au début, nous regardons de plus près les méthodes qui tentent de trouver une bonne partition floue et des prototypes de clusters en utilisant un critère global pour l'optimisation sous la forme d'une *fonction objectif*. La tâche de regroupement peut alors être formulée comme un problème d'optimisation d'une fonction. La *fonction objectif* dépend à la fois des prototypes de clusters et des appartenances des objets aux clusters. Elle ne peut pas être directement optimisée et donc un modèle AO (Alternating Optimization) est généralement appliqué pour optimiser un groupe de paramètres (par exemple, les degrés d'appartenance) en

maintenant l'autre groupe (par exemple, les prototypes) fixé et vice versa. Ce schéma de mise à jour itératif est répété dans l'attente d'approcher d'optimum global de la fonction critère. La grande variété des approches de clustering floues est due à des modifications de fonctions objectives. Ces modifications sont destinées à l'amélioration des résultats en fonction des problèmes particuliers (par exemple, le bruit, les valeurs aberrantes). Les variantes des algorithmes possibilistes et probabilistes sont des exemples des approches de clustering flou.

Les degrés d'appartenance obtenus avec l'approche probabiliste et avec ses variantes étant des quantités relatives. Ils ne peuvent pas traduire un autre point de vue qui consisterait à associer un objet à une classe si cet objet s'avère être un élément typique de cette classe. Cette notion de typicalité offre une nouvelle interprétation du regroupement, et la théorie des possibilités fournit un cadre adéquat à une telle interprétation. À partir de ce point de vue, Krishnapuram et Keller ont proposé une approche possibiliste [90] faisant appel à des partitions assouplies. Leur approche est censée conduire à une meilleure performance en présence de bruit. Mais leur travail est, essentiellement, motivé par le désir de remédier au caractère relatif des degrés d'appartenance générés par les approches probabilistes. Cette approche, théoriquement attirante, échoue malheureusement à produire des résultats conformes aux motivations qui sont à l'origine de son apparition.

Ainsi, notre contribution dans le domaine de clustering s'intéresse à la description d'un nouveau critère d'optimisation qui prend en compte toutes ces limitations. La combinaison des deux concepts d'appartenance flous à savoir les appartenances probabiliste et les appartenances possibiliste s'avère fructueuse de point de vu bénédiction des avantages et contournement aux inconvénients de chaque approche.

Un autre problème restant, aujourd'hui, ouvert dans le domaine du clustering est celui d'évaluation des résultats qui se traduit par une évaluation de la pertinence des clusters formés. La difficulté vient principalement du fait que l'évaluation des résultats des algorithmes est subjective par nature, car il existe, souvent, différents regroupements pertinents possibles pour un même jeu de données [26]. En pratique, il existe trois méthodes principales pour mesurer la qualité des résultats des algorithmes de clustering. Le premier type utilise des données artificielles pour lesquelles le regroupement attendu est connu, puis il compare les résultats obtenus aux résultats attendus. On parle dans ce cas *d'évaluation externe.* Une deuxième approche se base sur le calcul des propriétés résultant de cluster, comme la compacité et la séparation. Cette approche est appelée la *validation interne*. La troisième méthode, appelée *validation relative*, se base sur la comparaison des partitions générées par le même algorithme avec des paramètres différents.

II. Organisation du mémoire

Chapitre I : Ce chapitre est consacré à la présentation d'une vue générale du problème de la reconnaissance des formes à travers laquelle nous verrons que les formes peuvent subir plusieurs étapes de transformation distinctes. Ces transformations (appelées parfois : le prétraitement, la sélection ou l'extraction de caractéristiques) une fois opérées sur des données permettent, généralement, de réduire leur dimension en leurs transformant en formes plus approprié pour la classification ultérieure. Les principes de la théorie des sous-ensembles flous ainsi que la façon dont ils peuvent être utilisés dans un système de reconnaissance seront ensuite abordés. On va dresser, également, un état de l'art portant sur le problème de clustering en citant quelques exemples d'applications suivis d'une description des différents problèmes rencontrés dans le domaine. Cependant, il est à noter que la notion d'apprentissage non supervisée, classification non supervisée ou clustering ont le même sens.

Chapitre II : À ce niveau, les différentes méthodes de clustering traditionnelles sont abordées en établissant un classement particulier sur la base d'hypothèses critiquées. Les méthodes présentées seront discutées objectivement, tant au niveau de l'algorithme que des clusters qu'elles fournissent. Pour chacune de ces méthodes, nous donnons un aperçu général de ses caractéristiques principales, de ses forces et de ses faiblesses. Ainsi, selon l'application envisagée pour le clustering, l'utilisateur pourra rechercher la méthode la plus appropriée en fonction de l'ensemble de ses caractéristiques.

Chapitre III : Pour la reconnaissance des formes incomplètement définies, une introduction aux fondements du domaine étendu au clustering flou sera présentée à travers ce chapitre. Les partitions floues des clusters seront introduites avec un accent particulier sur l'interprétation des deux types les plus rencontrés de l'assignement graduel: les degrés d'appartenance floue et les degrés d'appartenance possibiliste. Une vue systématique des méthodes de clustering flou sera fournie, en soulignant sur les idées sous-jacentes des différentes approches. La classe des méthodes basées sur l'optimisation en alternance de la fonction objectif (Alternating Optimization : AO), la famille des algorithmes d'estimation de clusters en alternance « Alternating Cluster Estimation : ACE», le schéma d'estimation du maximum de vraisemblance floue (Fuzzy Maximum Likelihood Estimation : FMLE) avec son homologue de maximisation d'expectation « Expectation Maximization : EM» sera, ensuite, discutée.

Chapitre VI : Ce quatrième chapitre est consacré à la présentation de nos contributions dans le domaine de clustering précédée d'une étude comparative entre les algorithmes de clustering des c-moyennes floues. Nous verrons essentiellement les avantages et les inconvénients des algorithmes connus sous le nom de : FCM et PCM qui sont des algorithmes largement utilisés dans le domaine de clustering. L'approche proposée FPCM faisant appel à des partitions normalisées et absolues.

Chapitre V : Dans ce cinquième chapitre**,** nous présentons un certain nombre de mesures de validité proposées dans la littérature. Pour vérifier la validité de notre modèle de clustering, des testes ont été effectués sur trois bases de données (Iris, cuisse humaine et image des textures) avec trois algorithmes de clustering des c-moyennes floues : FCM, PCM et FPCM.

Finalement, des conclusions et des perspectives envisagées par nos travaux sont présentées.

Chapitre I
Reconnaissance des formes et vue générale du problème de clustering

I.1 Introduction

Le but principal de la reconnaissance des formes (RdF) est la classification supervisée ou non supervisée. Parmi les différents cadres dans lesquels la reconnaissance des formes a été habituellement formulée, l'approche statistique a été étudiée et utilisée plus intensément dans la pratique. La conception d'un système de reconnaissance nécessite une attention particulière aux problèmes suivants : définition des formes des classes, la perception de l'environnement, représentation de la forme, l'extraction et la sélection de caractéristiques, l'analyse de clusters, la conception du classifieur et apprentissage, la sélection de l'ensemble d'apprentissage et de test et l'évaluation des performances. Des nouvelles applications ; telles que data mining, recherche sur le web, recherche d'informations multimédia, reconnaissance de visage et reconnaissance d'écriture cursive, exigent des techniques robustes et efficaces de reconnaissance des formes.

Un système complet pour la RdF se compose : d'un capteur qui recueille les observations qui doivent être classées ; d'un mécanisme d'extraction des caractéristiques qui calcule l'information numérique ou symbolique à partir des observations ; d'un classifieur qui fait la tâche de classification, en s'appuyant sur les caractéristiques extraites et finalement d'un système d'évaluation du résultat de la classification.

La classification est généralement basée sur la disponibilité d'un ensemble de formes qui ont déjà été classées ou décrites. Cet ensemble de formes est appelé ensemble d'apprentissage, et la stratégie d'apprentissage résultante est dite « apprentissage supervisé ». L'apprentissage peut également être non supervisés, dans le sens que l'étiquetage des formes n'est pas donnée a priori. Dans ce cas, les classes sont établies en se basant sur les régularités statistiques des formes. Dans les statistiques, l'apprentissage non supervisé est étroitement lié au problème de l'estimation de densité. Cependant, l'apprentissage non supervisé est aussi englobe plusieurs d'autres techniques qui cherchent à résumer et expliquer les données, le clustering est un exemple de ces techniques.

Le clustering a pour objectif de déterminer une structuration des données manipulées. Pour cela, des regroupements (ou classes) sont recherchés à partir d'une base d'exemples non étiquetés. Chaque regroupement est caractérisé par un sous-ensemble d'exemples ayant des propriétés communes et présentant donc une

certaine similarité entre eux. Les termes classification non supervisée, clustering ou apprentissage non supervisé seront considérés équivalents.

Dans la deuxième section, nous décrivons les différentes étapes de la RDF, en citant quelques approches d'extraction et de sélection des caractéristiques pour la classification. La notion d'apprentissage supervisée et non supervisée fera l'objet de la troisième section. La quatrième section modélise un problème de reconnaissance des formes dans un environnement flou. Le concept de logique floue est inclus pour mieux considérer l'imprécision. La cinquième section présente une vue générale de clustering : les différentes problématiques rencontrées ; la notion de similarité et les difficultés engendrées par la notion de distance ; le choix de la méthode en fonction de l'application et des critères associées et les approches existantes permettant d'évaluer les résultats d'un algorithme de clustering.

I.2 Processus de la reconnaissance de formes

Le terme « forme » désigne le vecteur des mesures à *p-dimension* de donnée $x=(x_1,\ldots, x_p)$, dont les composantes x_i sont des mesures de caractéristiques (attributs) d'un objet. Des exemples des formes sont : des mesures d'une onde acoustique dans le problème de la reconnaissance vocale; des mesures prises sur un patient en vue d'identifier une maladie (diagnostic); mesures sur les variables météorologiques (pour la prévision ou la prédiction) et une image numérisée pour reconnaissance de caractères. Nous considérons le terme 'forme', dans son sens technique et qui ne réfère pas nécessairement à la structure des images. La forme est un couple comprenant une observation et un sens. La reconnaissance des formes est la déduction du sens de l'observation. La conception d'un système de reconnaissance de formes est l'établissement d'une cartographie à partir de l'espace de mesure vers l'espace de significations potentielles, en vertu duquel les différentes significations sont représentées dans cet espace comme des points figuratifs discrets [132]. Les composants de base dans un système de reconnaissance des formes sont le prétraitement, l'extraction, la sélection des caractéristiques et la conception du classifieur et l'optimisation.

I.2.1 Prétraitement

Le prétraitement consiste à segmenter la forme intéressante à partir de l'arrière-plan. Généralement, filtrage du bruit, lissage et normalisation devraient être faits dans cette étape. Le prétraitement définit également une représentation compacte de la forme. Dans l'exemple de la reconnaissance vocale, une étape de prétraitement peut être la transformation de la forme d'onde à une représentation de fréquence.

I.2.2 Sélection et extraction des caractéristiques

Les caractéristiques devraient être facilement calculées, robustes, insensibles aux diverses distorsions et variations dans les images, et invariants par rotation. La

sélection des attributs consiste à choisir le meilleur sous-ensemble de l'espace d'entrée. Son but ultime est de sélectionner les sous-ensembles d'attributs optimaux qui permettent d'atteindre le résultat avec une grande précision.

Dans l'extraction des caractéristiques, la plupart des méthodes sont supervisées. Ces approches nécessitent certaines connaissances a priori et des échantillons d'apprentissage étiquetés. Il existe deux types de méthodes supervisées utilisées: extraction des attributs linéaires et l'extraction des attributs non linéaires. Techniques d'extraction d'attributs linéaire incluent Analyse en Composante Principale (ACP), l'Analyse Discriminante linéaire (ADL) et Analyse en Composante Indépendante (ACI). Les méthodes d'extraction d'attributs non linéaire comprennent: noyau de l'ACP, réseau ACP, réseau auto-associatif non linéaire, Echèle Multidimensionnel (MDS) et cartes auto-organisatrices (SOM), …

I.2.3 Conception d'un classifieur

Après la sélection d'un sous-ensemble de caractéristiques optimales, un classifieur peut être conçu en utilisant différentes approches. Approximativement, il existe trois approches différentes [98]. La première approche est la plus simple et la plus intuitive ; approche qui est basée sur le concept de similarité. La seconde est une approche probabiliste. Elle comprend les méthodes basées sur la règle de décision de Bayes, le maximum de vraisemblance ou estimateur de densité. Trois célèbres méthodes sont : K-plus proche voisin (KPP), classifieur de fenêtre de Parzen et méthodes de branchement lié (branch-and bound : BnB). La troisième approche consiste à construire les limites de décision directement par l'optimisation de certains critères d'erreur : perceptrons multicouches, arbre de décision et SVM.

I.2.4 Optimisation

L'optimisation n'est pas une étape séparée, elle est combinée avec plusieurs parties de la structure de processus de la reconnaissance. Dans le prétraitement, l'optimisation garantit que la forme d'entrée a la meilleure qualité. Ensuite, dans la partie de sélection et d'extraction des caractéristiques, les sous-ensembles des caractéristiques optimales sont obtenus sous certaines techniques d'optimisation. Par ailleurs, le taux d'erreur de la classification finale est diminué dans la partie de la classification.

I.3 Apprentissage automatique à partir de données

L'objectif général de *l'apprentissage automatique* est d'extraire automatiquement l'ensemble des connaissances nécessaires à la modélisation du problème en se basant sur des exemples, c'est-à-dire sur un ensemble limité de données disponibles. Les données $\{x_j, j = 1, \ldots, N\}$ sont elles-mêmes, suivant les auteurs et les contextes, appelées observations, échantillons, individus ou encore

exemples. Quand elles ont été expertisées au préalable pour leur attribuer l'étiquette d'une des classes du problème, ce problème se décline en plusieurs variantes, en fonction des informations disponibles sur le problème traité [106]:

- L'apprentissage supervisé ;
- L'apprentissage non supervisé ;
- L'apprentissage semi-supervisé ;

I.3.1 L'apprentissage supervisé

L'objectif général de la *classification* est d'être capable d'étiqueter des données en leur associant une classe. Si les classes possibles sont connues et si les exemples sont fournis avec l'étiquette de leur classe, on parle *d'apprentissage supervisé* ou *d'analyse discriminante.* Dans ce cas, il s'agit alors d'utiliser les exemples fournis déjà classés pour apprendre un modèle qui permet ensuite d'associer à tout nouvel exemple rencontré sa classe la plus adaptée. Un exemple d'application de l'apprentissage supervisé concerne la médecine : étant donnés les résultats d'analyse d'un patient, et la connaissance de l'état d'autres patients pour lesquels les mêmes analyses ont été menées, il est possible d'évaluer le risque de maladie de ce nouveau patient en fonction de la similarité de ses analyses avec celles des autres patients.

I.3.2 L'apprentissage non supervisé

Au contraire, si seuls des exemples sans étiquette sont disponibles, et si les classes et leur nombre sont inconnus, on parle *d'apprentissage non supervisé,* ou *clustering.* Dans ce cas, l'apprentissage se ramène alors à cibler les groupes homogènes d'exemples existant dans les données, c'est-à-dire à identifier des groupes tels que les exemples les plus similaires appartiennent au même groupe, et que les exemples les plus différents soient séparés dans différents groupes, la notion de similarité étant le plus souvent ramenée à une fonction de distance entre paires d'exemples. Autrement dit, il s'agit à ce niveau de rechercher la distribution sous-jacente des exemples dans leur espace de description. En médecine, il peut par exemple être intéressant de détecter, parmi un ensemble de patients atteints d'une maladie donnée, les différents groupes de malades reflétant différentes causes possibles de la maladie.

Dans le cas d'un *apprentissage non supervisé*, la sortie n'est pas disponible a *priori* et l'apprentissage consiste à identifier une structure dans les données en regroupant celles qui possèdent des propriétés similaires. Ce type de modélisation s'effectue par des algorithmes *de classification non supervisé ou clustering*.

Dans la suite de ce chapitre, nous aborderons les méthodes reposant sur le clustering qui sont des outils essentiels pour extraire automatiquement une structure à partir des données et constituent une base pour notre approche. Nous verrons quelques méthodes de clustering plus particulièrement dans le chapitre II. Nous ne nous intéressons pas aux approches de classification supervisée, Il s'agit d'un domaine de recherche à part entière.

I.4 Représentation des connaissances incomplètement définies

Lorsque l'on souhaite modéliser un problème qui est par nature soumis aux imperfections (imprécision et incertitude), il est important de prendre en compte ces imperfections pour que l'exploitation du modèle soit la plus robuste possible. Dans ce cadre, les probabilités constituent un premier outil de représentation permettant de gérer certaines formes d'incertitudes. Elles peuvent être utilisées pour traduire le caractère aléatoire des phénomènes observés ou encore pour représenter l'incertitude liée à des connaissances a *priori* [106]. Mais les approches probabilistes sont généralement contraintes par les axiomes et les hypothèses sur lesquels repose la théorie et limitées par la difficulté d'obtenir de « bonnes » estimations. De plus, les probabilités ne permettent pas de traiter les imperfections liées aux imprécisions, phénomène particulièrement présent dès lors que des données du monde réel sont manipulées.

Pour faire face à ces limitations, Zadeh a introduit la *Théorie des sous- ensembles flous* [130] et la *Théorie des possibilités* [131]. Alors que la première vise à gérer les imprécisions liées aux observations, la seconde permet en plus de prendre en compte les incertitudes et notamment celles qui ne sont pas de nature probabiliste. Il existe aussi d'autres théories permettant de raisonner avec les deux types d'imperfections, telles que la *Théorie de l'évidence* de Shafer [114].

Dans notre étude, on considère le cas général des données qui présente des imperfections liées aux imprécisions et aux incertitudes. Dans la partie suivante, nous allons présenter que les principes de la théorie des sous-ensembles flous ainsi que la façon dont ils peuvent être utilisés dans un système de reconnaissance.

I.4.1 Théorie des sous-ensembles flous

Le concept de sous-ensemble flou a été introduit par Zadeh en 1965 [130]. Il permet d'éviter le passage brusque d'une classe à une autre et autorise également une appartenance intermédiaire, ou encore appartenir partiellement à chacune. Ce concept a été originellement motivé par des problèmes de classification et caractérisé particulièrement par l'abandon du tiers-exclu (soit une proposition est vraie, soit elle est fausse) et celle de la contradiction (une proposition ne peut-être à la fois vraie et fausse). Nous ne donnons ici que les bases nécessaires à notre étude.

– *Définition d'un sous-ensemble flou :* Soit un référentiel *X* dénombrable ou non et x un élément de X, alors un ensemble flou A de *X* est un ensemble de couples tel que :

$$A = \{(x, F_A(x))\}, x \in X$$

$F_A(x)$ est le degré d'appartenance de x à A. $F_A(x)$ est attribué à X par la fonction d'appartenance de A qui prend ses valeurs dans l'intervalle [0,1]. Si cet intervalle se réduit aux valeurs {0,1}, la fonction d'appartenance, prend alors les valeurs binaires 0 ou 1 et l'ensemble A est un ensemble ordinaire.

– *Notion d'appartenance partielle :* Dans la théorie des ensembles, un élément appartient ou n'appartient pas à un ensemble. La notion d'ensemble est à l'origine de nombreuses théories mathématiques. Cette notion essentielle ne permet cependant pas de rendre compte de situations pourtant simples et rencontrées fréquemment. Parmi des fruits, il est facile de définir l'ensemble des pommes. Par contre, il sera plus difficile de définir l'ensemble des pommes mûres. On conçoit bien que la pomme mûrit progressivement..., la notion de pomme mûre est donc graduelle. C'est pour prendre en compte de telles situations qu'a été crée la notion d'ensemble flou. La théorie des ensembles flous repose sur la notion d'appartenance partielle : chaque élément appartient partiellement ou graduellement aux ensembles flous qui ont été définis. Les contours de chaque ensemble flou (figure I.4) ne sont pas « nets », mais « Flous » ou « graduels ».

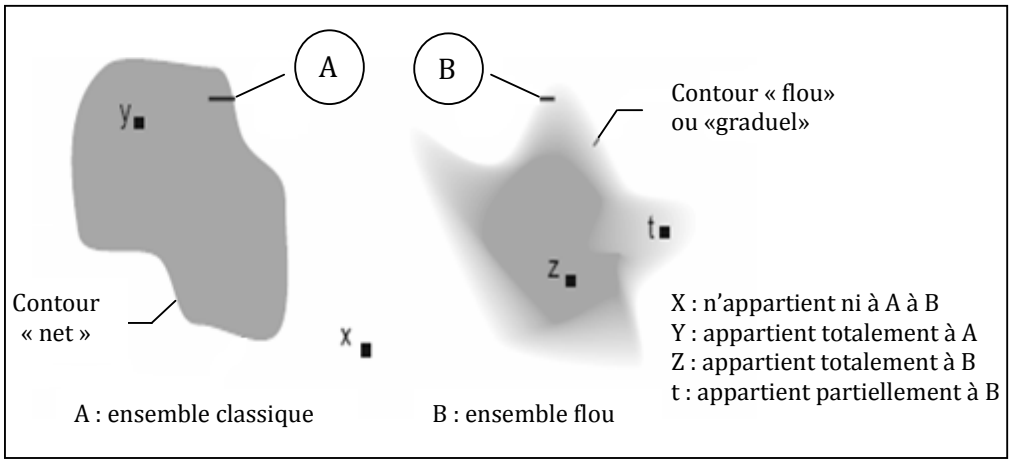


Fig. I.1 : Comparaison d'un ensemble classique et d'un ensemble flou.

I.4.2 Fonctions d'appartenance

Un ensemble flou est défini par sa « fonction d'appartenance », qui correspond à la notion de « fonction caractéristique » en logique classique. Supposons que nous voulions définir l'ensemble des personnes de «taille moyenne ». En logique classique, nous conviendrons par exemple que les personnes de taille moyenne sont celles dont la taille est comprise entre 1,60 m et 1,80 m. La fonction caractéristique de l'ensemble (Figure I.2) donne « 0 » pour les tailles hors de l'intervalle [1,60 m ; 1,80 m] et « 1 » dans cet intervalle. L'ensemble flou des personnes de « taille moyenne » sera défini par une « fonction d'appartenance » qui diffère d'une fonction caractéristique par le fait qu'elle peut prendre n'importe quelle valeur dans l'intervalle [0, 1]. A chaque taille possible correspondra un « degré d'appartenance » à l'ensemble flou des « tailles moyennes » (Figure I.3), compris entre 0 et 1.

L'exemple de la (Figure I.4) montre la gradualité que permet d'introduire la logique floue. Une personne de 1,80m appartient à l'ensemble « taille grande »

avec un degré 0,3 et à l'ensemble « taille moyenne » avec un degré de 0,7. En logique classique, le passage de moyen à grand serait brusque. Une personne de 1,80m serait par exemple de taille moyenne alors qu'une personne de 1,81 m serait grande, ce qui choque l'intuition.

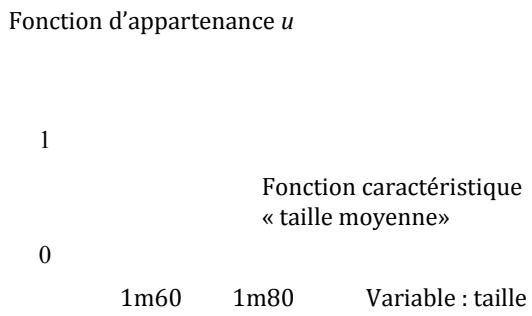


Fig. I.2: Fonction caractéristique**.**

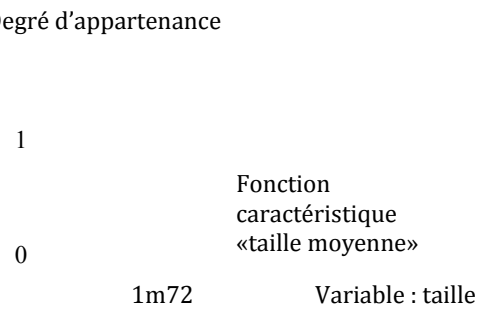


Fig. I.3 : Fonction d'appartenance.

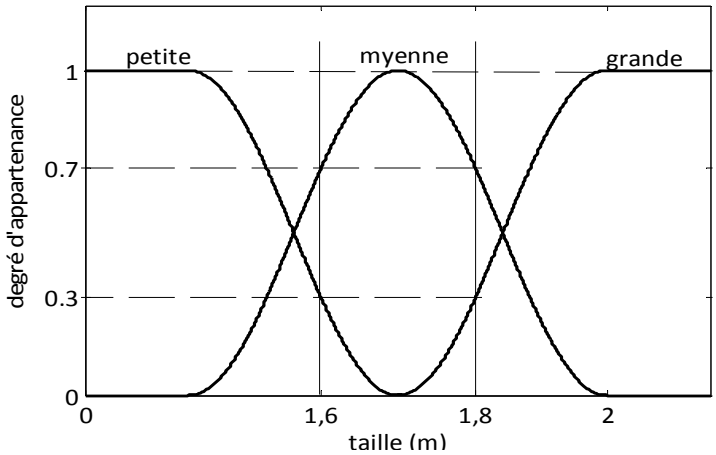


Fig. I.4 : Fonction d'appartenance, variable et terme linguistique.

La variable (par exemple : taille) ainsi que les termes (par exemple : moyenne, grande) définis par les fonctions d'appartenance portent respectivement les noms de variable linguistique et de termes linguistiques.

I.4.3 Formes des fonctions d'appartenance

Les fonctions d'appartenance peuvent théoriquement prendre n'importe quelle forme. Toutefois, elles sont souvent définies par des segments de droites, et dites « linéaires par Morceaux » (Figure I.5).

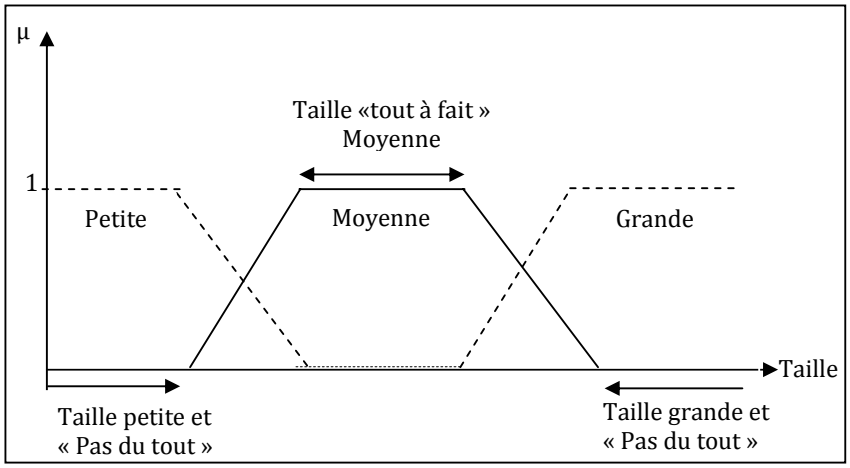


Fig. I.5 : Fonctions d'appartenance linéaires par morceaux.

I.4.4 Fuzzification - Degré d'appartenance

L'opération de fuzzification permet de passer du domaine réel au domaine du flou. Elle consiste à déterminer le degré d'appartenance d'une valeur (mesurée par exemple) à un ensemble flou. Par exemple (Figure I.6), si la valeur courante de la variable « entrée » est de 2, le degré d'appartenance à la fonction d'appartenance « entrée faible » est égal à 0,4 qui est le résultat de la fuzzification.

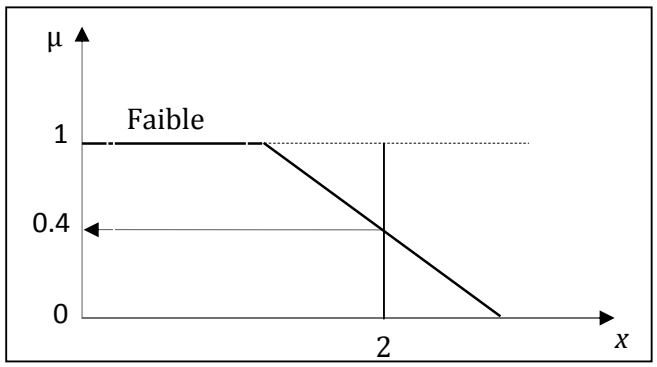


Fig. I.6 : Fuzzification

On peut aussi dire que la proposition « entrée faible » est vraie à 0,4. On parle alors de degré de vérité de la proposition. Degré d'appartenance et degré de vérité sont donc des notions similaires.

I.4.5 Caractéristiques d'un sous-ensemble flou

Dans ce paragraphe nous présentons les caractéristiques d'un sous-ensemble flou qui sont principalement liées à la forme de sa fonction d'appartenance et qui le démarquent des sous-ensembles classiques.

Le support: est la partie de X sur la quelle le degré d'appartenance de A n'est pas nul.

$$\sup(A) = \{x \in X / F_A(x) \succ 0\}$$

La hauteur: est la plus grande valeur prise par la fonction d'appartenance associé à :

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

Le noyau: est l'ensemble des éléments de X pour lesquels la fonction d'appartenance de A vaut 1.

$$noy(A) = \{x \in X / F_A(x) = 1\}$$

Les α -coupes: est l'ensemble des éléments de X pour lesquels les degrés d'appartenances à A sont au moins égaux à α.

$$A_\alpha = \{x \in X / F_A(x) \geq \alpha\}.$$

La figure I.7 illustre l'ensemble des caractéristiques sur un exemple simple:

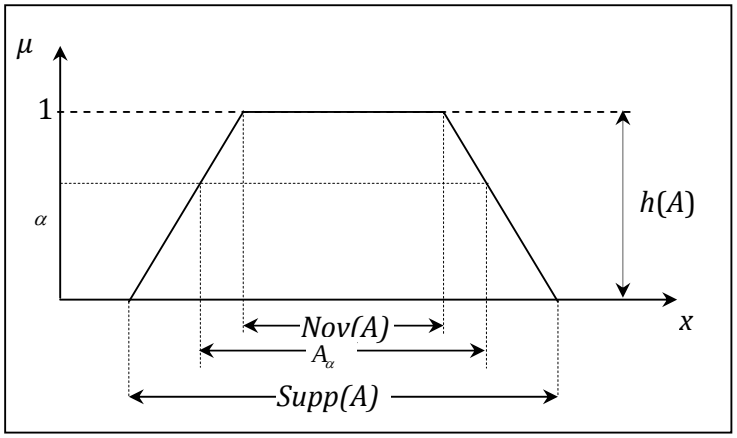


Fig. I.7: Caractéristiques les plus utiles qui représentent un sous-ensemble flou.

Dans les paragraphes suivants nous présentons une vue générale du problème du clustering, en introduisant quelques aspects liés à la classification non supervisée avant d'introduire ses extensions «floues» et notamment l'algorithme des *C-moyennes floues* [14]. Nous y reviendrons plus en détails dans le chapitre III. C'est en effet ce dernier qui constitue une base pour notre approche et qui va nous permettre de modéliser automatiquement les connaissances incomplètement définies.

I.5 Vue générale du problème de clustering

L'objectif, très général du clustering, consiste à diviser un ensemble d'objets dans un ensemble de classes ou *clusters* fondés sur la similitude, de telle manière que les objets affectés au même cluster doivent être aussi proches que possible, et que deux objets de différents clusters devraient aussi dissemblables que possible. On distingue essentiellement deux grands types d'approches pour la classification non supervisée: *les approches adaptatives* et *les approches hiérarchiques.* Les

premières visent à obtenir une seule partition de données d'entrée en un nombre déterminé de clusters. Ces méthodes cherchent souvent une partition qui optimise une fonction objective adéquate. Les secondes utilisent les données initiales soit pour les regrouper au fur et à mesure *(approches agglomératives),* soit au contraire pour effectuer des divisions successives *(approches divisives)*. Quelle que soit l'approche envisagée, toutes reposent sur un même fondement: l'utilisation d'une *mesure de similarité* qui permet de déterminer si deux individus de la base et par extension si deux sous-ensembles d'individus se ressemblent. Plusieurs études ont été faites notamment pour pouvoir comparer différents types d'individus. Il existe ainsi des mesures de similarité/dissimilarité entre des chaînes de caractères [122], entre des sous-ensembles flous ou des prototypes flous [107], entre des mots (au sens acoustique du terme), des images [1], etc. Dans les espaces de représentation numériques, les mesures de dis similarité les plus courantes reposent sur l'utilisation d'une distance (cf. annexe A). Le choix de celle-ci est un élément très important pour le bon fonctionnement des algorithmes de classification non supervisée. Il influe très fortement sur la forme des regroupements trouvés ainsi que sur les propriétés de classification qui en découlent. Ainsi, si la distance Euclidienne fonctionne bien dans des espaces où les données sont réparties de façon homogène dans toutes les dimensions, elle donnera des résultats moins satisfaisants si les données sont étalées selon une direction privilégiée dans l'espace.

I.5.1 Applications du clustering

Appliquer une méthode de clustering revient à mettre de l'ordre dans un jeu de données. À partir de ceci, on conçoit aisément que les domaines d'application du clustering peuvent être très variés. C'est pourquoi le sujet est fréquemment abordé par des métiers divers. Par exemple, le clustering peut s'attacher à dresser des profils de clients d'une société, à permettre de rassembler des malades présentant les mêmes symptômes, à classer des documents, ou encore à réaliser du crédit scoring dans le domaine bancaire,…Il existe de nombreuses applications possibles au clustering, que l'on peut classer en trois groupes principaux [26]:

- *Knowledge Extraction :* concerne les applications qui utilisent le clustering pour extraire de la connaissance d'une base de données. Concrètement, ceci vise à déterminer des « *sousconcepts »* afin de donner du sens à l'information dont on dispose. On espère ainsi pouvoir s'attaquer à des tâches telles que la génération d'hypothèses (modélisation prédictive), le diagnostic médical en se basant sur des caractéristiques communes de patients,…
- *Data Reduction :* vise à utiliser le clustering pour segmenter la base de données en groupes homogènes et ainsi réduire la taille de l'ensemble des données sur lequel on travaille. Il s'agit donc de déterminer des « *sous-espaces »* de l'espace des données. La compression d'information et la segmentation d'images sont des cas concrets de ce type d'approches.
- *Profiling :* utilise le clustering pour détecter des « *sous-populations »* ayant des caractéristiques proches dans une base de données afin de pouvoir prendre des décisions particulières, adaptées à chaque sous-population séparément. Ceci touche directement les applications où l'on cherche à regrouper des clients

(Customer Relationship Management, en marketing), dans les transports, dans les banques, dans les commerces, dans les télécommunications, dans la gestion de ressources (énergie, stocks,…),… On trouve également de nombreuses utilisations dans le domaine de la classification de document par l'intermédiaire de ces exemples. On perçoit la richesse des méthodes de clustering, leur tendance à se retrouver sur le chemin de bon nombre d'acteurs différents de la vie quotidienne qui n'ont pas toujours a priori de points en commun.

I.5.2 Les différentes problématiques rencontrées dans le clustering

Pour mener à bien leur tâche, les méthodes de clustering doivent s'attaquer à diverses difficultés. Ce paragraphe vise à informer de quelques problèmes généraux.

- *Problématique des larges bases de données* : Pour être applicables à des bases de données contenant un grand nombre d'objets ou d'attributs, les méthodes de clustering développées doivent porter une attention particulière à leur complexité. La plupart des méthodes « classiques » de clustering sont facilement abordables, mais ne parviennent pas à s'appliquer à de grands jeux de données. Leurs complexités en temps de calcul et en espace sont en effet souvent trop importantes. Il en découle que certaines techniques sont mieux habilitées à traiter des grands jeux de données que d'autres.
- *Problème de la nature des données*: comme nous l'avons déjà évoqué, un objet est caractérisé par l'ensemble des valeurs prises par les attributs du jeu de données. Le terme « valeur » est à prendre au sens large. En effet, tous les attributs ne sont pas forcément chiffrables. Par exemple, on peut parler de la couleur d'une voiture, de la nationalité d'une personne. Or, on vient de mentionner le fait que la plupart des algorithmes de clustering comparent deux objets sur base de la mesure d'une distance. Le problème se pose alors pour les attributs non numériques.
- *Problème du contexte*: Le problème de la prise en compte du contexte revient à admettre que certains attributs peuvent être déterminants pour la création de certaines classes et sans importance pour d'autres classes. Un exemple consiste à s'intéresser au diagnostic de cancers dans le domaine médical. Ainsi, si l'on souhaite détecter des cancers du sein, il est évident que la valeur « femme » de l'attribut « sexe » sera déterminante par rapport à la valeur « homme », et pourtant le sexe du patient n'a aucune influence sur les cancers du cerveau. **Fig. I.8** montre un second exemple d'un jeu de données dans lequel sont présents plusieurs clusters caractérisés par certains attributs qui leur sont propres. Par exemple, si l'on considère le groupe formé par l'ensemble des objets représentés par des ronds, on remarque qu'il est caractérisé par de faibles valeurs sur la dimension *X* et des valeurs élevées sur la dimension *Z;* par contre, les valeurs possibles de ces objets sur la dimension *Y* couvrant l'ensemble de l'intervalle de définition de *Y,* cette dimension n'aide pas à les différencier des autres objets. Le sous-espace de description de ce groupe est donc réduit à X * *Z.* De même, le sous-espace de description du groupe d'objets représentés par des triangles est *Y* * *Z*. La plupart des méthodes de clustering ne prennent pas en compte la notion de contexte, d'autant plus qu'elles se basent en général sur une

notion de distance entre objets. En effet, la distance entre deux objets étant définie globalement, ces techniques ne pourront traiter le fait que cette distance peut varier selon le contexte.

Afin de prendre en compte le fait que certains attributs décrivant les données ont plus au moins d'importance pour la classification, une première possibilité consiste à utiliser une méthode de sélection, d'extraction ou de pondération d'attributs en prétraitement de l'apprentissage. La sélection d'attributs est le processus qui consiste à identifier le sous-ensemble des attributs le plus efficace à utiliser pour le clustering. L'extraction d'attributs correspond à l'utilisation d'une ou plusieurs transformations des attributs initiaux pour produire de nouveaux attributs pertinents. Et la pondération d'attributs consiste à donner un poids plus important à certains attributs considérés comme plus pertinents pour le clustering.

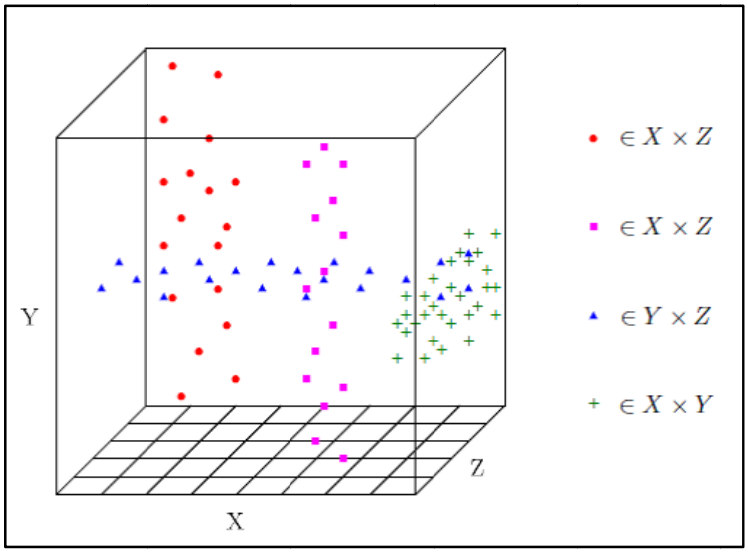


Fig. I.8 : Exemple de quatre clusters définis dans des sous-espaces différents.

- *Problème d'interprétation des résultats* : Un dernier problème est lié à la sortie des méthodes de clustering. Même si la procédure débouche sur un regroupement des objets, l'interprétation des groupes n'est pas toujours aisée. La plupart des algorithmes de clustering ont tendance à produire un type de clusters particulier, (les clusters ont une certaine forme en fonction de l'algorithme appliqué). Le problème est que les algorithmes détectent toujours des clusters de cette forme, même s'il n'y a aucune raison de déceler des clusters dans le jeu de données dont on dispose. Ainsi, si le but du clustering n'est pas seulement de faire de la compression d'information mais également d'extraire des comportements des différents groupes, il est important de s'assurer préalablement de la tendance du jeu de données à présenter une structure particulière en clusters. Dans le cas contraire, on risque simplement de réaliser des interprétations sur des groupes n'ayant pas la forme adéquate, voire en réalité aucune raison d'être ! [27].

I.5.3 Mesures de Similarités

La définition de la notion de similarité entre les données constitue une problématique centrale du clustering. Il existe trois concepts de similarité en clustering [26] : la similarité entre objets à maximiser pour deux objets appartenant au même cluster, et à minimiser pour deux objets appartenant à des clusters différents ; la similarité entre un objet et un cluster à maximiser si l'objet est associé au cluster pour une bonne *cohésion interne* du cluster; et la similarité entre clusters à minimiser pour une bonne *isolation externe* des clusters. Le problème est alors de définir cette notion de similarité. Typiquement, la similarité entre objets est estimée par une fonction calculant la distance entre ces objets (des détailles sont fournies dans l'annexe A). Une fois cette fonction distance définie, la tâche de clustering consiste alors à réduire au maximum la distance entre membres d'un même cluster tout en augmentant au maximum la distance entre clusters.

Similarités entre objets : Le choix de la mesure du rapprochement de deux objets est très important. Malheureusement, il s'agit trop souvent d'un choix non réfléchi, alors que les conséquences de celui-ci portent directement sur la forme des clusters qu'il est possible de détecter. Selon le type d'attribut, différentes mesures de similarité sont définies. Principalement, les attributs peuvent prendre des valeurs numériques, peu importe que celles-ci soient discrètes, continues ou par intervalles, il s'agit dans ce cas des attributs purement quantitatifs, ou pouvant prendre des valeurs non chiffrées, il s'agit des attributs qualitatifs (catégorielles), eux-mêmes se trouvent sous plusieurs figures : nominales (ou désordonnées) par exemple il s'agit d'une couleur, ou ordinales (ou ordonnées), par exemple , il s'agit du grade obtenu par un étudiant à une session d'examens. Finalement, dans le cas le plus fréquent, les données sont représentées par un mélange d'attributs quantitatifs et qualitatifs, il s'agit de ce cas d'attributs mixtes. Il existe différentes possibilités pour mesurer la distance entre deux objets sur base des valeurs prises par leurs attributs. Dans l'annexe A, une présentation de quelques mesures de distance couramment utilisées.

Similarité de deux clusters : Contrairement à la similarité entre objets, la similarité de deux clusters, ne nécessite pas de calcul complexe et ne fait intervenir que des concepts physiques. Supposons avoir choisi une manière de mesurer le rapprochement de deux objets (peu importe laquelle). Le problème de déterminer celui de deux clusters revient alors simplement à déterminer quels objets prendre dans chacun des clusters pour définir la mesure. Les possibilités les plus courantes sont reprises dans l'annexe A.

Rapprochement d'un objet et d'un cluster : Si l'on suppose avoir défini une mesure du rapprochement de deux objets, la définition de celui d'un objet et d'un cluster est presque immédiate puisqu'il suffit de choisir un point représentatif du cluster. On se ramène donc à la mesure du rapprochement de deux objets : l'objet en

question et le représentatif du cluster, pouvant être son centroïde (ou médoïde), l'objet le plus proche, l'objet le plus loin, ...

I.5.4 Différents problèmes liés à la notion de distance entre objets

Il existe plusieurs difficultés engendrées par la notion de distance qu'il est nécessaire de prendre en compte :

- *Problème d'échelles* : le premier obstacle apparaît suite aux différences de variations et d'unités dans les échelles des attributs de l'objet. Ainsi, à partir du moment où l'on calcule une distance sur base de différents attributs, il est important que les attributs soient ramenés à des valeurs comparables pour que la distance ait un sens. Plus précisément, si un attribut peut avoir des valeurs comprises dans [0,10] et un autre dans [50,100], il est évident que des différences dans les valeurs prises par le deuxième attribut contribueront en général davantage à augmenter la distance que les différences observées dans les valeurs prises par le premier attribut.
- *Prise en compte des attributs :* un autre problème est que tous les attributs sont pris en compte de la même manière dans le calcul d'une distance. Or, nous avons déjà mentionné le fait que certains attributs peuvent être plus pertinents que d'autres pour déterminer les clusters. Ceci peut être traité en allouant des poids aux différents attributs dans le calcul de la distance entre deux objets. Ainsi, on remplacera x_{ik} par $w_k . x_{ik}$, où w_k correspond au poids affecté à l'attribut k. La définition des poids devient difficile lorsque le nombre d'attributs est important. Dans ce cas, on peut recourir à la sélection d'attributs (et/ou extraction). On trouve aussi parfois des estimations des poids en fonction de la variance des attributs.
- *Type de données* : habituellement, les mesures de distance ne peuvent être appliquées que dans le cas où tous les attributs possèdent des valeurs numériques. Il n'est pas toujours aisé d'intégrer des attributs qualitatifs dans des expressions mathématiques.
- *Distorsions* : même si l'on standardise les attributs, des problèmes de distorsions dans le calcul de distances peuvent apparaître suite à la présence d'attributs corrélés.

Afin d'éviter que les attributs dont l'intervalle de définition des objets est plus large n'aient plus d'importance que les autres, une phase préalable de normalisation des données est souvent conduite. Typiquement, les valeurs des objets sur les attributs numériques sont normalisées entre 0 et 1. Cette phase sera cependant évitée si on se place dans le cadre d'une application où l'espace de description a un sens géométrique l'espace euclidien dans le cadre de la segmentation de bases de données spatiales par exemple. Enfin, il est également possible de prendre en compte les connaissances d'un expert du domaine dans le calcul de distance entre objets. Celui-ci affecte alors un poids *à* chaque attribut en fonction de l'importance de cet attribut pour le problème considéré.

I.5.5 Méthodes de clustering

On distingue deux grands types de méthodes de clustering: celles dites hiérarchiques qui fournissent des suites de classes emboîtées qui définissent une hiérarchie et celles dites non hiérarchiques (de partition) qui produit une partition de l'ensemble de départ en un nombre C de classes de même niveau.

Dans le cas du clustering par partition, plusieurs catégories de méthodes se distinguent fortement : le *clustering par critère du carré des écarts aux centroïdes* est une famille de méthodes de partition qui se basant sur un critère de « qualité ». Un exemple de ces méthodes est l'algorithme bien connu des K-means (K-moyennes) ; le *clustering statistique* est basé sur l'hypothèse que les données ont été générées en suivant une certaine loi de distribution, le but étant alors de trouver les paramètres de cette distribution, ainsi que les paramètres cachés déterminant l'appartenance des objets aux différentes composantes de cette loi ; Le *clustering stochastique* consiste à parcourir l'espace des partitions possibles selon certaines heuristiques, et à sélectionner celle qui optimise un critère donné ; le *clustering basé sur la densité* a pour but d'identifier dans l'espace les zones de forte densité entourées par des zones de faible densité pour la formation des classes; comme son nom l'indique, le *clustering basé sur les grilles* utilise une grille pour partitionner l'espace de description des objets en différentes cellules, puis identifie les ensembles de cellules denses connectées pour former les classes; Le *clustering basé sur les graphes* consiste à former le graphe connectant les objets entre eux et dont la somme des valeurs des arcs, correspondant aux distances entre les objets, est minimale, puis à supprimer les arcs de valeurs maximales pour former les classes; Enfin, la base du *clustering spectral* consiste à projeter itérativement les objets dans des sous-espaces de variance maximum, puis à utiliser une méthode de partitionnement dans de tels sous-espaces pour séparer les données.

Dans les méthodes de classification hiérarchique, il existe deux catégories principales de méthodes : ascendantes (CHA) ou descendantes (CHD). Les premières (dites aussi agglomératives) partent des individus isolés assimilés à des classes et procèdent, à chaque étape, par agrégation des deux classes les plus proches au sens de la norme choisie, jusqu'à ce qu'il n'y ait plus qu'une seule classe. Les secondes sont des méthodes de classification divisives, elles partent de l'ensemble des individus et procèdent par divisions successives de classes jusqu'à l'obtention de classes vérifiant certaines règles d'arrêt.

Différentes méthodes *hybrides* ont également été proposées qui mélangent les caractéristiques des méthodes hiérarchiques et des méthodes par partition, cherchant ainsi à bénéficier des atouts de chaque méthode. Enfin, une autre famille de méthodes ayant pour objectif de s'attaquer à la problématique du contexte. Portant le nom de méthodes de *subspace clustering* [103], elles ont pour objectif de cibler simultanément les groupes d'objets présents dans les données ainsi que les sous-espaces spécifiques dans lesquels ces groupes sont définis. Ces méthodes sont plus générales qu'une méthode de clustering qui utilise une phase initiale de sélection ou d'extraction d'attributs, car les sous-espaces sont spécifiques à chacun des clusters formés. Elles permettent de faire face au problème de la *malédiction de la dimensionalité* (*curse of dimensionality*) dans les bases de données contenant de nombreux attributs. De plus, elles permettent d'obtenir une description réduite

des clusters puisque chaque cluster est décrit par un nombre restreint d'attributs qui lui sont spécifiques. Deux grandes familles de subspace clustering se distinguent :

1. Les méthodes ascendantes sur les dimensions, considérant des sous-espaces de dimensionalité croissante et ciblant ensuite les groupes d'objets existant dans ces sous-espaces;

2. Et les méthodes descendantes sur les dimensions, utilisant différentes techniques permettant de cibler les sous-espaces spécifiques à certains groupes d'objets considérés.

I.5.6 Choix d'une méthode de clustering

Sur l'ensemble des problèmes envisageables, aucune méthode de clustering ne peut être considérée comme meilleure que toutes les autres. On considère plutôt que certaines méthodes sont plus adaptées que d'autres dans certains cas. Afin d'aider les utilisateurs à faire leur choix parmi les nombreuses méthodes existantes en fonction de l'application ciblée, nous proposons ici un ensemble de critères qui peuvent leur être associées.

- *Connaissances a priori* : Tout d'abord, les connaissances a priori requises de la part de l'utilisateur sont un premier critère important à prendre en compte dans le choix de la méthode ; il s'agit souvent, d'informations concernant le nombre de clusters recherchés, la distance minimale entre clusters disjoints, ou la densité minimale à l'intérieur des clusters.
- *Présentation des résultats :* On peut ensuite différencier les méthodes en fonction de leur façon de présenter les résultats; on peut en particulier différencier les méthodes fournissant en sortie une hiérarchie de clusters des méthodes fournissant en sortie une partition de l'ensemble des objets.
- *Complexité* : La complexité des méthodes est évidemment également un critère important à considérer ; en particulier, il est admis que la complexité des méthodes doit être linéaire en fonction du nombre d'objets dans le cas de larges bases de données; il faut dès lors éviter toute méthode basée sur le calcul des distances deux à deux entre objets, menant à une complexité quadratique en fonction du nombre d'objets;
- *Déterministe* : Il est également possible de distinguer les méthodes déterministes des méthodes stochastiques: avec les mêmes données en entrée, un algorithme déterministe exécutera toujours la même suite d'opérations et fournira donc toujours le même résultat alors qu'une méthode stochastique pourra donner des résultats différents car elle permet l'exécution d'opérations aléatoires;
- *Incrémental :* Une autre caractéristique qui peut se révéler importante pour les méthodes de clustering est leur capacité de traiter des données de façon incrémentale, c'est-à-dire en les intégrant au fur et à mesure de leur arrivée dans l'algorithme; à l'inverse, une méthode non-incrémentale va considérer un ensemble de données fournies en entrée et sera exécutée sur cet ensemble de données, et si par la suite une nouvelle donnée est fournie en entrée de l'algorithme, celui-ci devrait être relancé à nouveau ;

- *Any-time* : De la même façon, on peut différencier les méthodes *any-time,* capables de fournir un résultat intermédiaire, même sous-optimal, à n'importe quel moment du déroulement de la méthode, des méthodes nécessitant l'accomplissement total de la méthode avant d'être capable de fournir un résultat.
- *Hard* : On peut ensuite différencier les méthodes *hard* qui associent à chaque objet un unique cluster, des méthodes *soft* qui associent à chaque objet un degré variable d'appartenance à chacun des clusters formés ; notons d'ailleurs dès à présent qu'un clustering *soft* peut être converti en clustering *hard* en assignant chaque objet au cluster dont la mesure d'appartenance est la plus forte.
- *Tolérance au bruit :* Il peut également être utile pour une méthode d'être capable de gérer le bruit qui peut exister dans les données, c'est-à-dire la possible présence d'objets qui ne suivent pas la distribution générale des autres objets.
- *Tolérance à l'effet de chaîne* : En particulier, il s'avère souvent utile d'être capable de faire face au problème de l'effet *de chaîne,* illustré par la figure (I.9) et qui fait que des clusters proches, mais distincts peuvent être fusionnés s'il existe une chaîne d'objets qui les relie.

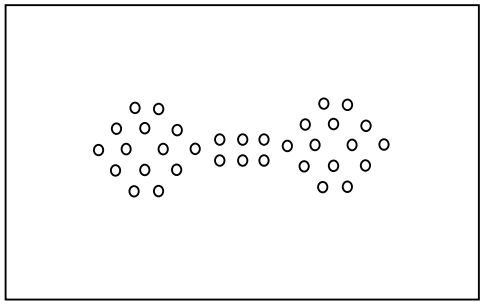


Fig. I.9 : Problématique de l'effet de chaîne.

- *Tolérance aux clusters de tailles ou de densités variées* **:** Enfin, un autre critère qui peut s'avérer important est le type de clusters que la méthode est capable de détecter; en particulier, il est souvent utile d'être capable de détecter des clusters de tailles ou de densités variées, comme dans le cas de la figure (I.13) ;

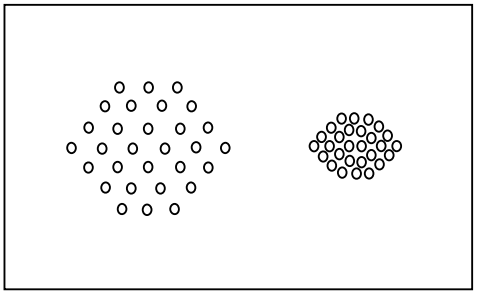


Fig. I.10: Problématique des clusters de tailles et de densités variées.

- *Tolérance aux clusters de formes quelconques ou concentriques* **:** De même, il est dans certains cas utile qu'une méthode de clustering soit capable d'identifier des

clusters de forme quelconque, et aussi éventuellement des clusters concentriques, c'est-à-dire inscrits les uns dans les autres, comme dans le cas de la figure (I.11).

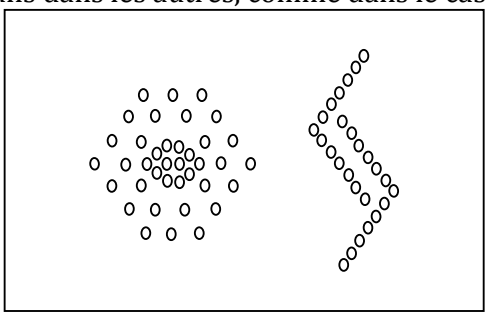


Fig. I.11 : Problématique des clusters de formes variées et concentriques.

I.5.7 Évaluation des performances d'un classifieur

Dans la plupart des applications, tant que les algorithmes de clustering ne sont pas supervisés, les partitions finales des données exigent une certaine forme d'évaluation. En général, l'évaluation des résultats de clustering peut répondre, entre autres aspects, des questions comme : où se situent nos partitions ? Est-il possible de trouver une partition meilleure? L'évaluation des résultats d'une méthode de clustering, de même la comparaison de différentes méthodes, constitue une problématique importante ; on peut envisager, souvent, différents regroupements pertinents possibles pour un même jeu de données, car le problème est subjectif par nature. Nous décrivons dans ce paragraphe les critères les plus souvent employés pour mesurer la qualité des résultats d'algorithmes de clustering. En pratique, il existe trios méthodes principales pour mesurer la qualité des résultats d'algorithmes de clustering.

1. Utiliser des données artificielles pour lesquelles le regroupement attendu est connu, et comparer les résultats obtenus aux résultats attendus. On parle dans ce cas *d'évaluation externe.* Mais les méthodes sont alors évaluées uniquement sur les distributions spécifiques correspondantes, et les résultats sur données artificielles ne peuvent pas être généralisés aux données réelles.

2. Utiliser des critères numériques, tels l'inertie intra-cluster ou la séparation inter-clusters. On parle dans ce cas *d'évaluation interne.* Mais de tels critères ont une part importante de subjectivité, car ils utilisent des notions prédéfinies de ce qu'est un bon clustering. Par exemple, la séparation des clusters n'est pas toujours un bon critère à utiliser car, parfois, des clusters qui se chevauchent peuvent être plus pertinents.

3. Utiliser un expert pour évaluer le sens d'un clustering donné dans un champ d'application spécifique. Mais s'il est possible à un expert de dire si un regroupement donné a un sens, il est beaucoup plus problématique de quantifier son intérêt ou de dire si un regroupement est meilleur qu'un autre. De plus, l'intérêt de la méthode ne peut être généralisé à différents types de jeux de données.

Deux exemples pour l'évaluation externe sont : le taux de reconnaissance et le taux d'erreurs. Ces taux sont évalués grâce à la base de données entière. Elles sont

étiquetées par leur classe réelle d'appartenance afin de pouvoir vérifier les réponses du classifieur. Les performances en termes de taux de reconnaissance sont alors déterminées en présentant au classifieur chacun des exemples de la base et en comparant la classe donnée en résultat à la vraie classe. En considérant que la base contient *N* individus et que sur ceux-ci *Ncorrects* sont biens classés par le système, le taux *de reconnaissance* est simplement défini par :

$$\tau_{reco} = \frac{N_{correct}.100}{N}$$

$$\tau_{err} = \frac{N_{err}.100}{N}$$

Le *taux d'erreur* τ_{err} est défini à partir du nombre d'individus N_{err} mal classés. Lors de l'exploitation du système dans des conditions réelles d'utilisation, d'autres critères peuvent aussi devenir très importants. Ainsi, s'il existe des contraintes sur la vitesse de traitement, comme par exemple pour des applications interagissant directement avec l'utilisateur, le *temps de reconnaissance* devient un critère important. Si l'on considère l'embarquement du système sur des machines aux ressources limitées, la *taille mémoire* du modèle ou d'une façon plus générale le *nombre de paramètres* nécessaires à la modélisation sont des critères de compacité importants à prendre en compte.

I.6 Conclusion

L'idée de base que nous avons retenue est que dans un processus de reconnaissance des formes ; les formes les plus pertinentes conduisent aux meilleurs sous-ensembles des caractéristiques, et par conséquent, le plus simple classificateur sera appliqué, enfin les meilleures décisions seront prises. Les méthodes qui reposent sur un apprentissage non supervisé, sont des outils essentiels pour modéliser un problème de reconnaissance des formes et ne nécessitent pas une connaissance a priori de l'appartenance aux classes pour extraire automatiquement une structure à partir des données. Dans le cadre de nos travaux, les formes sont considérées incomplètement définies. En introduisant les principes de la théorie des sous-ensembles flous qui, en plus de la robustesse face aux imprécisions, permet de synthétiser les données sous une forme plus interprétable. L'introduction de la logique floue dans les algorithmes de classification non supervisée constitue alors une alternative séduisante comme nous allons le voir dans le chapitre III.

Chaque application du clustering, ayant un objectif différent, et donc un critère de qualité différent. La première étape de la mise en œuvre ou de l'utilisation d'une méthode de clustering est d'identifier l'application visée afin d'identifier ses besoins.

Quelle que soit l'approche de clustering envisagée, elles reposent sur l'utilisation d'une mesure de similarité. Le choix d'une mesure de distance entre objets est très important. Malheureusement, trop souvent, il s'agit d'un choix arbitraire, sensible à la représentation des objets, et qui traite tous les attributs de la même manière. Une solution pour pallier à cette limitation est celle de la prise en compte de la connaissance d'un expert, qui identifiera certains attributs,

considérés comme plus pertinents que d'autres pour le problème considéré, et leur attribuera un poids plus important lors du calcul des distances entre objets.

Généralement, le problème de reconnaissance des formes complexes avec orientation, emplacement et échelle arbitraire reste sans solution. Néanmoins, procéder à standardiser les attributs, c'est-à-dire à supprimer la dépendance avec leurs échelles et unités, en ramenant les attributs à une moyenne nulle et un écart-type unitaire s'avère efficace pour résoudre le problème de l'échelle.

Les méthodes de clustering doivent porter une attention particulière à leur complexité, pour être applicables à des bases de données contenant un grand nombre d'objets ou d'attributs. Typiquement, la meilleure solution est de développer des méthodes qui dépendent de façon linéaire du nombre d'objets et d'attributs. Plus amples détails sont fournie dans l'annexe B.

Pour l'évaluation des résultats, on commence par le test de l'efficacité de la méthode dans différentes situations contrôlées. Il est ensuite nécessaire de confronter la méthode à des cas d'applications réels. Il faut alors disposer d'un expert du domaine étudié pour qu'il mette en avant la pertinence des résultats fournis. Cependant, même si une méthode donne des résultats satisfaisants lors de ces deux étapes, son efficacité n'est pas garantie dans d'autres cas d'applications réelles car, les propriétés des données et les critères d'intérêt du clustering peuvent varier d'une application à l'autre. Le dernier chapitre donne un ensemble de critères pour l'évaluation interne, externe et relative des résultats du clustering.

Chapitre II
Techniques de clustering

II.1 Introduction

Le clustering est un outil fréquemment utilisé dans bon nombre de disciplines. Par ailleurs, dès les années 60 [63], [97], [6], [80], des milliers d'articles ont été publiés en mettant en jeu soit de nouvelles techniques de clustering, soit des versions améliorées de méthodes déjà connues. Il est évident, cependant, qu'aucune de ces techniques ne peut prétendre à une efficacité universelle. En effet, une méthode donnée de clustering peut être bien adaptée à un certain type de situations, et s'avérée complètement inutile face à certaines autres catégories de données. Et c'est ce qui explique la multitude d'approches diverses dédiées. En effet, il serait très difficile de fournir une liste exhaustive englobant toutes ces dernières. Au cours de ce chapitre, nous avons choisi, tout d'abord, de présenter les propriétés générales des techniques de clustering, et sur la base d'hypothèses choisies, on essaye d'établir un classement particulier. Les méthodes présentées seront discutées objectivement, tant au niveau de l'algorithme que des clusters qu'elles fournissent. Il est évident que les hypothèses choisies influenceront le classement obtenu. Ceci n'est pas limitatif ; selon les propriétés choisies, on peut avoir différents classement possibles. Notons encore qu'il y aura autant de variantes à une méthode qu'il y a de choix de mesures de distance. Les méthodes de clustering, exposées dans ce chapitre, seront considérées comme basées sur une mesure de distance euclidienne entre deux objets.

Après le rappel de quelques propriétés générales des méthodes de clustering, dans la troisième section, nous présentons, en se basant sur certaines hypothèses, une première subdivision des méthodes de clustering en quatre catégories distinctes. Les deux familles des méthodes : de partition et hiérarchiques seront ensuite étudiées successivement dans la quatrième et la cinquième section. Pour chacune des méthodes, nous présentons d'abord son fonctionnement général, avec éventuellement différentes alternatives possibles. Ensuite, dans la sixième section, nous détaillons ses atouts et ses limites, que nous résumons finalement à l'aide des caractéristiques associées aux méthodes de clustering.

II.2 Propriétés générales des méthodes de clustering

Diverses propriétés peuvent s'attribuer à chaque technique de clustering [78], [26], reprises ci-dessous :

- *Méthode hard ou fuzzy* : encore dite méthode dure ou floue. Dans une approche dure, les clusters générés sont mutuellement exclusifs. À l'opposé, une méthode floue alloue des clusters aux objets avec différents degrés d'appartenance. Un objet peut donc appartenir en partie à plusieurs clusters.
- *Méthode déterministe ou* stochastique : cette propriété est liée au caractère de la sortie de la méthode. Dans une approche déterministe, le résultat du clustering sera toujours identique si l'on entre plusieurs fois le même jeu de données, de la même façon. À l'opposé, une méthode stochastique génère une partition résultant de choix à diverses étapes du déroulement de la méthode. Ces choix n'étant pas constamment identiques, la sortie de la méthode variera, même si l'on entre à plusieurs reprises le même jeu de données, de la même façon.
- *Méthodes agglomératives ou divisives* : la distinction s'établit sur le principe général suivi par la méthode. Dans les méthodes agglomératives, la démarche est de partir de petits clusters nombreux et ensuite de les regrouper progressivement en clusters plus conséquents. Par contre, les méthodes divisives arrivent à une partition de l'ensemble des données en « divisant » successivement de gros clusters en groupes plus petits.
- *Méthode monothétique ou polythétique* : cette caractéristique traite de la manière de prendre en compte les attributs des objets pour arriver à une partition. Dans une approche polythétique, tous les attributs sont considérés simultanément. C'est typiquement le cas de méthodes basées sur les distances (au sens large), pour lesquelles le calcul du rapprochement de deux objets fait intervenir tous les attributs de l'objet. Dans le cas d'une méthode monothétique, les attributs sont considérés un à un, chacun d'eux amenant successivement une partition de l'espace des données. Ce type d'approche amène spontanément à opter pour une description des clusters à l'aide de conjonctions logiques.
- *Méthode incrémentale ou non incrémentale* : cette propriété s'appuie sur la manière de classer les objets. Dans une approche non incrémentale, les objets sont regroupés selon certains critères et peuvent changer de groupe en cours d'exécution de l'algorithme. Par contre, dans une méthode incrémentale, les objets sont placés selon leur ordre d'arrivée dans un cluster et n'en bougent plus.

De nombreuses combinaisons sont possibles entre ces propriétés. Tous ces recoupements génèrent au moins chacun une méthode, ou plus généralement une catégorie de méthodes. Dans les paragraphes suivants, nous présenterons les méthodes de clustering les plus couramment rencontrées.

II.3 Première subdivision des méthodes de clustering

Dans la suite, on ne considère que des méthodes hard de type polythétique qui traitent des bases de données de taille raisonnable avec attributs numériques. On peut ainsi, annoncer une partition en quatre catégories de méthodes présentée ci-dessous :

- *Les méthodes de partition* : ces méthodes visent à optimiser une partition initiale de manière itérative en optimisant une fonction critère de « qualité » (différente de l'évaluation de la qualité de la partition, il s'agit d'un critère de construction ici).
- *Les méthodes hiérarchiques* : ces méthodes utilisent les données initiales soit pour les regrouper au fur et à mesure (approches agglomératives), soit au contraire pour effectuer des divisions successives (approches divisives). Typiquement, elles reposent sur le choix d'un critère d'agrégation, déterminant la manière d'agglomérer deux clusters ou de diviser un cluster.
- *Error based clustering* : la plupart des méthodes de clustering ne prennent pas en compte l'éventualité d'incertitudes sur les données. Ces erreurs peuvent pourtant influencer les résultats de la segmentation. Nous n'aborderons pas davantage ces méthodes dans le cadre de ce travail.
- *Subspace clustering* : consistent à détecter des zones denses par projections des données dans des sous-espaces de l'ensemble des données. Nous n'aborderons pas non plus ce type de techniques dans la suite.

Différentes méthodes hybrides ont également été proposées qui mélangent les caractéristiques des méthodes hiérarchiques et des méthodes par partition, cherchant ainsi à bénéficier des atouts de chaque méthode. Le tableau suivant fournit une comparaison schématique des méthodes de partition et des méthodes hiérarchiques. Nous reprendrons ces différents points dans la suite.

Méthodes de partition	Méthodes hiérarchiques
Généralement itératives : modification de la partition à une étape	Généralement progressives : évolution de la partition à une étape
Obtention d'une partition : une seule partition à la sortie de la méthode	Obtention d'un dendrogramme : partitions successives à la sortie de la méthode
Moins souples	Plus souples
Moins coûteuses en CPU	Plus coûteuses en CPU
Critère de « qualité »	Critère d'agrégation

Tableau II.1:Comparaison des méthodes de partition et des méthodes hiérarchiques

II.4 Méthodes de partition

Dans le cas du clustering par partition, plusieurs méthodes se distinguent fortement : le clustering basé sur l'optimisation d'inertie est une famille de méthodes de partition la plus antique, par exemple : l'algorithme de regroupement autour de centres mobiles [63]. Le clustering statistique est basé sur l'hypothèse que les données ont été générées en suivant une certaine loi de distribution, le but étant alors de trouver les paramètres de cette distribution, ainsi que les paramètres cachés déterminant l'appartenance des objets aux différentes composantes de cette loi. Le clustering stochastique consiste à parcourir l'espace des partitions possibles selon certaines heuristiques, et à sélectionner celle qui optimise un critère donné ; le clustering basé sur la densité a pour but d'identifier dans l'espace les zones de forte densité entourées par des zones de faible densité pour la formation des clusters ; comme son nom l'indique. Le clustering basé sur les grilles utilise une grille pour partitionner l'espace de description des objets en différentes cellules, puis identifie les ensembles de cellules denses connectées pour former les clusters ; Le clustering basé sur les graphes consiste à former le graphe connectant les objets entre eux et dont la somme des valeurs des arcs, correspondant aux distances entre les objets, est minimale, puis à supprimer les arcs de valeurs maximales pour former les clusters ; enfin, la base du clustering spectral consiste à projeter itérativement les objets dans des sous-espaces de variance maximum, puis à utiliser une méthode de partitionnement dans de tels sous-espaces pour séparer les données. Dans les paragraphes suivants, nous présenterons brièvement les principales méthodes de clustering par partition, ainsi qu'une liste des caractéristiques principales qui peuvent leur être associées.

II.4.1 Méthodes de clustering basées sur l'optimisation d'inertie

Cette famille de méthodes de clustering se base sur l'optimisation d'un critère de qualité (construction) qui minimise la somme des carrés des écarts (distance) entre chaque objet et le centroïde du cluster courant qui lui est associé. Un critère usuel de regroupement consiste à optimiser un critère de type inertie. Soit P une partition d'un ensemble $X = \{x_1, x_2, \dots, x_n\} = X_1 \cup X_2 \cup \dots \cup X_c$ à n éléments en c sous-ensembles. Soient $v_1, v_2, \dots, v_c$ les centres de gravité de ces groupes et I_i $(i = 1, 2, \dots, c)$ leurs inerties. Rappelons que l'inertie est la moyenne des carrés des distances au centre de gravité :

$$I_i = \frac{1}{n_i} \sum_{x_j \in X_i} d^2\left(v_i, x_j\right) \qquad \text{(II. 1)}$$

Où n_i est le cardinal du groupe X_i (nombre d'objets dans la classe). (Nous avons supposé, ici, que tous les points du nuage ont le même poids. Dans les ouvrages de statistiques, on trouve généralement cette formule légèrement modifiée en supposant des poids différents). Le théorème de König-Huyghens [11]

stipule que l'inertie totale du nuage, I, autour du centre de gravité global, g, des n points x_j $(j = 1, 2, \ldots, c)$ est égale à la somme de deux termes:

$$I = I_w + I_b \quad \text{(II.2)}$$

Le premier terme : $I_w$correspond à l'inertie intraclasse (II.3), et le deuxième I_b à l'inertie interclasse (II.4) :

$$I_w = \sum_{i=1}^{c} \frac{n_i}{n} I_i \quad \text{(II.3)}$$

$$I_b = \sum_{i=1}^{c} \frac{n_i}{n} d^2(v_i, g) \quad \text{(II.4)}$$

L'inertie intraclasse constitue une mesure de la compacité des classes. Une classe est d'autant plus compacte que son inertie est faible. Ou encore, l'inertie d'une classe donnée est d'autant plus élevée que les points lui appartenant sont éloignés de son centre. L'inertie interclasse, quant à elle, reflète la séparabilité des différentes classes. Comme on peut le voir sur l'équation (II.4), cette inertie est d'autant plus élevée que les différents centres v_i, sont loin du centre de gravité global g, du nuage. Un critère usuel de regroupement consiste à minimiser l'inertie intraclasse I_w ,afin que les classes soient, en moyenne, bien homogènes. Ceci revient, également, à chercher le maximum de l'inertie interclasse I_b, assurant, ainsi, l'obtention de classes bien séparées. Les méthodes des centres mobiles et des nuées dynamiques sont basées sur ce principe (minimisation de l'inertie intraclasse). Un groupe X_i sera constitué des points de X plus proches de v_i (dans le cas des centres mobiles) que de tout autres centres. La méthode des centres mobiles peut être considérée comme un cas particulier de techniques connues sous le nom de nuées dynamiques étudiées en détail par Diday [42].

II.4.1.1 Méthode des centres mobiles

Cette méthode consiste à construire une suite de partitions dont les inerties intraclasse correspondantes vont en décroissant. À une étape donnée, on commence par chercher les centres de gravité des différents groupes, et on affecte ensuite chacun des points au groupe correspondant au centre qui lui est le plus proche (figure II.3). On montre que, d'une partition à l'autre, l'inertie intraclasse décroît [24]. La partition initiale peut être construite en choisissant c points au hasard à partir desquels on divise l'espace en c domaines polyèdraux convexes séparés par les hyperplans médiateurs (en dimension deux, les médiatrices) de ces c points. Il peut arriver, en pratique, qu'une classe se vide. Il est possible, dans ce cas, de tirer un nouveau centre au hasard. Le centre d'un groupe quelconque peut être vu comme le barycentre de l'ensemble des points x_j, (j = 1, 2, ..., n), affectés de coefficients ne prenant que deux valeurs : 0 ou 1, et qui correspondent, pour chaque point, à la valeur prise, en ce point, par la fonction caractéristique du groupe étudié.

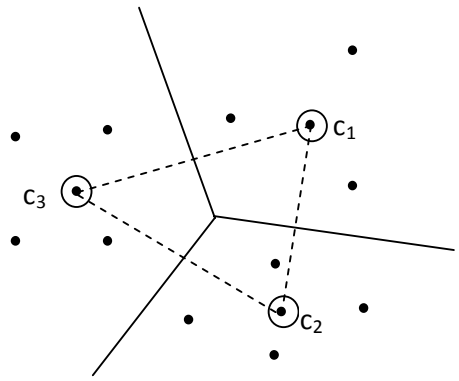


Fig. II.1: Principe de la méthode des centres mobiles. c_1, c_2 et c_3 sont les centres de gravité des 3 groupes de points correspondant à la partition courante. La partition suivante est obtenue en affectant chacun des points au centre qui lui est le plus proche. Cette opération s'accompagne de la diminution de l'inertie intraclasse.

II.4.1.2 Méthode des nuées dynamiques

Cette méthode est une extension des centres mobiles. Les différents groupes ne sont plus représentés par leurs centres de gravité uniquement, mais par leurs "noyaux" [41]. Un noyau peut être un ensemble de q points (les plus centraux), un axe principal ou un plan principal. Il faut donc disposer d'une fonction qui, à un ensemble donné, associe son noyau. Il faut ensuite réaffecter les points aux différents noyaux. On itère ensuite les deux phases : affectation, représentation, jusqu'à convergence du critère choisi. La partition finale dépend cependant de la configuration des noyaux de départ. On applique alors la méthode pour s tirages différents, les ensembles d'éléments ayant été regroupés pour tous les tirages sont appelés « formes fortes » [43]. Les groupements correspondant aux formes fortes présentent, donc, le plus haut degré de confiance et mettent en évidence les zones à forte densité.

II.4.1.3 Méthode des k-moyennes

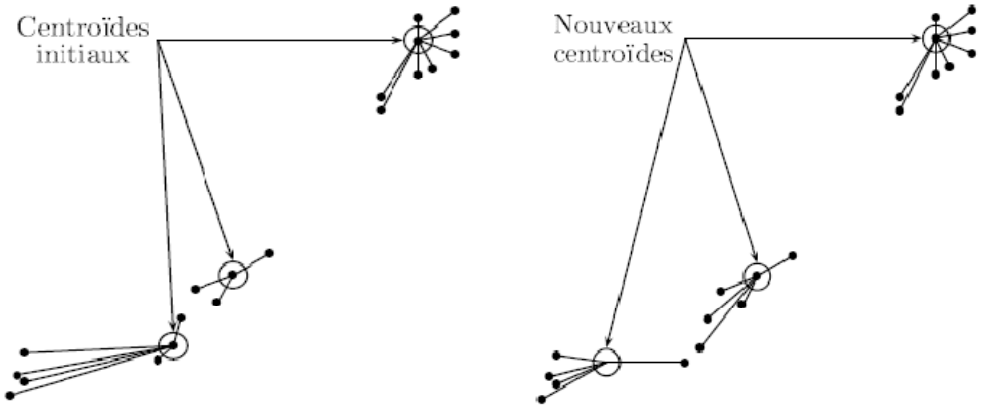
L'algorithme de regroupement autour de centres mobiles est généralement imputé à Forgy [63]. En réalité, de nombreux travaux ont été menés parallèlement sur le thème des centres mobiles, introduisant des variantes [6], [97], ou des généralisations [41]. Cette méthode est connue, en anglais, sous le nom de k-means. La méthode des k-moyennes est imputée à MacQueen [97], dont l'algorithme commence également par un tirage des centres, mais contrairement à la technique des centres mobiles, chaque réaffectation d'un point entraîne une modification immédiate du centre correspondant.

Algorithme des K-means

La méthode des K-means est très couramment utilisée dans bon nombre d'applications. Il est donc utile d'en discuter un peu plus longuement que d'un simple point de vue descriptif. L'algorithme peut se présenter comme :

1. Choisir au hasard le centroïde de chacun des *k* clusters.
2. Attribuer chaque objet au cluster dont le centroïde lui est le plus proche.
3. Recalculer les positions des nouveaux centroïdes et mettre à jour les assignations des objets aux clusters en fonction de leur proximité aux nouveaux centroïdes.
4. Répéter les étapes 2 et 3 jusqu'à convergence, c'est-à-dire jusqu'à ce que les centroïdes ne bougent plus (ou ne subissent pas de déplacement significatif).

La figure ci-dessous présente un exemple de clustering par K-means.



(a) : Etape 1 et 2 (b) : Une itération de l'etape3

Fig.II.2: clustering k-means.

Discussion de la méthode des K-means

La méthode des K-means possède de nombreux avantages :

- la facilité.
- le résultat de la méthode ne dépend pas de l'ordre d'entrée des objets.
- la méthode a une complexité linéaire.
- la méthode s'adapte aux larges bases de données.
- la convergence rapide.

Mais elle possède également des inconvénients :

- le nombre *k* de clusters doit être fixé à l'avance et influence la sortie.
- résultat est sensible à la situation de départ, tant au niveau du nombre de clusters qu'au niveau des positions initiales données aux centroïdes.
- la méthode peut converger vers des optima locaux.

- la méthode gère difficilement la détection de clusters mal isolés et/ou de forme allongée.
- la méthode est très sensible aux éléments marginaux.

De manière générale, la méthode des K-means fournira de bons résultats sur un jeu de données où les clusters à détecter sont compacts, bien isolés et de forme sphérique.

Remédiation aux problèmes des K-means

La méthode des K-means étant d'usage fréquent (surtout suite à l'avantage de sa complexité linéaire), différentes variantes ont été développées pour remédier aux difficultés présentées au paragraphe précédent. Pour permettre de choisir un nombre de clusters le plus cohérent possible avec le jeu de données, on utilisera [40] :

- *une analyse des données :* visualisation des données par analyse discriminante [95].
- *une phase de « split and merge » :* qui consiste à permettre des séparations et fusions dans les clusters à la sortie de la méthode. Typiquement, un cluster sera divisé si sa variance dépasse une valeur seuil. D'autre part, deux clusters seront fusionnés lorsque la distance séparant leur centroïde est inférieure à un certain seuil. Une méthode s'appuyant sur cette démarche est la méthode ISODATA [78].

Pour faire face à la dépendance de la solution finale avec le choix des centres initiaux des clusters (optimum local), deux approches sont reprises ci-dessous [40]:

- *approche simple aléatoire : k* étant supposé choisi convenablement, on exécute à plusieurs reprises la méthode, avec le même nombre de clusters, mais avec des positions initiales des centres différents. On garde la solution donnant la plus petite valeur à la fonction critère.
- *approche par groupements forts :* pour un nombre *k* fixé, on exécute plusieurs fois la méthode. Des groupes d'objets restant constamment associés à travers les différentes partitions générées sont appelés groupes forts (ou stables). L'idée est alors d'utiliser les centroïdes de ces groupes parmi les positions initiales des centroïdes à fixer afin de relancer la méthode une dernière fois. Notons que cette démarche peut aussi permettre de corriger une valeur de *k* qui s'avérerait inappropriée.

II.4.2 Le clustering statistique

L'approche statistique du clustering consiste à supposer que les données ont été générées selon une loi paramétrique connue, mais de paramètres inconnus. On considère que les données ont été générées par un mélange de distributions de probabilités, la problématique étant alors de trouver, en fonction des données observées, les paramètres de ces distributions, ainsi que les probabilités du mélange, et les paramètres cachés du modèle correspondant

aux affectations des objets aux différentes composantes du mélange. Par exemple, lorsque les données sont de type numérique, on suppose classiquement qu'elles ont été générées selon un mélange de distributions gaussiennes [128], caractérisées par leur centre et par leur matrice de covariance, chacune de ces distributions ayant une certaine probabilité d'être utilisée.

Le principe est donc d'utiliser un modèle statistique permettant d'approcher au mieux la distribution du jeu de données, sur base de paramètres de départ. Ceci permet de déterminer les attributions de clusters aux objets. Ensuite, on peut recalculer les paramètres du modèle sur base des nouvelles attributions réalisées. Cette approche est appelée « EM » [78], [26]. Elle comprend les deux phases suivantes :

- *Espérance*: attribuer les clusters aux objets sur base des paramètres courants du modèle.
- *Maximisation*: déterminer les nouveaux paramètres suite aux dernières attributions réalisées.

On améliore donc le modèle d'itération en itérant jusqu'à atteindre un critère d'arrêt de convergence.

Si l'on fait l'hypothèse que les matrices de covariance des clusters correspondent toutes à la matrice identité, et que les probabilités du mélange sont toutes égales, alors on obtient le modèle utilisé par la méthode K-means [29]. En fait, ce modèle généralise celui utilisé par la méthode K-means. Ainsi que pour la méthode K-means, cette méthode statistique est confrontée à la problématique de la spécification du nombre de clusters recherchés, et est également sensible à la solution initiale proposée. Pour pallier à ces limitations, les solutions sont les mêmes : générer plusieurs modèles en faisant varier le nombre de clusters recherchés, et utiliser un critère statistique pour choisir le modèle le plus approprié aux données, en considérant pour chaque modèle plusieurs initialisations aléatoires. Plusieurs critères ont été proposés, le lecteur intéressé peut se tourné vers [27].

II.4.3 Le clustering stochastique

Les approches basées sur la recherche stochastique visent également à assurer l'obtention de la partition optimale de l'espace des données (optimisant l'erreur quadratique par exemple). Dans ce cas, la technique consiste simplement à parcourir l'espace des solutions possibles et à sélectionner la solution rencontrée qui maximise le critère cible choisi. Évidemment, l'espace des solutions étant souvent beaucoup trop vaste pour être parcouru entièrement, des heuristiques sont généralement utilisées pour le parcourir. Typiquement, une solution considérée dans ce cadre est l'ensemble des associations des objets aux clusters, le nombre de clusters étant fourni par l'utilisateur. À chaque étape, une nouvelle solution est considérée, et conservée pour l'étape suivante si elle fait partie des solutions optimales du point de vue du critère ciblé. L'originalité des différentes méthodes stochastiques existantes

réside alors dans leur façon de parcourir l'espace des solutions possibles, c'est-à-dire dans la technique utilisée pour proposer une nouvelle solution courante à évaluer. En pratique, les méthodes mises en œuvre sont souvent construites de manière à trouver un bon compromis entre l'exploration nécessaire d'une partie importante de l'espace des solutions, et l'exploitation des solutions optimales identifiées [27]. Trois types de méthodes courantes qui appartiennent aux familles d'heuristiques (métaheuristiques) basées sur la recherche Tabou, le recuit simulé, ou les algorithmes géniques [105].

- La recherche Tabou se base sur un principe assez simple qui consiste à explorer l'espace des solutions en stockant les dernières solutions rencontrées et en s'interdisant de les revisiter. On force ainsi la diversification de la recherche.
- Le recuit simulé, souvent catalogué comme procédure lente, trouve son origine dans le principe métallurgique du recuit. Ainsi, on autorise généralement de plus grandes dégradations de la solution en début de procédure. Ensuite, au fur et à mesure de l'avancée de l'algorithme, un paramètre (appelé la température) contrôle l'autorisation, de plus en plus rare, de dégrader la solution courante. La nouvelle solution à chaque étape est donc choisie de plus en plus proche de la solution précédente.
- Enfin, la technique des *algorithmes génétiques* considère plutôt des populations de solutions potentielles, et combine ces différentes solutions pour obtenir à l'étape suivante un autre ensemble potentiel de solutions, parmi lesquelles sont conservées celles qui optimisent le critère cible choisi.

Une autre technique envisagée dans de nombreux cas est de considérer qu'une solution correspond à un ensemble de *c*-médoïdes, c'est-à-dire un ensemble de *c* objets de la base servant de points centraux des clusters à identifier. Les autres objets sont alors associés au cluster dont le médoïde est le plus proche. Et une méthode pour parcourir l'espace des solutions possibles consiste alors à modifier l'un des médoïdes, et à le conserver pour l'itération suivante si la solution est améliorée par ce changement. Les caractéristiques de telles méthodes dépendent très fortement du critère cible choisi. Ainsi, si c'est l'erreur quadratique qui est retenue, alors la méthode ne peut faire face à des clusters proches dont les tailles sont très variées. Si au contraire le critère choisi est un test statistique basé sur une distribution gaussienne des données, associant à chaque cluster un centre et une matrice de covariance, alors la méthode sera capable de gérer de tels clusters [27].

II.4.4 Clustering basé sur la densité

Une autre famille de méthodes utilise la notion de densité pour détecter les clusters. Elle les considère comme des régions homogènes de haute densité, entourées par des régions de faible densité. Un exemple d'algorithme est appelé DBSCAN [57], est ainsi basé sur l'utilisation de deux paramètres fournis en entrée de l'algorithme et contrôlant la notion de densité du voisinage d'un objet. Il repose sur l'utilisation de deux paramètres liés à la densité : le rayon maximum du voisinage d'un objet (R_{max}) et le nombre minimum d'objets qui

doit être contenu dans ce voisinage (R_{min}). Pour déceler des voisinages denses, on procède comme suit :

1. Choisir aléatoirement un objet du jeu de données parmi les objets non encore classés.
2. Vérifier que son voisinage respecte le critère de densité, c'est-à-dire qu'il y a au moins O_{min} objets dans un rayon de R_{max} autour de l'objet concerné.
3. Si le critère de densité est respecté, intégrer tous les objets correspondants dans un cluster et répéter le procédé avec ces objets. On construit les clusters par agglomérat de « voisinages denses voisins » comme l'illustre la figure II.3. Sinon, aller à l'étape 4.
4. S'il reste des objets non traités, retourner à l'étape 1. Sinon, fin de la procédure.

Cet algorithme sera assez performant si le choix des paramètres R_{max} et O_{min}, est adéquat. En effet, selon la valeur donnée à R_{max}, on peut obtenir des situations de sous-partitionnement ou de surpartitionnement comme dans le cas de l'utilisation de grilles. Par contre, par rapport aux grilles, ce problème est allégé par le choix de O_{max} (paramètre supplémentaire sur lequel on peut jouer).

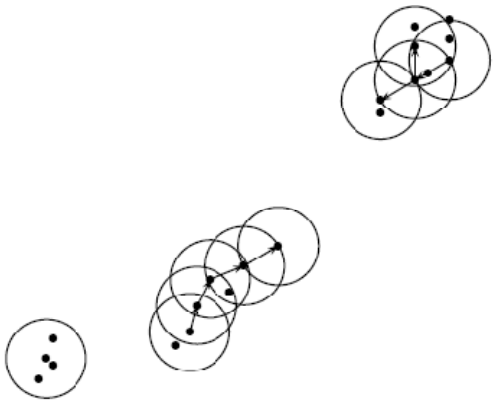


Fig.II.3: Illustration de regroupement de "voisinages denses voisins"

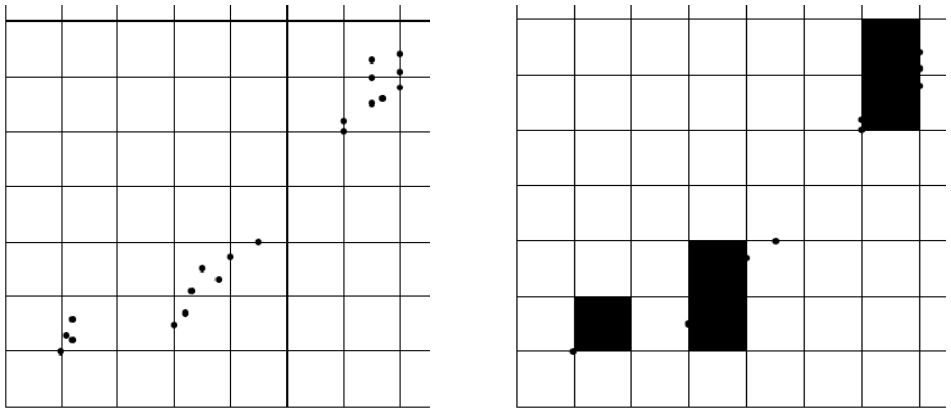
II.4.5 Clustering basé sur les grilles

Illustrée par la figure ci-dessous, la méthode de clustering basée sur les grilles utilise une grille partitionnant l'espace des données en de multiples cellules à *p* dimensions (*p* étant le nombre d'attributs). Ensuite, les densités de ces cellules bien délimitées peuvent être calculées et deux types d'approches sont possibles [26]:

1. Détection de zones denses : approche qui cherche à détecter les clusters comme des zones denses (possédant beaucoup d'objets par unité de volume) dans l'espace des données. On fusionne donc des cellules de sorte que leur regroupement ait une densité supérieure à une valeur seuil fixée et suffisamment uniformément répartie.

2. Détection de zones peu denses : approche qui vise à déceler des zones inoccupées de l'espace afin d'établir les frontières entre clusters. On se base donc sur l'existence de changements (brusques) de densités au travers des limites des clusters afin de reconstituer ceux-ci.

La principale difficulté de ces méthodes est de choisir convenablement les paramètres liés à la taille des cellules. En effet, des cellules de trop grande taille reprendront beaucoup d'objets qui ne sont pas forcément bien uniformément répartis. On pourra donc avoir des clusters peu homogènes, nécessitant d'être subdivisés par la suite. Il y'a donc sous-partitionnement. Par contre, des cellules de petite taille seront généralement très denses à partir du moment où elles contiennent des objets. On aura alors tendance à détecter des frontières qui n'auraient pas lieu d'être et à avoir un sur-partitionnement.



(a) Partitionnement de l'espace par l'utilisation d'une grille.

(b) le clustering associé.

Fig.II.4 : Clustering basé sur les grilles.

II.4.6 Clustering par la théorie des graphes

Ce type de méthodes est surtout utilisé lorsque les données sont présentées sous forme de graphe. Le principe le plus connu consiste à déterminer un arbre partiel minimum (MST) [120], et ensuite à effacer les arrêtes de plus grandes valeurs afin de déterminer les clusters. Pour cela, deux approches sont possibles : soit on fixe le nombre de clusters à trouver, soit on efface les arrêtes dont la valeur est supérieure à une valeur seuil. Ce principe est illustré à la figure II.4 où l'on obtient une partition en trois clusters. Le principal inconvénient de cette méthode est sa complexité, essentiellement liée à la construction de l'arbre partiel minimum.

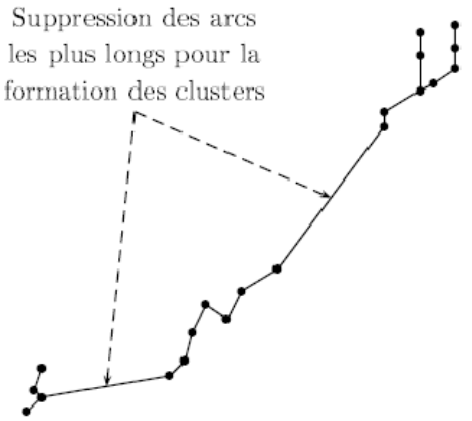


Fig. II.4: Illustration d'un clustering par MST

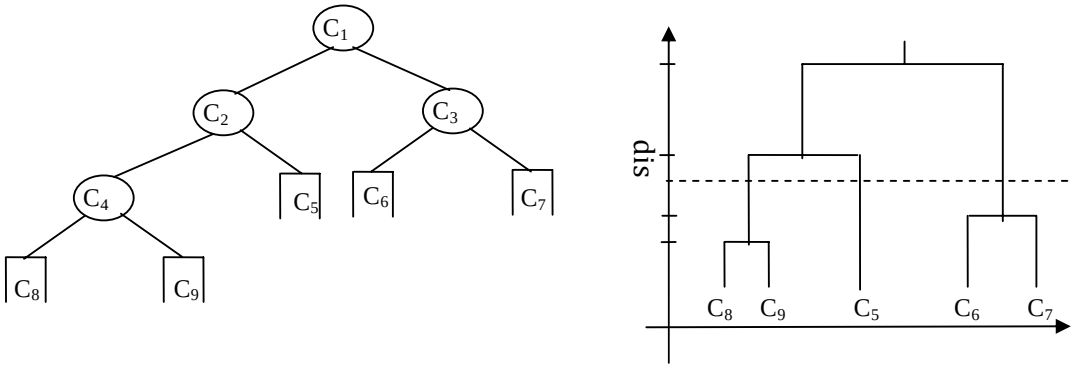
II.4.7 Clustering spectral

Le fondement du clustering spectral consiste à projeter les objets dans un sous-espace de variance maximum, en utilisant une matrice de similarité entre les objets, puis à identifier les clusters en s'appuyant sur une méthode de partitionnement choisie dans un tel sous-espace [121]. La première étape de la méthode consiste donc à calculer les valeurs propres de la matrice de similarité, ainsi que les vecteurs propres associés. Les objets sont alors projetés dans l'espace formé par une partie de ces vecteurs propres associés. Ensuite, différentes techniques peuvent être utilisées pour partitionner l'ensemble des objets projetés. On peut soit utiliser un algorithme de type K-means dans le nouvel espace, ou bien utiliser un algorithme de partition de graphe pour trouver la zone de coupure maximisant la séparation entre les objets. Cette technique de partitionnement est alors utilisée pour diviser la base des objets en deux, puis elle est itérée sur chaque sous-groupe ciblé, jusqu'à ce que le nombre de clusters recherchés par l'utilisateur soit atteint. Autrement dit, une telle méthode requiert souvent le nombre de clusters recherchés, mais plusieurs techniques ont été suggérées pour définir automatiquement ce paramètre. L'intérêt d'utiliser de telles transformations des données est de se placer à chaque étape de la méthode dans un sous-espace dans lequel les différences entre clusters sont plus importantes que dans l'espace initial. Par exemple, utiliser une telle technique avec l'algorithme K-means permet de cibler des clusters d'orientation quelconque, alors que K-means seul n'est pas capable de cibler de tels clusters.

II.5 Les méthodes hiérarchiques

En opposition aux méthodes de partition, le fondement du clustering hiérarchique est de créer une hiérarchie de clusters [80]. À la racine de l'arbre est associé un unique cluster contenant l'ensemble des objets de la base, puis plus on descend dans l'arbre, plus les clusters sont spécifiques à un certain

groupe d'objets considérés comme similaires. La sortie d'une méthode hiérarchique n'est donc pas directement une partition de l'espace des données, mais un arbre de partitions successives appelé dendrogramme. Un exemple de dendrogramme d'une partition à 5 classes (C_5, C_6, C_7, C_8, C_9) est présenté à la figure II.4. L'axe horizontal correspond aux objets tandis que l'axe vertical indique la dissimilarité entre les différents niveaux.



Arbre de décision d'une partition en 5 classes

Hiérarchie indicée d'une partition en 5 classes

Fig. II.4: Exemple de dendrogramme

Afin de former une telle hiérarchie de clusters, il existe deux méthodes principales :

- la méthode ascendante, démarrant avec autant de clusters que d'objets initiaux dans la base, puis fusionnant successivement les clusters considérés comme les plus similaires, jusqu'à ce que tous les objets soient réunis dans un unique cluster stocké à la racine de la hiérarchie formée ;
- la méthode descendante, démarrant avec un unique cluster contenant l'ensemble des objets de la base, puis divisant successivement les clusters de manière à ce que les clusters résultants soient les plus différents possible, et ce jusqu'à obtenir aux feuilles de la hiérarchie autant de clusters que d'objets dans la base.

À ce niveau, on trouve donc la différence entre les deux méthodes : une méthode ascendante démarre avec une solution tout à fait spécifique aux données, qui est ensuite généralisée à chaque étape, alors qu'une méthode descendante démarre avec une solution complètement générale, qui est ensuite spécialisée à chaque étape. Une fois la hiérarchie formée, une étape optionnelle peut être ajoutée pour affiner le résultat fourni à l'utilisateur. Il s'agit alors de déterminer le niveau de coupure le plus approprié à appliquer dans l'arbre pour un regroupement des données aussi pertinent que possible. Ceci revient à couper l'arbre au moment où l'on commence à rassembler des éléments faibles similaires (ou lorsqu'on commence à diviser des éléments homogènes). Cette

coupe est présentée en traits pointillés sur la Fig. II.4. Pour cela, on peut demander à l'utilisateur de spécifier le nombre de clusters attendus, et utiliser les différentes mesures de la qualité interne des différentes partitions possibles pour sélectionner la plus pertinente. Ou bien on peut demander à l'utilisateur de spécifier des seuils sur la cohésion interne ou l'isolation externe minimale des clusters.

II.5.1 Méthodes hiérarchiques ascendantes

Comme nous l'avons vu, les méthodes hiérarchiques ascendantes consistent à rassembler, à chaque étape, les éléments (objets ou clusters) les plus similaires au sein d'un même nouveau cluster. Leur schéma général peut être présenté comme :

1. Créer autant de clusters qu'il y a d'objets. Pour *N* objets, on aura donc *N* singletons. Définir une valeur seuil de distance (ou dissimilarité) au-dessus de laquelle deux éléments ne devront pas être rassemblés.
2. Comparer toutes les paires d'éléments possibles et marquer la paire ayant la plus petite distance (ou dissimilarité).
3. Si cette distance (ou dissimilarité) est inférieure à la valeur seuil, rassembler les deux éléments dans un même cluster et retourner au point 2. Sinon, fin de la procédure.

Il est évident que le choix de la valeur seuil est déterminant pour la sortie puisqu'elle impose finalement le niveau de coupe dans le dendrogramme. Partant du même principe général, les différentes méthodes hiérarchiques ascendantes se distinguent par la manière d'agglomérer les clusters, et plus précisément par la façon de déterminer les deux clusters les plus similaires à une étape. On parle de critère d'agrégation. Étant donnée une mesure de distance entre deux clusters, ci-dessous, nous allons présenter les différentes méthodes de clustering hiérarchique ascendantes [9]:

a) Méthode basée sur lien simple-Single link (SLINK) : Une méthode de clustering agglomératif nécessite la définition d'une distance entre deux clusters. Une possibilité est celle du lien minimum (voir annexe A). La méthode correspondante sera dite « SLINK ». Cette méthode possède plusieurs particularités :

- *effet serpent* : tendance à regrouper les clusters de proche en proche et à aboutir à des chaînes dans l'espace des données.
- *souplesse* : permet par exemple de détecter et d'isoler des clusters concentriques.
- *tendance à produire de gros clusters rapidement* : il suffit en effet que les deux objets les plus proches entre les clusters soient effectivement proches (au sens du seuil) pour rassembler les deux groupes, bien que les clusters peuvent s'étaler largement dans les autres directions de l'espace.
- *isole mal les clusters mal séparés* : deux clusters proches qui devraient être distincts sont facilement rapprochés.

– déséquilibre dans la taille des clusters.

Globalement, cette méthode produit de bons résultats pour la détection de clusters dans un jeu de données contenant des classes non isotropes (allongées), bien séparées et pouvant présenter des clusters concentriques ou en chaînes.

b) Méthode basée sur le Lien complet-Complete link (CLINK) : Un autre critère d'agrégation repose sur le lien minimum (complet) pour définir la distance (ou dissimilarité) entre deux clusters. La méthode correspondante est dite «CLINK» et a les particularités suivantes :

– tendance à produire des groupes de tailles similaires, isotropes.
– tendance à former des petits clusters compacts

Cette méthode performera donc en général assez bien sur un jeu de données où les clusters à détecter sont isotropes, bien isolés et de tailles équilibrées.

c) Méthode basée sur le Lien moyen de groupe-Group average link (GALINK) : Une autre mesure de distance (ou dissimilarité) entre deux clusters sur laquelle on peut se baser est celle du lien moyen de groupe, encore dite lien des centroïdes. La méthode correspondante est dite « GALINK » et possède des particularités intermédiaires aux deux méthodes précédentes.

d) Méthode basée sur le lien moyen-Average link (ALINK) : Le critère d'agrégation peut encore se baser sur le lien moyen. La méthode correspondante est alors dite « ALINK » et ses particularités sont :

– Tendance à des clusters plus uniformes à partir du moment où tous les objets participent à la mesure de la distance (ou dissimilarité) entre les deux clusters. chaque objet est en moyenne plus proche de son groupe que de tout autre groupe.
– Tendance à détecter des groupes isotropes.
– Les tailles des groupes peuvent être différentes.

De manière générale, cette méthode produira de bons résultats lorsque les groupes à détecter sont plus ou moins isotropes et peuvent avoir des tailles différentes.

e) Méthode de WARD : Cette méthode ne se base pas directement sur une mesure de distance (ou dissimilarité) entre clusters, mais fait reposer le critère d'agrégation sur la notion de variance. Ainsi, à chaque étape, on rassemblera la paire de groupes produisant la plus petite variance dans le groupe obtenu par agglomération de la paire. Cette méthode est coûteuse en temps de calcul (évaluation des combinaisons possibles et de leur variance), mais produit en général de bons résultats, essentiellement lorsqu'il est nécessaire de détecter des clusters sphériques.

II.5.2 Méthodes hiérarchiques descendantes

Les méthodes hiérarchiques descendantes déterminent, à chaque étape, le groupe courant le moins homogène et le divisent en deux sous-groupes. Leur schéma général est le suivant :

1. Rassembler tous les objets dans un même cluster. Définir une valeur seuil de distance (ou de dissimilarité) au-dessus de laquelle deux objets ne pourront pas être considérés comme appartenant à un même groupe.
2. Comparer tous les objets deux à deux dans chaque cluster et marquer la paire d'objets ayant la plus grande distance (ou dissimilarité).
3. Si cette distance (ou dissimilarité) est supérieure à la valeur seuil, diviser le cluster correspondant en deux et retourner au point 2. Sinon, fin de la procédure.

Contrairement aux méthodes agglomératives, seule la définition d'une distance (ou dissimilarité) entre deux objets est nécessaire pour les méthodes divisives. Les diverses méthodes se distinguent alors uniquement sur la manière de diviser un cluster en deux sous clusters.

II.6 Comparaison entre les méthodes de clustering

De plus de sa sensibilité à l'initialisation, son éventualité de convergence vers des solutions optimales locales et son exigence de spécifier le nombre des classes recherchées, la méthode de K-means ne produit que des clusters de forme hypersphérique et n'est pas capable de gérer des clusters proches dont les tailles sont très différentes. Une solution pour pallier à ce problème est de permettre de diviser ou de fusionner a posteriori les clusters obtenus. Typiquement, un cluster est divisé quand sa variance est au-dessus d'un certain seuil préspécifié, et deux clusters sont fusionnés lorsque la distance entre leurs centroïdes est en-dessous d'un autre seuil préspécifié. Une autre solution consiste à utiliser un modèle plus riche pour la représentation des clusters. Par contre, contrairement à K-means, la méthode statistique EM est capable d'identifier des clusters de forme allongée, et est capable de gérer des clusters proches dont les tailles sont très différentes. Elle est également capable de gérer le bruit qui peut exister dans les données. Cependant, elle nécessite souvent de nombreuses itérations avant de converger et sa complexité est quadratique en fonction du nombre de dimensions [27].

L'avantage des algorithmes de clustering stochastiques face aux larges bases de données est qu'ils sont capables de fournir une solution au problème à n'importe quel moment de l'exécution de l'algorithme, de telle sorte que l'utilisateur peut choisir d'interrompre la recherche quand il le souhaite [27].

Dans le cas du clustering basé sur la densité, un avantage important est le fait qu'il est capable de cibler des clusters de formes très variées ainsi que des clusters concentriques. Elle est également naturellement capable de faire face au bruit qui peut exister dans les données. Par contre, sa complexité est quadratique en fonction du nombre d'objets de la base. De plus, elle souffre du problème d'effet de chaîne. Par ailleurs, elle est sensible aux valeurs fixées pour les deux paramètres R_{max} et O_{min}. Or il est souvent très difficile à un utilisateur de spécifier les valeurs les plus appropriées pour ces paramètres. De plus, ces

paramètres étant fixés de manière globale, ils ne permettent pas de cibler des clusters dont les densités sont très différentes.

Les techniques basées sur la grille ont les mêmes forces et faiblesses que les méthodes basées sur la densité : elles souffrent de l'effet de chaîne et prennent difficilement en compte le fait que des clusters de densités différentes puissent exister, mais elles sont capables d'identifier des clusters de formes très variées ou concentriques et de gérer le bruit qui peut exister dans les données.

Dans les techniques basées sur les graphes, la limite est principalement la complexité de la méthode utilisée pour former le graphe MST initial : en utilisant une méthode classique de construction de MST, la complexité est en $O(N^2 \log N)$. Différentes techniques utilisées pour réduire cette complexité peuvent être trouvées dans [104], mais celle- ci sera de toute façon au moins quadratique dans le meilleur des cas.

Une méthode de type spectral ayant émergé relativement récemment, de nombreuses variantes ont été proposées. Puisqu'elles sont capables de transformer l'espace de description des objets, elles sont potentiellement capables d'identifier des clusters de forme quelconque. Par contre, partant d'une matrice de similarité entre toutes les paires d'objets, puis utilisant des méthodes basées sur des calculs matriciels, leur complexité est élevée.

Bien que largement utilisées, les méthodes hiérarchiques deviennent difficilement utilisables face à de larges bases de données, car sa complexité est quadratique en fonction du nombre d'objets de la base, puisque les distances entre toutes les paires d'objets possibles doivent être calculées. Par ailleurs, cette méthode ne peut jamais défaire ce qu'elle a déjà fait auparavant. Elle ne peut donc être utilisée de façon incrémentale. Enfin, elle souffre du problème de l'effet de chaîne, c'est-à-dire que des clusters proches, mais distincts peuvent être fusionnés s'il existe une chaîne d'objets qui les relie.

En fin, le tableau reporté dans l'annexe C résume les caractéristiques associes à chaque méthode de clustering décrites ultérieurement.

II.7 Clustering hybride

Le développement de certaines méthodes hybrides entre les versions hiérarchiques et les versions par partition du clustering a été motivé par l'exploitation des avantages de ces deux techniques qui semblent complémentaires : le clustering hiérarchique est plus souple que le clustering par partition, alors que ce dernier a une complexité linéaire en temps et en espace qui est inférieure. Ainsi, une méthode de combinaison proposée dans [95], [40] consiste, dans un premier temps, à appliquer la méthode des K-means, en adoptant volontairement un nombre de clusters largement supérieur au nombre de clusters souhaité dans la partition finale. Ceci nous fournit une première partition de l'espace des données. Dans un deuxième temps, on regroupe hiérarchiquement les centroïdes des clusters trouvés à l'étape 1. Ceci nous permet de ne pas devoir appliquer d'algorithme agglomératif sur l'ensemble complet du jeu de données. Le gain est conséquent vu la complexité quadratique des méthodes agglomératives. On obtient une partition du jeu de centroïdes de l'étape 1 par coupure du dendrogramme au niveau adéquat. On dispose alors d'un nouveau

nombre de clusters, plus approprié. Finalement, on consolide le résultat par réallocation des objets aux centres des clusters de l'étape 2, par la méthode des K-means.

Dans [123], une autre méthode de combinaison a été proposée. Dans ce cas, la méthode hiérarchique est combinée à une méthode basée sur les grilles. En haut de la hiérarchie, une unique cellule représente l'ensemble de l'espace, puis plus on descend dans la hiérarchie, plus la résolution de la division de l'espace par les grilles est fine.

Dans [84], le principe suivi consiste à créer un ensemble initial important de *petits* clusters en utilisant une méthode par partition du graphe des plus proches voisins, puis à les fusionner en utilisant un algorithme hiérarchique ascendant.

Il existe d'autres méthodes hybrides qui ne font pas forcément appel à une méthode hiérarchique et une méthode de partition simultanément. Un exemple combine deux méthodes de partition [78] : algorithmes génétiques et K-means. Dans ce cas, on utilise un algorithme génétique pour déterminer une bonne partition, supposée proche de l'optimum global. Ensuite, on utilise cette partition comme point de départ de la méthode des K-means qui convergera donc probablement vers la solution optimale (de manière plus rapide que le temps qu'il aurait encore fallu à l'algorithme génétique).

II.8 Conclusion

Dans ce chapitre, nous avons présenté une grande variété de méthodes de clustering. Chaque technique présente certaines caractéristiques qui la rendent applicable à des situations bien spécifiées ; les méthodes hiérarchiques deviennent difficilement utilisables face à des larges bases de données, elles seront coûteuses en CPU et elle souffre du problème de l'effet de chaîne. Alors que le clustering basé sur les graphes peut également être utilisé dans de tels cas. Face aux larges bases de données, le clustering stochastique est capables de fournir une solution au problème à n'importe quel moment de l'exécution de l'algorithme. Si au contraire des problèmes de temps d'exécution se posent, alors c'est souvent le clustering K-means qui est utilisé. Lorsque les clusters peuvent être de forme allongée, alors le clustering statistique est plus approprié que le clustering K-means. Si l'objectif est de cibler des clusters qui peuvent être de forme quelconque, alors ce sont les clustering basés sur la densité, sur les grilles ou le clustering spectral qui sont utilisés.

Les méthodes hybrides alliant une méthode de partition des k-means (pour sa complexité linéaire) avec une méthode hiérarchique (pour sa plus grande souplesse) paraît une solution pour le traitement des grandes bases de données.

Pour résoudre le problème de l'optimum local posé dans les K-means, une solution consiste à combiner deux méthodes de classification [78] : algorithmes génétiques et k-means.

Les méthodes de clustering discutées dans ce chapitre se basent sur certains nombre d'hypothèses. Dans le cas d'attributs qualitatifs, par exemple,

l'algorithme des K-means appliqué dans ce cadre repose sur la notion de médoïde au lieu de celle de centroïde, et l'algorithme correspondant est appelé «K-modes». Supposant des clusters non mutuellement exclusifs, moyennant l'introduction de degrés d'appartenance à chacun des clusters, on réalise du «fuzzy clustering». Notons qu'une partition en clusters mutuellement exclusifs peut être obtenue à partir d'une partition floue en attribuant à chaque objet le cluster pour lequel il a le plus grand degré d'appartenance. Nous verrons, dans le chapitre suivant, que l'algorithme Fuzzy C-Means (FCM) ressemble beaucoup à K-means. En effet, avec le FCM, on cherche aussi des classes homogènes, mais en construisant une suite de partitions floues.

Chapitre III
Le clustering flou

III.1 Introduction

Les méthodes de clustering peuvent être distinguées selon la façon d'affectation des données aux clusters, c'est-à-dire, suivant le type de partitions qu'elles forment. Dans les méthodes de clustering discutées dans le chapitre précédent, chaque objet doit être attribué à un seul cluster. Ces méthodes classiques produisent des partitions exhaustives de l'ensemble de données en sous-ensembles non vides et disjoints. Une telle attribution peut être inadéquate en présence de points qui sont équidistants des deux ou plusieurs clusters. Ces points de donnée peuvent représenter des formes hybrides ou un mélange d'objets qui sont aussi semblables à deux ou plusieurs formes. Une partition dure force l'attribution totale de ces points à l'un des clusters, mais ils doivent également appartenir à tous les clusters. Pour pallier à cet inconvénient, certaines techniques d'analyse de clusters «crisp» permettent de prévoir les clusters chevauchants. Ensuite, chaque objet doit être attribué à un cluster au moins, mais avec la possibilité d'assignement simultanée à d'autres clusters. Par conséquent, le clustering flou permet l'appartenance progressive des objets aux clusters dans [0,1]. Cela donne la flexibilité d'exprimer l'appartenance des objets aux plusieurs clusters en même temps. En plus de l'assignement des objets aux clusters avec différents degrés d'appartenance, le degré d'appartenance peut également exprimer le niveau d'ambiguïté ou de certitude d'appartenance d'un objet à un cluster. Le concept de ces degrés d'appartenance est prouvé par la définition et l'interprétation des ensembles flous. Un ensemble flou μ d'un ensemble X est une application [130]:

$$\mu: X \rightarrow [0,1] \qquad \text{(III. 1)}$$

Soit μ_A un ensemble flou de l'ensemble X et soit B un sous-ensemble classique de X, $B \subset X$. Alors, la valeur $\mu_A(x)$ est appelée degré d'appartenance de x à μ_A. Elle indique dans quelle mesure l'objet x satisfit certaine propriété imprécise A modélisée par μ_A (ou dans quelle mesure elle correspond à un concept vague A défini par sa fonction d'appartenance correspondante μ_A). Alors que, pour le sous-ensemble classique B, chaque objet x, soit il appartient à B $(x \epsilon B)$ ou non $x \notin B$, l'ensemble flou μ_A permet des transitions douces des appartenances des éléments

de *X*, entre $\mu_A(x) = 1$ (appartenance complète) et $\mu_A(x) = 0$ (pas membre). Une valeur de $\mu_A(x)$ proche de 0 signifie un faible degré d'appartenance, une valeur proche de 1 signifie un degré d'appartenance élevé de x à l'ensemble flou. À partir de la définition, il est évident que les ensembles flous sont en fait des fonctions. Elles peuvent être considérées comme la généralisation des fonctions caractéristiques des ensembles classiques (les fonctions caractéristiques indiquent le degré d'appartenance à l'ensemble avec 1, et 0 sinon). Toutefois, les ensembles flous permettent à tous les degrés d'appartenance pour qu'ils soient dans [0,1]. Dans le clustering flou, les degrés d'appartenance des objets aux clusters ont été fuzzifiés pour permettre une solution fine placée sous forme des partitions floues de l'ensemble des données $X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\}$. Les clusters Γ_i qui sont été des sous-ensembles classiques dans les méthodes traditionnelles, sont désormais représentés par les ensembles flous μ_{Γ_i}de l'ensemble de données *X*. En respectant la théorie des ensembles flous, l'assignement u_{ij} est désormais le degré d'appartenance d'un point $\vec{x}_j$ au cluster Γ_i, tel que $u_{ij} = \mu_{\Gamma_i}(\vec{x}_j) \in [0,1]$. Puisque les appartenances aux clusters sont floues, donc, il n'y a pas une seule étiquette indiquant à quel cluster un point de donnée lui appartient. Au lieu de cela, les méthodes de clustering floues associent un vecteur d'étiquettes floues à chaque point de données $\vec{x}_j$ qui désigne leurs appartenances aux *c* clusters.

$$\vec{u}_j = \{u_{1j}, \dots, u_{cj}\}^T \qquad \text{(III. 2)}$$

La $c \times n$ matrice $\mathbb{U} = (u_{ij}) = \{\overrightarrow{u_1}, \dots, \overrightarrow{u_n}\}$ est alors appelée la matrice de partition floue. Il existe deux types de partitions floues communément utilisées ; qui peuvent être distinguées par satisfaction de l'ensemble des contraintes par les appartenances graduelles. La section III.2 présente les deux grands types d'assignements progressives (les degrés d'appartenances flous et les degrés d'appartenances possibilistes) ainsi que des définitions et des exemples des deux types de partitions floues. La section III.3 est consacrée aux algorithmes de clustering flous. Les approches qui sont fondées sur un critère global d'optimisation des résultats de clustering sont présentées dans la section III.3.1. Ces algorithmes minimisent les fonctions objectives qui sont basées sur l'utilisation du schéma d'Optimisation en Alternance (AO) afin d'obtenir soit des modèles de clustering flous (section III.3.1.1) ou possibilistes (section III.3.1.2). Les variantes floues ou possibilistes des algorithmes basés sur la fonction objectif sont comparées dans la section III.3.2. Le cadre de l'estimation du clustering en alternance (ACE) est présenté dans la section III.3.3. La méthodologie d'ACE permet de formuler une classe d'algorithmes séparée, qui est plagiée dans AO, mais abandonne la formulation et la minimisation du critère des fonctions du clustering. Au lieu de cela, l'utilisateur est tenu de la flexibilité pour choisir les formes des ensembles flous voulus qui répondent aux exigences des applications spécifiques. L'estimation du maximum de vraisemblance floue (FMLE) fera l'objectif de la section III.3.4 ainsi que la comparaison avec son proche parent : l'algorithme de maximisation d'expectation (EM). La section III.4 conclut le chapitre indiquant les questions liées et certains développements dans le domaine.

Exemple

Ruspini [111] a été le premier chercheur proposant l'utilisation des ensembles flous dans le clustering, en 1969. Son premier exemple d'une classification floue du jeu de données "papillon" avec deux clusters est présenté dans la figureIII. 1. Les points de données dans la figure, sont associés à des vecteurs d'étiquettes, qui indiquent les degrés d'appartenance au cluster à gauche et à droite. Ruspini a noté les deux avantages suivants :

- Les points "centraux" d'une classe auront un degré d'appartenance égal à 1 pour cette classe, et les points qui lui sont périphériques pourront lui être plus ou moins fortement attribués,
- Les points appartenant à des "ponts" (faisant la jonction) entre des classes différentes pourraient être étiquetés "indéterminés", et auront des degrés d'appartenance d'autant moins élevés que leur similitude avec les points centraux des différentes classes est plus faible.

Les caractéristiques de l'ensemble de données dans la figure III. 1 aident à décrire l'expression la plus élevée des partitions floues. Avec des partitions classiques, les points équidistants au milieu de la figure devront être arbitrairement assignés avec un poids total à l'un des deux clusters .Cependant, dans cette partition floue, ils peuvent être associés avec des vecteurs$(0.5,0.5)^T$. Les partitions « Hard » ne peuvent pas, en outre, expriment la différence entre les points dans le centre et ceux qui sont plutôt à la limite d'un cluster. Les deux types de points seront attribués totalement au cluster.

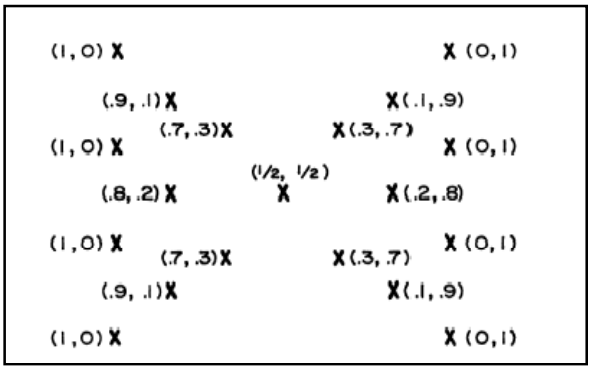


Fig. III. 1 : Le premier exemple d'un partitionnement flou de Ruspini (1969).

III. 2 Partitions floues

Les partitions floues sont destinées à fournir un moyen beaucoup plus riche pour représenter la structure de cluster. Les méthodes de clustering flou décrites dans la section suivante ont permis donc de trouver des modèles plus réalistes, puisque les frontières entre de nombreuses classes sont en fait très mal définies (c'est-à-dire, très floues). Dans le domaine de clustering flou, deux types de

partitions floues ont été évolué. Elles diffèrent par les contraintes imposées sur les degrés d'appartenance et la façon dont les valeurs d'appartenance doivent être interprétées. Nous commençons notre discussion avec le type le plus largement utilisé : les partitions floues probabilistes, car elles ont été proposés en premier. Dans la littérature, il est à noter que, les partitions floues probabilistes sont parfois appelées partitions floues (abandon du mot " probabiliste"). Dans la suite, nous utiliserons l'indice f pour les approches floues et p pour les approches possibilistes qui constituent le deuxième type de partitions floues. Les deux modèles peuvent être considérés comme des généralisations des partitions classiques (hard).

Définition 1 : partition floue Probabiliste

Soit l'ensemble des exemples $X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\}$ et soit c le nombre de clusters $(1 < c < n)$, représenté par les ensembles flous $\mu_{\Gamma_i}, (i = 1, \dots, c)$. Alors, on appelle partition floue (probabiliste) de X si :

$$\sum_{j=1}^{n} u_{ij} > 0 \quad \forall i \in \{1, \dots, c\} \tag{III. 3}$$

Et

$$\sum_{i=1}^{c} u_{ij} = 1 \quad \forall j \in \{1, \dots, n\} \tag{III. 4}$$

Nous interprétons $u_{ij} \in [0,1]$ comme le degré d'appartenance d'un objet $\vec{x}_j$ au cluster Γ_i relativement à tous les autres clusters. La contrainte (III. 3) garantit l'absence d'un cluster vide. La condition (III. 4) assure que la somme des degrés d'appartenance pour chaque donnée est égale à 1. Cela signifie que chaque donnée reçoit le même poids par rapport à toutes les autres données et, par conséquent, toutes les données sont (également) incluses dans les clusters. Par conséquent, de ces deux contraintes, aucun cluster ne peut contenir l'appartenance totale de tous les points de données.

Exemple

La classification floue de Ruspini de la base de données " papillon " est une partition floue (probabiliste). La condition (III.4) correspond à une normalisation des appartenances pour chaque point de donnée. Ainsi, les degrés d'appartenance, d'un objet, ressemblent dans leur forme, aux probabilités pour qu'il soit un membre dans le cluster correspondant. Bien qu'il soit souvent souhaitable, le caractère « relatif » des degrés d'appartenance probabilistes peut être trompeur [118]. Mais, ce n'est pas toujours le cas. Prenons, par exemple, le cas simple des deux clusters (figure III. 2). L'objet $\vec{x}_1$ a la même distance des deux clusters et, par conséquent, il est accordé avec un degré d'appartenance d'environ 0,5. Ceci est plausible. Toutefois, les mêmes degrés d'appartenances sont attribués à $\overrightarrow{x_2}$

cependant, il est encore loin des deux clusters et doit être considéré comme moins représentatif. En raison de la normalisation, la somme des appartenances doit être1. Par conséquent, $\vec{x}_2$ reçoit un degré d'appartenance assez élevé aux deux clusters. Pour une interprétation correcte de ces appartenances, il faut garder à l'esprit qu'ils sont plutôt des degrés de partage que de typicalité, car le poids constant 1 donné pour un point doit être répartir sur les clusters.

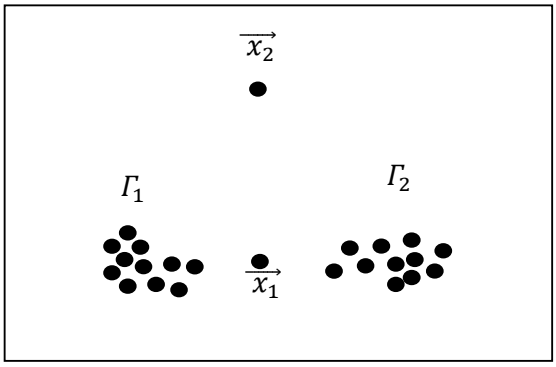


Fig. III.2 : Une situation dans laquelle l'assignement probabiliste des degrés d'appartenance est contre l'intuition pour $\overrightarrow{x_2}$

Les degrés d'appartenance probabilistes sont un peu contraires à l'intuition et ne reflètent pas la typicalité d'un point de donnée, en particulier si le point est encore loin d'un seul cluster ou même de la majorité des clusters. En réduisant la contrainte de normalisation (III. 4), la définition 1 permettrait des degrés d'appartenance plus intuitifs, qui pourrait dépendre de leur similitude (ou typicalité) à un cluster. Dans ce cas, le point $\vec{x}_2$ dans la Fig.III. 2 peut recevoir des degrés d'appartenance plus faible que les appartenances égales à 0,5, puisque cette donnée est encore loin de ces deux clusters et donc moins représentative pour chacun d'eux.

Définition 2 Partition possibiliste

Soit $X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\}$ l'ensemble des exemples et c le nombre de clusters $(1 < c < n)$, qui sont représentés par les ensembles flous μ_{Γ_i} , $(i = 1, \dots, c)$. Alors, nous appelons $\mathbb{U}_p = (u_{ij}) = \left(\mu_{\Gamma_i}(\vec{x}_j)\right)$ une partition possibiliste de X si :

$$\sum_{j=1}^{n} u_{ij} > 0 \quad \forall i \in \{1, \dots, c\} \tag{III. 5}$$

Nous interprétons $u_{ij} \epsilon [0,1]$ comme le degré de représentativité ou de typicalité d'un objet $\vec{x}_j$ au cluster Γ_i. Avec cette définition, on essaie de parvenir à une affectation plus intuitive des degrés d'appartenance en évitant les effets indésirables de normalisation discutés ci-dessus. Les degrés d'appartenance d'un objet ressemblent maintenant la possibilité (dans le sens de la possibilité théorique, [50] pour être un membre dans le cluster correspondant [90] et [34].

Exemple

La Fig.III. 3 illustre une classification probabiliste et la Fig.III. 4 montre une classification possibiliste de la base de données Iris [61]. L'échelle de gris indique l'appartenance au cluster le plus proche. Pendant que les appartenances probabilistes divisent l'espace de données, les appartenances possibilistes dépendent seulement de la typicalité au cluster respectif le plus proche. Les partitions générées sont optimales dans le sens de trouver des bonnes solutions au problème de clustering. Trouver de bonnes partitions de cluster est la question de la prochaine section.

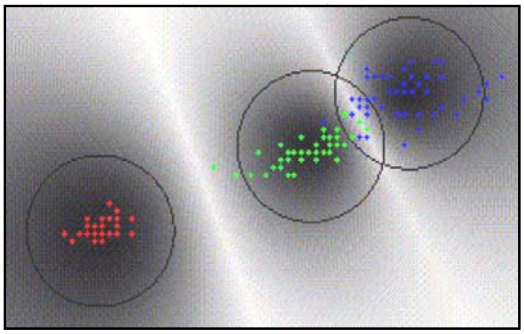


Fig.III.3: L'ensemble de données Iris classé avec l'algorithme flou probabiliste.

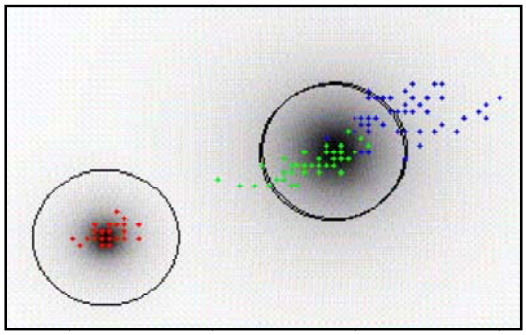


Fig.III.4: L'ensemble de données Iris classé avec l'algorithme flou possibiliste

III.3 Les algorithmes de clustering flous

Il existe une grande variété de méthodes qui ont été proposées dans une plus grande quantité d'articles et d'ouvrages. Ces dernières visent à trouver tous les clusters flous qui peuvent exister dans un certain ensemble d'exemples donné. Pour la présentation de telles méthodes, il est fructueux de se concentrer sur leurs idées sous-jacentes, puisque les méthodes peuvent être mises en différentes classes en regardant aux principes sur lesquels elles sont basées. Au début, nous regardons de plus près aux méthodes qui tentent à trouver une bonne partition floue et des prototypes de clusters en utilisant un critère global pour l'optimisation sous la forme d'une *fonction objectif*. La tâche du clustering peut alors être formulée comme un problème d'optimisation d'une *fonction objectif*. Cette dernière dépend à la fois des prototypes de clusters et des appartenances des objets aux clusters. Elle ne peut pas être directement optimisée et donc un modèle AO est généralement appliqué pour optimiser un ensemble de paramètres (par exemple, les degrés d'appartenance) en maintenant un autre ensemble fixé (par exemple, les prototypes) et vice versa. Ce schéma de mise à jour itératif est répété dans l'attente de se rapprocher de l'optimum global de la fonction critère. L'utilisateur peut alors produire une partition floue des données et des descriptions des clusters qui sont considérés comme optimaux en fonction de la fonction objectif choisie. Le schéma d'AO pour l'optimisation d'une fonction

objectif a donné lieu à la seconde classe de méthodes appelées ACE. Cette famille d'algorithmes est ensuite présentée. Elle généralise le schéma de mise à jour de l'AO et l'utilisateur peut choisir entre plusieurs équations pour la mise à jour des prototypes et des degrés d'appartenance. Par conséquence, les étapes itérées pour estimer les prototypes des clusters ainsi que les degrés d'appartenance ne reflètent pas nécessairement l'optimisation d'une fonction critère bien particulière. De cette façon, l'utilisateur aura plus de liberté quand à l'utilisation de prototypes répondant à certaines propriétés, ainsi que pour choisir les fonctions d'appartenance pour les clusters qui ont des formes mieux adaptées à une application particulière. Mathématiquement, résoudre le problème de clustering, est interprété par le fait d'avoir une flexibilité pour adapter de nouveaux algorithmes de clustering pour des besoins spécifiques.

ACE est justifié par le fait que la convergence ne constitue que rarement un problème dans les exemples pratiques (les minima locaux ou les points selles peuvent être évités).

La grande variété des approches de clustering floues est due à des modifications dans les fonctions objectives. Ces modifications sont destinées à l'amélioration des résultats en fonction des problèmes particuliers (par exemple, le bruit, les valeurs aberrantes, voir chapitre suivant). Dans cette section, nous présentons les formes non modifiées des fonctions objectives. Nous allons ainsi se concentré sur les différences entre les variantes des algorithmes possibilistes et celles des algorithmes probabilistes. Nous allons discuter l'algorithme FMLE dans la section III.3.4. Il est relatif de l'algorithme de maximisation de l'espérance (EM). FMLE est largement connu dans la communauté floue et s'allonge vers la frontière du clustering « fortement » probabiliste comme il est poursuivi dans la communauté statistique.

III.3.1 Fonctions objectives des algorithmes de clustering

Dans le clustering basé sur la fonction objectif, chaque cluster est représenté par un prototype. Ce prototype représente le centre du cluster (dont le nom indique déjà sa signification) et peut contenir des informations supplémentaires sur la taille et la forme du cluster. Le centre du cluster est une instanciation des attributs utilisés pour décrire le domaine. Toutefois, le centre du cluster est calculé à l'aide d'un algorithme de clustering et peut ou ne peut pas apparaître dans l'ensemble de données. Les paramètres tels que la taille et la forme déterminent l'extension du cluster dans des directions différentes du domaine sous-jacent.

Les degrés d'appartenance d'un objet appartenant aux différents clusters sont calculés à partir des distances (c'est-à-dire dissimilarité) de l'objet aux centres des clusters. Comme les clusters doivent être aussi homogènes que possible, le problème de partition de l'ensemble de données $X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\} \subset \Re^p$ en c clusters peut être déclaré comme l'assignement de donnée aux clusters de manière à ce que la somme des carrés des distances entre les clusters (c'est-à-dire, leurs prototypes) et les objets de chaque cluster soit minimale. C'est l'idée de base de la fonction objectif J appliquée dans le clustering flou.

III.3.1.1 Clustering flou probabiliste

La plupart des algorithmes de clustering flous probabilistes qui déterminent une partition floue (probabiliste) optimale d'un ensemble de données X en c clusters qui soumis aux contraintes (III.3) et (III.4) des degrés d'appartenance probabilistes $\mathbb{U}_f$, minimisent la fonction objectif [12]:

$$J_f(X, \mathbb{U}_f, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \, d_{ij}^2 \qquad \text{(III. 6)}$$

d_{ij} est la distance entre l'objet $\vec{x}_j$ et le cluster i. Dans le cas général, l'ensemble de clusters C comprend toutes les propriétés des clusters en indiquant l'emplacement (c'est-à-dire, le centre du cluster $\vec{c}_i$) ainsi que les paramètres de la taille et la forme pour chaque cluster. La condition (III.3) permet d'éviter la solution triviale du problème de minimisation ($u_{ij} = 0$ pour tout $i \in \{1, \dots, c\}$ et $j \in \{1, \dots, n\}$). La propriété de partitionnement de tout algorithme de clustering probabiliste, qui "distribue" le poids d'un objet aux différents clusters, est due à la contrainte de normalisation (III. 4). Le paramètre m, $m > 1$, est appelé le fuzzifieur.

Strictement, les méthodes basées sur la fonction objectif, utilisent une généralisation du « least-squared error functional» qui a été déjà connu dans l'algorithme hard c -means. Toutefois, cet algorithme et sa fonction objectif conduit à une partition hard [51]. Dunn a présenté une version floue de l'algorithme des c-moyennes afin de traiter des données appartenant à plusieurs clusters [55]. Il a montré que pour $m = 1$, les appartenances aux clusters sont Hard en minimisant J_f, même si elles ne sont pas limitées dans {0,1}. Ainsi, il a proposé une valeur du fuzzifieur égale à 2, car u_{ij}^2 conduit à la fuzzification de la partition résultante désirée. La généralisation des exposants $m > 1$ a été proposée dans Bezdek [12], mais généralement $m = 2$ est choisie. m détermine la fuzzification de la classification : avec des valeurs plus élevées de m, les frontières entre les clusters deviennent plus douces, alors que avec des valeurs plus faibles, elles sont dures.

Malheureusement, la fonction objectif J_f ne peut pas être minimisée directement. Par conséquent, un algorithme itératif est utilisé, qui optimise alternativement les degrés d'appartenance et les paramètres des clusters. En premier lieu, les degrés d'appartenance sont optimisés pour des paramètres des clusters fixés, puis les paramètres des clusters sont optimisés pour des degrés d'appartenance fixés :

$$\mathbb{U}_t = j_U(C_{t-1}) \qquad \text{(III. 7)}$$

Et

$$C_t = j_C(\mathbb{U}_t) \qquad \text{(III. 8)}$$

Les mises à jour des formules j_U et j_C sont obtenues en mettant la dérivée de la fonction objectif J_f à zéro (en tenant compte de la contrainte (III. 4)). Indépendamment de la mesure de distance choisie, la formule de mise à jour suivante pour les degrés d'appartenance est obtenue à partir de J_f [77]:

$$u_{ij} = \frac{d_{ij}^{-2/(m-1)}}{\sum_{t=1}^{c} d_{tj}^{-2/(m-1)}} \quad \text{(III. 9)}$$

Cette équation de mise à jour, montre clairement le caractère relatif du degré d'appartenance floue (probabiliste). Il ne dépend pas seulement de la distance de l'objet $\vec{x}_j$ au cluster i, mais aussi des distances entre cet objet et les autres clusters. Les formules de mise à jour j_C pour les paramètres du cluster dépendent, bien sûr, des paramètres utilisés pour décrire un cluster (emplacement, forme, taille) et sur la mesure de distance choisie. Par conséquent, une formule de mise à jour générale ne peut être donnée.

Pour les fonctions objectifs « classiques » (et leurs paramètres spécifiques de cluster ainsi que les mesures de distance) ; les équations de mise à jour sont développées plus tard. Notons que les modèles de clustering ne peuvent, bien sûr, pas seulement être optimisés en utilisant des algorithmes AO. La reformulation de la fonction critère [74] et [17] et l'optimisation avec des algorithmes génétiques [4] et [86] sont possibles.

III.3.1.2 Clustering flou Possibiliste

Dans le clustering flou possibiliste, on essaye de parvenir à une conception plus intuitive des degrés d'appartenance par la relaxation de la contrainte de normalisation (III. 4), qui permet aux appartenances d'exprimer la typicalité. Toutefois, ceci conduit à un problème mathématique lorsque la fonction objectif se minimise ($u_{ij} = 0$ pour tout $i \in \{1, \dots, c\}$ et $j \in \{1, \dots, n\}$)), c'est-à-dire, les points de données ne sont pas affectés à aucun cluster et tous les clusters sont vides. Afin d'éviter cette solution triviale, un terme de pénalité est introduit, ce qui force les degrés d'appartenance pour êtres plus loin de zéro. Ainsi, la fonction objectif J_f est modifiée comme la suite :

$$J_p(X, \mathbb{U}_p, C) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \, d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} \left(1 - u_{ij}\right)^m \quad \text{(III. 10)}$$

Où $\eta_i > 0,\ i \in \{1, \dots, c\}$ [90]. Le premier terme conduit à une minimisation des distances pondérées. Le second terme supprime la solution triviale tant que cette somme récompense des degrés d'appartenance très élevés (proche de 1) qui rapprochent l'expression $\left(1 - u_{ij}\right)^m$ de 0.

Les constantes spécifiques η_i des clusters sont utilisées pour équilibrer les objectifs opposés exprimés dans les deux termes de J_p. Il s'agit d'une valeur de référence indiquant à quelle distance à un cluster, un point de données devrait recevoir une appartenance plus élevée à celle-ci. Ces considérations décrivent la différence avec les approches de clustering probabilistes. Tandis que dans le clustering flou probabiliste, chaque point de données a un poids constant de 1, les méthodes de clustering possibilistes doivent apprendre le poids des points de

données. La formule de mise à jour des degrés d'appartenance qui est dérivée de la fonction objectif possibiliste J_p par la mise à zéro de sa dérivée est [90] :

$$u_{ij} = 1 + \left({d_{ij}^2}/{\eta_i} \right)^{1/(m-1)} \quad \text{(III. 11)}$$

Tout d'abord, cette équation de mise à jour montre clairement que l'appartenance d'un objet $\vec{x}_j$ au cluster i ne dépend que de sa distance d_{ij} à ce cluster. Une valeur petite de distance correspond à un degré d'appartenance élevé, tandis qu'une grande valeur de distance (c'est-à-dire, une forte dissimilarité) engendre un faible degré d'appartenance. Ainsi, les u_{ij} ont une interprétation dans le sens de typicalité. L'équation (III. 11) aide à expliquer les paramètres η_i des clusters. Considérant le cas $m = 2$ et remplaçant η_i dans d_{ij}^2 produit $u_{ij} = 0{,}5$. Il devient évident que η_i est un paramètre qui détermine la distance au cluster *i* dont lequel le degré d'appartenance devrait être 0,5. Puisque la valeur d'appartenance peut être considérée comme un assignement défini à un cluster, l'extension permise au cluster peut être contrôlée avec ce paramètre. Selon la forme des clusters, η_i peut avoir différentes interprétations géométriques.

Si des clusters hyper-sphériques sont envisagés, $\sqrt{\eta_i}$ signifient leurs diamètres. Si tous les clusters ont les mêmes propriétés, la même valeur peut être choisie pour tous les clusters. Toutefois, l'information sur la propriété de la forme actuelle décrit par η_i n'est pas connue à l'avance. Dans ce cas, ces paramètres doivent être estimés. Des bonnes estimations peuvent être trouvées en utilisant un modèle de clustering flou (probabiliste) de l'ensemble de données. Les η_i sont alors estimées par la distance intra-cluster flou en utilisant la matrice $\mathbb{U}_f$ comme elle a été déterminée par l'algorithme probabiliste, équivalent de l'algorithme choisi [90]. Ceci est, pour tous les clusters ($i = 1, \dots, n$):

$$\eta_i = \frac{\sum_{j=1}^{n} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m} \quad \text{(III. 12)}$$

Les équations de mise à jour pour les prototypes J_c sont aussi obtenues en mettant la dérivée de la fonction objectif J_p et les paramètres de prototype à optimiser à zéro (en maintenant les degrés d'apparence $\mathbb{U}_p$ fixés). En examinant les deux fonctions objectives, on peut déduire que les équations de mise à jour des prototypes dans l'algorithme possibiliste doivent être identiques à leurs homologues probabilistes. Cela est dû au fait que le second terme additionnel dans J_p disparaît dans la dérivée pour des appartenances fixées u_{ij}. Comme déjà mentionné, plusieurs algorithmes de clustering flou peuvent être distingués en fonction de l'information additionnelle sur la taille et la forme contenue dans les prototypes des clusters et la manière dont les distances sont déterminées. Les algorithmes les plus connus et largement utilisé sont décrits dans les paragraphes qui suivent.

III.3.1.3 Algorithmes classiques flous AO

a. *Algorithme des C-moyennes floues (FCM)*

Dans un ensemble de données $X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\} \subset \Re^p$ de dimension p, l'algorithme des FCM reconnaît les c-hyper-sphériques nuages de points. Les clusters sont supposés de nombre connu et avoir approximativement la même taille. Les prototypes des FCM sont simples. Chaque cluster est uniquement représenté par son centre $\vec{c}_i$. La distance euclidienne entre une donnée et un prototype est utilisée comme une mesure de dissimilitude d'un point de donnée à un cluster. Concernant les prototypes, la minimisation de la fonction objectif ((III. 6) ou (III. 10) conduit à [77]:

$$\vec{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \, \vec{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \qquad \text{(III. 13)}$$

La formule (III. 13) qui calcule les centres optimaux des clusters pour des appartenances données a la forme d'une valeur moyenne généralisée, d'où l'algorithme prend son nom. En tenant compte de cette équation de mise à jour pour les centres des clusters et en utilisant les Eq. (III. 9) ou (III. 11) pour les mises à jour des appartenances, ceci conduit, respectivement, au schéma spécifique AO des FCM probabilistes ou possibilistes. La forme générale du schéma AO (Alternating Optimization) d'équations réunies (III. 7) et (III. 8) commence par une mise à jour de la matrice d'appartenance à la première itération de l'algorithme ($t = 1$). Le premier calcul des appartenances est basé sur un ensemble initial de prototypes C_0. Même si l'optimisation d'une fonction objectif pourrait également et mathématiquement commencer par une matrice initiale (dans le sens de la définition 1 ou 2), une initialisation C_0 est plus facile et donc une plus pratique dans toutes les méthodes de clustering floue [45]. Principalement, les FCM peuvent être initialisé par les prototypes qui ont été disposés au hasard dans l'espace d'entrée. Il est également plus pratique d'initialiser les méthodes de clustering les plus compliquées par des prototypes initiaux qui résultent des simples algorithmes. Cette approche tient compte du fait que l'optimisation des modèles complexes est assez sensible à leur initialisation. Dans les modèles complexes, le nombre élevé de paramètres de prototype à optimiser conduit un à une grande vulnérabilité pour être bloquée dans un minimum local. Ainsi, la probabilité de produire seulement des résultats de clustering sous- optimaux peut être réduite. La mise à jour répétitive dans le schéma AO peut être arrêtée si le nombre d'itérations t dépasse un nombre prédéfini d'itérations maximum t_{max}. Toutefois, AO est habituellement arrêté lorsque les changements dans les prototypes sont inférieurs à une certaine précision de terminaison. La détection de la stabilisation des prototypes C nécessite des comparaisons plus simples que de mesurer la variation dans la matrice de partition $\mathbb{U}$.

L'algorithme *des* FCM *probabilistes* est connu comme une méthode de classification stable, fiable et rapide. Dans la pratique, il n'est pas susceptible de se bloquer dans un minimum local de sa fonction objectif. Grâce à sa simplicité,

l'algorithme des FCM probabilistes est un initialisateur largement utilisé pour les autres méthodes de clustering. Sur le plan théorique, il a été prouvé que la séquence de convergence des FCM probabilistes converge à un point selle ou à un minimum, mais non pas à un maximum de la fonction objectif [14]. D'autres preuves de convergence de techniques de clustering ont également été fournies, mais malheureusement, pas de résultat général sur la convergence pour toutes les techniques probabilistes qui sont fondées sur le modèle AO de J_f est connu [77].

b. *Algorithme de Gustafson-Kessel (GK)*

Gustafson et Kessel [72] ont proposé une variante des FCM adaptée à des ensembles dont les groupes naturels présentent des formes différentes. Dans cet algorithme de clustering flou localement adaptatif, les prototypes de clusters sont dotés d'une matrice de covariance floue en plus de ses vecteurs de centres pour la détection des clusters ellipsoïdaux $C = \{C_i \setminus C_i = \{\vec{c}_i, \sum_i \quad\}, i = 1, \dots, c\}$, [72].

La structure propre de la matrice Σ_i(définie positive de dimension $p \times p$) représente la forme du cluster i. Les tailles des clusters, si elles sont connues à l'avance, peuvent être contrôlées en utilisant les constantes $\varrho_i > 0$ exigeant que $det(\Sigma_i) = \varrho_i$. Habituellement, les clusters sont supposés avoir le même paramètre de taille ($det(\Sigma_i) = 1, \forall\, i$).

Etant donné que chaque cluster peut avoir une taille et une forme spéciale, la distance d'une donnée à un cluster particulier doit prendre en compte son extension spécifique. En conséquent, la dissimilarité est calculée en respectant les paramètres dans c_i:

$$d^2(\vec{x}_j, C_i) = \det(\Sigma_i)^{1/p} (\vec{x}_j - \vec{c}_i)^{\mathrm{T}} \Sigma_i^{-1} (\vec{x}_j - \vec{c}_i) \qquad \text{(III. 14)}$$

Selon cette mesure de distance, dans J_f ou J_p, la fonction objectif dépend aussi des paramètres C_i. Les équations de mise à jour respectives provenues de la dérivation de la fonction objectif conduit à l'équation (III. 13) pour les vecteurs des centres, avec :

$$\Sigma_i = \frac{\sum_{j=1}^{n} u_{ij} (\vec{x}_j - \vec{c}_i)(\vec{x}_j - \vec{c}_i)^{\mathrm{T}}}{\sum_{j=1}^{n} u_{ij}} \qquad \text{(III. 15)}$$

Bien sûr, l'initialisation de l'algorithme GK avec quelques itérations des FCM probabilistes est fortement recommandée. Par rapport aux FCM, l'algorithme de Gustafson et Kessel [72] demande trop de calcul à cause des inversions des matrices. Une restriction de clusters qui ont des formes à axe-parallèle réduit les coûts de calcul. Les ellipsoïdes à axes-parallèles sont généralement préférés lorsque le clustering est appliqué pour la génération des systèmes de règles floues [77].

c. *Autres modèles de prototypes de type quelconque*

Il y a une grande variété de modèles avec des prototypes plus complexes. Les a*lgorithmes de clustering solides* cherchent les clusters à « nuages semblables » tels que l'algorithme GK. Ils sont surtout utiles dans des applications d'analyse de données. Un autre domaine d'application d'algorithmes de clustering flou est la reconnaissance et l'analyse d'images. Les algorithmes de clustering *shell « shell* clustering algorithms » sont utilisés pour la segmentation et la détection des contours géométriques spéciaux dans les images tels que les frontières des cercles et des ellipses. Tous les exemples suivants sont des algorithmes basés sur des fonctions objectifs et ayant des variantes probabilistes ainsi que possibilistes (fig. III. 5, fig. III. 6, fig. III. 7, fig. III. 8, fig. III. 9 et fig. III. 10) [77]. Toutefois, la mesure de distance utilisée pour la détection des formes particulières de cluster a été modifiée. Les algorithmes des *c*-variétés floues (FCV) et des *C*-elliptotypes floues sont capables de reconnaître des lignes, des plans ou des hyperplans (fig. III. 5). Ces algorithmes peuvent également être utilisés pour la construction des modèles de données localement linéaires. L'algorithme flou adaptive des *c*-elliptotypes (AFCE) assigne des segments de lignes disjoints aux différents clusters (fig. III. 6). Le contour du cercle peut être détecté par les algorithmes des *c*-shells flous (fuzzy c-shells) et des c-shells sphériques flous (fuzzy *c*-spherical shells).Comme les objets de frontière circulaire, en 3D sont projetés sous forme d'une image plane, la reconnaissance d'ellipse peut être nécessaire. L'algorithme des c-ellipsoïdales shells flous est capable de résoudre ce problème. L'algorithme des c-quadriques shells flous (FCQS) est en outre capable de reconnaître des hyperboles, des paraboles ou des clusters linéaires. Sa flexibilité peut être observée dans les figures III. 7 et III. 8. Les techniques de clustering shell ont également été étendues aux structures non lisses telles que des rectangles et autres polygones. Voir fig. III. 9 et fig. III. 10 pour des exemples d'algorithmes shells : C-rectangulaire flou (FCRS) et c-2-rectangulaire flou (FC2RS). Pour une discussion complète de ces méthodes, le lecteur intéressé peut se référer à Höppner et al. [77] et Bezdek et al.[18].

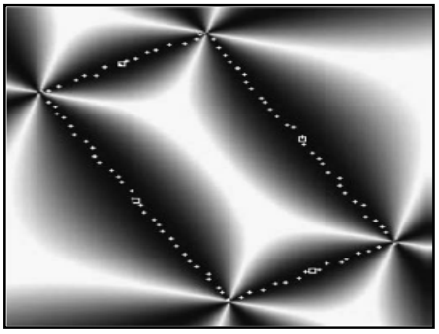


Fig.III.5: analyse FCV.

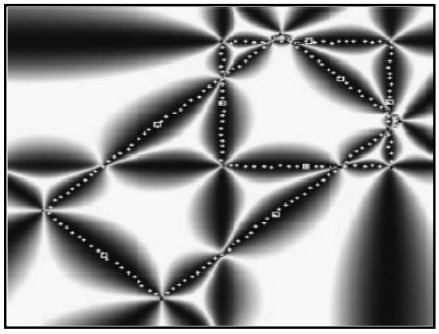


Fig. III. 6: analyse AFCE

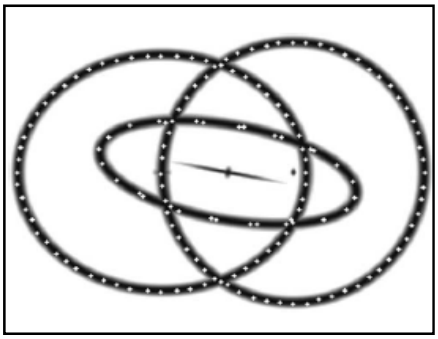


Fig.III. 7: analyse FCQS.

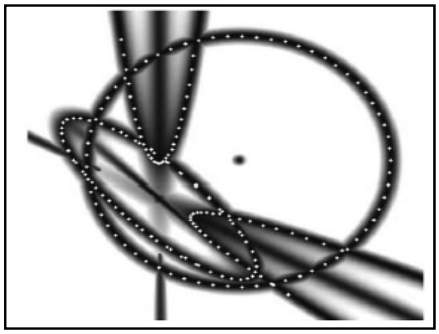


Fig.III.8: analyse FCQS.

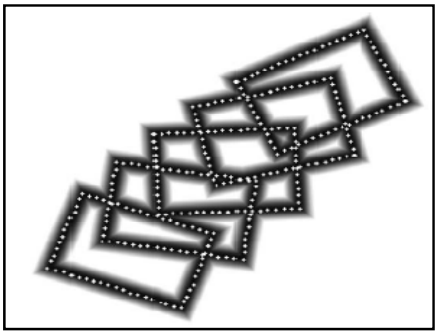


Fig.III.9 : analyse FCRS.

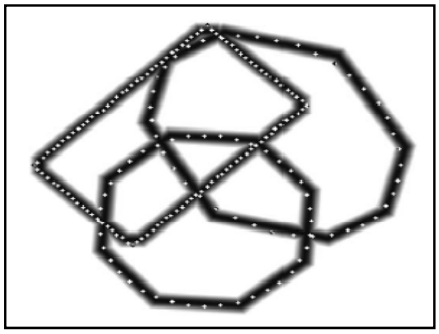


Fig.III.10 : analyse FC2RS.

III.3.2 Modèles possibiliste/probabilistes et algorithmes

En dehors de l'interprétation différente des appartenances, il y a quelques propriétés générales qui permettent de distinguer les résultats des approches de clustering floues possibilistes ou probabilistes.

Exemple

Les Fig.III.3 et III.4 illustrent une classification avec les FCM probabilistes et les FCM possibilistes des données Iris réparties en trois clusters [61]. Les partitions affichées de l'ensemble de données sont le résultat d'une optimisation en alternance de J_p et J_f [118]. Sur la gauche, l'ensemble est divisé en trois clusters. Sur la droite, l'algorithme possibiliste ne détecte que deux clusters, puisque deux des trois clusters en haut à droite de la fig.III.4 sont identiques. Notez que ce comportement est spécifique à l'approche possibiliste. En contrepartie probabiliste, les centres de clusters sont entraînés à part, car un cluster, en quelque sorte, " saisit " une partie du poids d'une donnée ce qui minimise l'attraction entre les centres de clusters. Dans l'approche possibiliste il n'y a rien qui correspond à cet effet.

Coïncidence des clusters : L'une des caractéristiques majeures dans lesquelles les approches se diffèrent réside dans le fait que les algorithmes probabilistes sont forcés de partitionner les données, tandis que les approches possibilistes ne sont pas obligés de le faire. Les premières distribuent l'appartenance totale des points sur les clusters (somme égale à 1), tandis que les seconds requièrent, plutôt, à déterminer, eux-mêmes, les poids d'un point de donnée. À cause de la contraint probabiliste, les algorithmes probabilistes tentent de couvrir tous les points de données avec les clusters. Dans le cas possibiliste, il n'y a pas de dépendance entre les clusters. Ainsi, les clusters obtenus dans les modèles possibiliste peuvent être situés beaucoup plus près les uns des autres que dans un clustering probabiliste. Les clusters peuvent même coïncider, ce qui a été largement observé [8] et [92]. Cela conduit à des solutions où un cluster effectivement être présent dans un jeu de données peut être représenté par deux clusters dans le modèle possibiliste. Le nombre c connu ou désiré des clusters a été interprété comme une bonde supérieure, car la coïncidence des clusters, en effet, conduit à un plus petit nombre de clusters dans le modèle [77]. Pour réduire la tendance de coïncidence des clusters et pour une meilleure couverture de la totalité de l'espace de données, généralement, une analyse probabiliste est effectuée préalablement (exploitation de sa propriété de partitionnement). Le résultat est utilisé pour l'initialisation des prototypes de la première exécution de l'algorithme possibiliste ainsi que pour obtenir les estimations initiales des . Après que la première analyse possibiliste a été effectuée, les valeurs des η_i sont réévaluées une autre fois à l'aide de la première partition floue possibiliste. Les estimations améliorées sont utilisées pour exécuter l'algorithme possibiliste une seconde fois en produisant les clusters finaux [77].

Répulsion de cluster : Nous avons découvert que la fonction objectif J_p est, en général, authentiquement minimisée que si tous les centres de clusters sont identiques [118]. La fonction objectif possibiliste peut être décomposée en c termes indépendants $J_{p(i)}$ pour chaque cluster i (il s'agit de la quantité de contribution de chaque cluster à la valeur de J_p). S'il y'a un point optimal unique pour chaque centre du cluster, tous les centres de clusters seront glissés dans le point résultant de la minimisation du J_p. Par conséquent, d'autres résultats différents (tous les centres de clusters ne sont pas identiques) sont obtenus seulement si l'algorithme se coince dans un minimum local d'une fonction objectif $J_{p(i)}$. Dans l'exemple du modèle possibiliste (fig.III.4), le cluster en bas à gauche dans la figure a été trouvé, car il est bien séparé et forme ainsi un minimum local de la fonction objectif. Ceci, bien sûr, n'est pas une situation souhaitable. Les bonnes solutions issues de la minimisation de J_p ne correspondent pas à ce que nous considérons comme une bonne solution du problème de clustering. Ainsi, les algorithmes possibilistes peuvent être améliorés en modifiant la fonction objectif de telle manière que les problématiques examinées ci-dessus sont supprimées. Dans Timm et al. [118], ceci a été appliqué avec un terme supplémentaire pour J_p qui ajuste les forces de répulsion de clusters. La force de répulsion des clusters peut être contrôlée avec un paramètre qui consiste à équilibrer les deux objectifs du clustering : l'homogénéité au sein des clusters contre hétérogénéité entre les

clusters. Les résultats montrent que les modifications proposées conduisent à une bonne détection des clusters chevauchants. Telles accumulations de points situés à proximité ont été une problématique, puisque les clusters possibilistes s'abusent dans la direction où la plupart des données peuvent être trouvés dans l'environnement de η_i, ce qui conduit facilement à la coïncidence des clusters. Néanmoins, les techniques possibilistes modifiés devraient aussi être initialisées avec les algorithmes probabilistes correspondants, ne laissant ainsi aucune structure d'un cluster non classifié. Les développements récents qui tentent d'atténuer les propriétés problématiques des algorithmes de clustering possibilistes, proposent d'utiliser une combinaison d'appartenance à la fois floue et possibiliste (voir section III.4.4).

Reconnaissance des positions et des formes : Les modèles possibilistes n'entraînent pas seulement des propriétés problématiques. Les appartenances qui ne dépendent que de la distance à un cluster (tout en étant totalement indépendant des autres clusters), conduisent à des prototypes qui reflètent mieux l'intuition humaine. Les centres de clusters possibilistes ainsi que leurs formes et leurs tailles (qui sont calculés à base des pondérations qui reflètent la typicalité), sont plus représentatifs par rapport à leurs contreparties probabilistes. Cela est dû aux raisons suivantes : si les clusters sont très proches ou même se chevauchent, alors, ils sont bien séparés parce que l'appartenance de partage est désavantageuse (voir en haut à droite dans la fig.III.3). Les appartenances élevées seront affectées aux points qui sont situés dans les directions à partir de ce chevauchement. Ainsi, les centres repoussent les uns des autres. Les formes des clusters sont alors susceptibles d'être légèrement déformées. Le bruit et les valeurs aberrantes sont une autre raison de distorsion légère des prototypes. Ils ont un poids dans les partitions probabilistes, ce qui découle des prototypes moins intuitifs. Contrairement, les techniques possibilistes sont moins sensibles aux valeurs aberrantes et aux bruits. Les appartenances faibles seront attribuées en raison d'une plus grande distance. Grâce à cette propriété et à la détermination la plus intuitive des positions et des formes ; les techniques possibilistes sont des outils attractifs dans les applications de traitement d'image.

III.3.3 Estimation en alternance du cluster

Les techniques ACE (Alternating Cluster Estimation) adoptent l'algorithme basé sur la fonction objectif AO, et utilisent des équations d'heuristiques pour construire les partitions et les paramètres d'estimation du cluster. Puisque les règles de génération de clusters sont choisies en utilisant des heuristiques, cette approche peut être utilisable lorsque les modèles de clusters devenus trop complexes pour les minimiser d'une manière analytique ou la fonction objectif n'est pas dérivable [77]. L'architecture générale de l'algorithme AO (éq. (III.7), (III.8)) est généralisée, mais le but de minimisation des fonctions objectifs avec J_U et J_C est abandonné. Ainsi, la tâche de classification est directement décrite par les équations de mise à jour retenue par l'analyse de données. La plus grande flexibilité du choix, parmi les différentes équations de mise à jour, est en

particulier intéressante, lorsque le clustering est appliqué pour la construction des systèmes à basés des règles floues. Dans telles applications, les ensembles flous portent un sens sémantique, par exemple, ils sont assignés des étiquettes linguistiques comme " faible ", " environ zéro ", ou " élevé ". Par conséquent, les ensembles flous, dans les contrôleurs flous par exemple, sont tenus d'être convexes, ou encore monotones [130]. En outre, ils doivent avoir un support limité, i.e. les degrés appartenances différents de zéro sont autorisés que dans un petit intervalle de leur univers. ACE offre la flexibilité de définir les algorithmes de clustering flous qui produisent les clusters Γ_i dont les ensembles flous $\mu_{\Gamma i}$correspondants remplis ces exigences. Contrairement, les clusters et les degrés d'appartenances $\mu_{\Gamma_i}(\overrightarrow{x_j}) = u_{ij}$ obtenues avec les techniques de clustering basées sur la fonction objectif n'effectuent pas les propriétés souhaitées. Le u_{ij} obtenu par l'AO peut-être interprété comme des échantillons discrets des fonctions d'appartenance continues $\mu_i: \Re^p \rightarrow [0,1]$ pour chaque cluster. Pour les algorithmes AO flous probabilistes, la fonction d'appartenance continue qui résulte de l'équation (III.9)(avec d_{ij} étant la distance euclidienne $\|.\|$) est définie par:

$$\mu_i(\vec{x}) = \frac{\|\vec{x} - \vec{c_i}\|^{-2/(m-1)}}{\sum_{t=1}^{c}\|\vec{x} - \overrightarrow{c_t}\|^{-2/(m-1)}} \qquad \text{(III. 16)}$$

La Fig.III.11 montre les fonctions d'appartenance qui résulteraient de l'algorithme des FCM probabilistes pour les deux clusters. Évidemment, les fonctions d'appartenance μ_i ne sont pas convexes ($i = \{1,2\}$). Les fonctions d'appartenance possibiliste, qui découlent d'une extension continue selon l'Eq. (III.11) sont convexes, mais elles ne sont pas limitées à des environnements locaux autour de leurs centres (i.e., les appartenances n'atteindrons jamais zéro pour les distances plus importante). Ainsi, si les ensembles flous avec un support limité (comme dans les contrôleurs flous) sont souhaités, les fonctions d'appartenance possibilistes sont aussi bien inadéquates. La transformation des fonctions d'appartenance des techniques basées sur la fonction objectif en formes souhaitées à l'application particulière est possible, mais conduit souvent à des erreurs d'approximation et aux modèles moins précis [45].

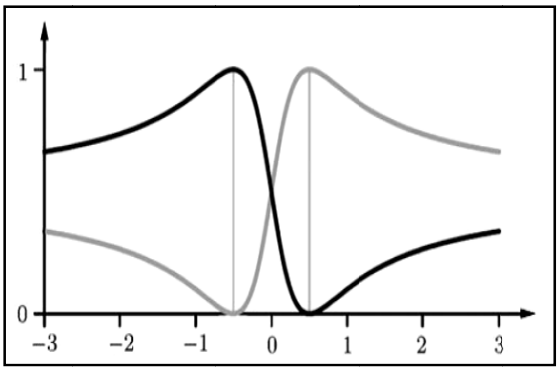


Fig.III.11 : Les fonctions d'appartenance obtenues par l'AO probabiliste pour les deux clusters à -0,5 et 0,5.

Par conséquent, ACE permet de choisir d'autres fonctions d'appartenance différentes de celles qui découlent d'une fonction objectif basée sur AO. Les propriétés désirées de la fonction d'appartenance peuvent être facilement incorporées dans l'ACE. L'utilisateur peut choisir entre les fonctions paramétrées: gaussienne, trapèze, Cauchy et triangulaire [77]. D'une manière exemplaire, nous présentons dans la Fig.III.12, la forme triangulaire, car il possède tout les propriétés souhaitées considérées ci-dessus :

$$\mu_i(\vec{x}) = \begin{cases} 1 - \left(\frac{\|\vec{x} - \vec{c}_i\|}{r_i}\right)^{\propto} & si\ \|\vec{x} - \vec{c}_i\| \le r_i, \\ 0 & ailleurs, \end{cases} \qquad \text{(III. 17)}$$

Où r_i sont les rayons des clusters, $\propto \in \Re > 0$. Dans l'algorithme ACE, en utilisant des clusters sous forme « hyper-cône » (α=1), les appartenances des données aux clusters fixés sont estimées en utilisant l'équation ci-dessus, tel que $u_{ij} = \mu_i(\vec{x}_j)$.

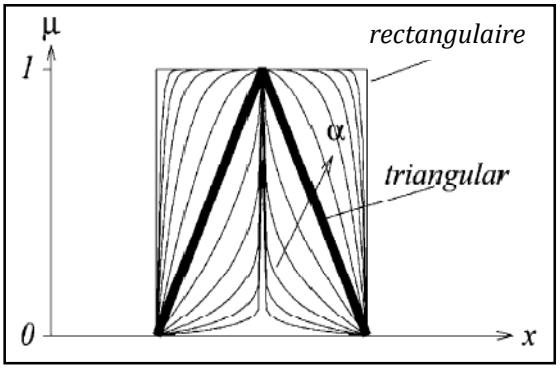


Fig. III.12 : L'ensemble flou paramétré d'une forme triangulaire.

En opposition avec le modèle AO, dans l'ACE, une grande variété d'équations paramétrées de méthodes de défuzzification est offerte pour l'estimation des centres de clusters pour des appartenances fixées. La référence aux techniques de défuzzification se pose, car un centre "crisp" est calculé à partir des points de données pondérés d'une manière floue. Également un ordre plus élevé pour les prototypes comme des lignes, des segments de droite a été proposé pour ACE [77]. Dans le cas le plus simple, lorsque les clusters sont représentés par leurs centres seulement, de nouveaux vecteurs de centres pourraient être calculés comme la moyenne pondérée des points de données de chaque cluster (comme dans les FCM, voir Eq. (III.13)).

Après le choix des équations de mise à jour pour U et C, des appartenances et des paramètres de cluster sont estimés en alternance, comme ils ont été définis. Cela conduit à une séquence $\{(U_1, C_1), (U_2, C_2), \dots\}$ qui se termine après un nombre prédéfini d'itérations $t_{\max}$ ou lorsque le C_t se stabilise. Certains cas de l'ACE pourraient être sensibles à l'initialisation des centres de clusters. Déterminant ainsi C_0 avec quelques itérations des FCM probabilistes pourraient être recommandé. Notons que tout algorithme conventionnel qui se base sur la

fonction objectif peut être représenté comme un exemple dans le cadre le plus général de l'ACE en sélectionnant leur fonction d'appartenance ainsi que leur équation de mise à jour des prototypes. Une comparaison expérimentale entre les algorithmes d'ACE qui ne reflète pas la minimisation d'une fonction objectif et l'algorithme AO classique tel que présenté ci-dessus peut être trouvé dans [77].

III.3.4 Estimation floue par maximum de vraisemblance

Afin d'adapter les FCM à des cas correspondants à des classes : hyper-ellipsoïdales, de densités différentes, ou à des distributions variées de l'ensemble de points au sein des classes, Gath et Geva [67] ont proposé une variante des FCM qui s'inspire de l'estimation par le maximum de vraisemblance [67]. FMLE est basé sur un modèle de mixture pour le processus de génération de données. Chaque cluster est caractérisé par *p*-variante distribution de probabilité, qui est décrite par une probabilité a priori du cluster et une fonction de densité de probabilité conditionnelle (cpdf).

À partir d'un modèle de mixture, Gath et Geva [67] ont drivé une mesure de distance telle que la dissimilarité entre un point de donnée et un cluster est inversement proportionnelle à la probabilité que cette donnée a été générée par ce cluster. L'idée derrière la définition de l'algorithme FMLE est multiple : Premièrement, la mesure de distance intuitivement définie est utilisée dans l'équation de mise à jour des degrés d'appartenance probabiliste (Eq. (III.9)). Deuxièmement, les équations de mise à jour des paramètres des prototypes sont les fuzzifications des estimateurs du maximum de vraisemblance pour les probabilités a priori et pour les paramètres de la cpdf. Pour une comparaison avec l'algorithme EM [39], les deux parties définissant le FMLE sont développées ultérieurement.

Dans un modèle de mixture, on suppose qu'un jeu de données X a été prélevé sur une population de clusters c. Le processus de génération des données peut être alors imaginé comme suit: dans la première étape, un cluster i, $i \in \{1,\ldots,c\}$, est choisi à titre d'exemple, ensuite un cpdf à utiliser est indiqué, puis l'exemple est échantillonné à partir de ce cpdf. Par conséquent, la probabilité d'occurrence d'un point de donnée $\vec{x}$ peut être calculée comme :

$$P_{\vec{X}}(\vec{x}; C) = \sum_{i=1}^{c} P_C(i; C_i) P_{\vec{X}/C}\,(\vec{x}/i\,; C_i) \qquad \text{(III. 18)}$$

Où C est une variable aléatoire qui décrit le cluster i (qui est choisi dans la première étape), $\vec{X}$ est un vecteur aléatoire décrivant les valeurs d'attribut du point de donnée. L'ensemble $C=\{C_1,\ldots,C_c\}$ englobe tous les paramètres du modèle avec chaque C_i, $i=1,\ldots,c$, contient les paramètres pour un seul cluster (c'est-à-dire sa probabilité a priori et les paramètres de la cpdf) [58]. En outre, il peut être supposé que la cpdf assemblant des attributs numériques est une distribution normale multi-variée, c'est-à-dire :

$$P_{\vec{X}/C}(\vec{x}/i\,;C_i) = N(\vec{x};\,\vec{c}_i,\Sigma_i) = \frac{1}{\sqrt{(2\pi)^p|\Sigma_i|}}exp\left(-\frac{1}{2}(\vec{x}-\,\vec{\mu})^T\Sigma_i^{-1}(\vec{x}-\,\vec{\mu})\right) \quad \text{(III. 19)}$$

Où $\vec{c}_i$ est le vecteur moyen et Σ_i la matrice de covariance de la distribution normale i=1,…,c. Par conséquent, les paramètres C_i du $i^{\text{ème}}$ cluster sont $C_i = \{\theta_i, \vec{c}_i, \Sigma_i\}$, où θ_i est la probabilité a priori du $i^{\text{ème}}$ cluster.

Supposons que les exemples dans un ensemble de données sont indépendants et qui sont tirés de la même distribution (i.e, les distributions de leurs vecteurs aléatoires sous-jacents $\vec{X}_J$ sont identiques), la probabilité d'occurrence de l'ensemble de données X est alors :

$$P(X\,;C) = \prod_{j=1}^{n}\sum_{i=1}^{c}\theta_i N(\vec{x}_j;\vec{c}_i,\Sigma_i)$$

Notons que la valeur de la variable aléatoire C_j(qui indique le cluster pour chaque cas d'exemple $\vec{x}_j$) est inconnue. Toutefois, étant donné le point de donnée, la probabilité a posteriori qu'un point de donnée a été échantillonné à partir de cpdf de l'$i^{\text{ème}}$ cluster peut être calculée en utilisant la règle de Bayes :

$$P_{\vec{X}\setminus C}(i\setminus\vec{x};C) = \frac{P_C(i;\theta_i)P_{\vec{X}\setminus C}(\vec{x}\setminus i;\theta_i)}{P_{\vec{X}}(\vec{x};C)} = \frac{P_C(i;\theta_i)P_{\vec{X}\setminus C}(\vec{x}\setminus i;\theta_i)}{\sum_{t=1}^{c}PC(t;\theta_t)P_{\vec{X}\setminus C}(\vec{x}\setminus t;\theta_t)} \qquad \text{(III. 20)}$$

Cette probabilité a posteriori peut être utilisée pour compléter l'ensemble de données et le cluster, en divisant chaque exemple $\vec{x}_j$ en c exemples, un pour chaque cluster, qui sont pondérés par la probabilité a posteriori $P_{C\setminus\vec{X}_j}(i\setminus\vec{x}_j;C)$. Cette idée est utilisée dans l'algorithme EM bien connu [39].

Le modèle de mixture donne les moyens pour définir la mesure de similarité, qui constitue la première partie de l'idée derrière l'algorithme FMLE. Gath et Geva [67] définissent la distance d_{ij} entre la donnée $\vec{x}_j$ et le cluster i comme l'inverse de la probabilité que la donnée $\vec{x}_j$ se produit et qu'elle a été générée par la distribution sous-jacente du cluster i. Puis, une probabilité élevée résulte dans une valeur de distance faible, alors qu'une faible probabilité que la donnée a été crée par la distribution du cluster i indique une grande distance. La mesure de distance, construite de cette façon intuitive, est la réciproque du numérateur dans l'équation (III. 20)des probabilités a posteriori. Nous obtenons :

$$d_{ij} = \frac{1}{PC_j(i;C_i)P_{\vec{X}_j\setminus C_j}(\vec{x}_j\setminus i;C_i)} = \frac{\sqrt{(2\pi)^p|\Sigma_i|}}{\theta_i exp\left(-\frac{1}{2}(\vec{x}_j-\vec{c}_i)^T\sum_t^{-1}(\vec{x}_j-\vec{c}_i)\right)} \qquad \text{(III. 21)}$$

Cette définition a une autre propriété intéressante: en insérant d_{ij} dans l'équation de mise à jour pour les degrés d'appartenance (III.9), il se produit des

degrés d'appartenance qui sont égales aux probabilités postérieures des points de données à condition que le fuzzifier m = 2 (voir l'équation (III. 20)). Seulement pour cette valeur de l'exposant de la pondération, les assignements partielles u_{ij} des points de données dans les clusters sont leurs probabilités a posteriori [18]. Dans le FMLE les équations pour la ré-estimation des paramètres des prototypes sont similaires aux estimateurs du maximum de vraisemblance pour les probabilités a priori et pour les paramètres de la cpdfs.

Différemment de l'algorithme EM, les exemples, ne sont pas pondérés par leur probabilité a posteriori $P_{C\backslash\vec{x}_j}(i\backslash\vec{x}_j; C)$. On se référant aux équations de l'algorithme GK pour la détection des clusters hyper-ellipsoïdes avec une matrice de covariance floue (voir Eq. (III. 15)), Gath et Geva [67] ont suggéré de choisir les poids u_{ij}^m pour arriver à des équations analogues. Ainsi, les équations de mise à jour du FMLE sont les estimateurs du maximum de vraisemblance dans lesquels u_{ij}^m ont été remplacés par les probabilités a posteriori. Ils ont mis comme suit :

$$\theta_i = \frac{1}{n}\sum_{j=1}^{n} u_{ij}^m \qquad \text{(III. 22)}$$

$$\vec{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \vec{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \qquad \text{(III. 23)}$$

$$\Sigma_i = \frac{\sum_{j=1}^{n} u_{ij}^m (\vec{x}_j - \vec{c}_i)(\vec{x}_j - \vec{c}_i)^T}{\sum_{j=1}^{n} u_{ij}^m} \qquad \text{(III. 24)}$$

Comparaison entre FMLE et AM

Le FMLE est très semblable à l'algorithme célèbre EM appliqué à la décomposition de la mixture sous l'hypothèse d'une variable cachée, indiquant l'appartenance à la classe [39]. L'ascendance des méthodes est intimement liée à la dérivation presque identique des estimateurs. Les différences, cependant, sont explicites dans le choix des pondérations d'un exemple dans les deux algorithmes.

L'algorithme EM maximise la probabilité de l'ensemble de données de se produire, d'abord en calculant les probabilités a posteriori qu'un point de donnée a été crée par la cpdfs des clusters. Ceci est fait pour les paramètres fixés du modèle et produit des poids partiels avec lesquels un point de donnée appartient aux clusters (voir l'équation(III. 20)). Ensuite, ces appartenances pondérées aux clusters sont utilisées comme coefficients de pondération dans les estimateurs afin d'optimiser la vraisemblance. Ainsi, dans l'algorithme EM, les assignements partiels d'une donnée aux clusters sont identiques aux poids d'un exemple qui est utilisé lors de l'estimation des paramètres.

D'autre part, dans FMLE, les assignements partiels, à savoir les degrés d'appartenance u_{ij}, sont différents des poids des exemples qui sont utilisés lors de la ré-estimation des paramètres, qui sont u_{ij}^{m}. Il n'y a pas le choix du m tel que l'algorithme FMLE devient identique à l'algorithme EM, car m apparaît également dans la formule (III.9) pour le calcul des degrés d'appartenance, ce qui exclut le choix de m = 1. C'est le couplage des exposants m et $2 / (m - 1)$ qui distingue FMLE de l'EM.

III.4 Questions connexes et recherches actuelles

Le clustering flou est une stratégie d'apprentissage non supervisé pour classifier les données. Il est très utile pour la construction des règles floues si-alors à partir des données et il est également appliqué en traitement d'image. Dans ce chapitre, nous nous sommes concentrés sur la fonction objectif sur laquelle elles se basent quelques approches de clustering floues : les FCM probabilistes, Les FCM Possibilistes, le FMLE. Nous avons essayé de donner une vue d'ensemble globale du champ en indiquant les grandes orientations.

III.4.1 Le clustering des données non-vectorielles

Toutes les méthodes décrites dans ce chapitre supposent que les attributs ont une représentation vectorielle. Dans certaines applications, une telle représentation n'est pas donnée, mais un rapport de dissemblance entre les objets peut être défini. Il existe une grande variété de techniques de clustering floues pour tels paramètres [19]. Ces algorithmes de clustering relationnels peuvent être divisés en approches hiérarchiques et de partitionne. Ces dernières sont essentiellement basées sur les idées qui ont été présentées ici, c'est à dire, elles sont drivées d'une fonction d'objectif. Aussi ACE a été adapté pour le clustering relationnel [109]. L'Estimation Relationnelle en Alternance des Clusters (RACE) a été appliquée avec succès dans le domaine du « Web mining » [110]. L'approche floue pour le clustering est une mesure appropriée pour traiter les données vagues ou imprécises comme observées dans certaines applications. Le clustering flous des données à temps variables peut être trouvé dans [31] et [32].

III.4.2 Manipulation des bruits et des valeurs aberrantes

Le bruit et les valeurs aberrantes sont un problème commun et qui doit être traité au sein des clusters. Toutefois, les méthodes décrites ne sont pas également troublées par ces phénomènes. Les algorithmes de clustering possibiliste basés sur AO peuvent gérer des points bruités et des valeurs aberrantes, tout naturellement, en les attribuant les faibles poids. Ainsi, la détection des vrais clusters n'est pas perturbée. La même chose pour les algorithmes ACE avec fonctions d'appartenance choisies d'une manière appropriée. Les valeurs aberrantes sont situées à une distance élevée de la partie majeure des données. Ils ne peuvent pas influencer la détection des clusters tant que la fonction d'appartenance choisie attribue un poids faible aux points éloignés. Alors, les valeurs aberrantes ont une

influence réduite ou même pas sur les centres des clusters estimés. Contrairement au comportement des approches possibilistes et approches ACE, les points aberrants affectent fortement les méthodes de clustering probabilistes à cause de la contrainte imposée sur leurs degrés d'appartenance. En dehors de la sensibilité de l'approche probabiliste au bruit, la normalisation des degrés d'appartenance (Eq. (III.4)) en combinaison avec le calcul de l'inverse du carré des distances dans l'Eq. (III.9) conduit aux appartenances élevées des points qui sont plus loin de la majorité des données [44]. Cette dernière tendance est déjà visible dans l'exemple des deux clusters (voir Fig. III.11), où les appartenances (à peu près égales) aux deux clusters sont données aux points à grandes distances aux deux clusters (dans le sens gauche et droit). Puisque la forte déviation des points donne une appartenance considérable, leur poids influence le calcul des prototypes. Il en résulte des déformations dans la forme du cluster et les centres sont légèrement décalés. Le chapitre qui suit donne un aperçu approfondi sur ce problème.

III.4.3 Validité et problème du nombre inconnu des clusters

Toutes les techniques de clustering décrites, nécessitent de spécifier le nombre de clusters *c* à rechercher. Habituellement, ce nombre n'est pas connu à l'avance. À partir des estimations initiales, plusieurs partitions de clusters peuvent être déterminées pour différents nombres de clusters. Mais, là, une question se pose, laquelle des partitions est mieux ou correcte. En outre, différents algorithmes de clustering peuvent être utilisés qui assument certaines formes et tailles des clusters. Mais ces hypothèses sont-elles satisfaisantes pour l'ensemble de données spécifié ? Ces problèmes sont traités au moyen de mesures de validité. Une grande variété d'indices de la validité ont été proposés dans la littérature [18] et [77]. *Globalement*, les fonctions de validité évaluent la partition complète et aident à déterminer le nombre correct de clusters. Puis, le clustering est effectué avec des valeurs différentes pour *c*. Par comparaison des valeurs de la fonction de validité pour les partitions produites, le nombre le plus approprié des clusters peuvent être choisi. Une autre stratégie est de commencer avec une limite supérieure pour *c* et d'utiliser des fonctions de validité *locales* qui permettent d'évaluer la bonté des clusters individuels. Les clusters sont comparés les uns aux autres tels que les clusters similaires peuvent être fusionnés à un nouveau cluster, alors que les clusters défectueux peuvent être éliminés. Ensuite, une analyse de cluster est effectuée à nouveau avec un nombre réduit de clusters. La procédure est répétée jusqu'à ce que les clusters doivent être fusionnés ou supprimés. Pour une discussion détaillée sur les fonctions de validité et les propriétés spécifiques des résultats de clustering qu'ils sont adaptées aux besoins, le lecteur peut être référé à [18] et [77].

III.4.4 Certains thèmes de recherche actuels

Régularisation des Formes et tailles : les algorithmes de clustering flou les plus sophistiqués, comme l'algorithme GK, l'algorithme FMLE, mais également l'algorithme EM sont capables d'induire des clusters de formes ellipsoïdales et de tailles différentes. Toutefois, les paramètres de cluster supplémentaires

(additionnels) peuvent réduire la robustesse des algorithmes, et rendre, parfois, leur application difficile. Ainsi, les algorithmes les plus complexes sont généralement initialisés avec des modèles plus simples. Dernièrement, des méthodes ont été proposées qui introduisent des contraintes sur la taille et sur la forme pour gérer le grand degré de liberté dans ces algorithmes de manière efficace. Les expériences montrent que ces méthodes de régularisation améliorent la robustesse des algorithmes de clustering flou les plus sophistiqués, et sans eux, les algorithmes souffrent d'instabilité, même sur des ensembles de données assez simples. Le clustering régularisé et avec contraint est robuste ; il peut être utilisé sans initialisation par l'algorithme des FCM. Avec un paramètre de régularisation de forme qui dépend du temps, on peut, encore, obtenir une transition douce de l'algorithme des FCM (clusters sphériques) à l'algorithme GK (général clusters ellipsoïdale) [21].

Comprendre les principes derrière le fuzzifieur : Le fuzzifieur *m* est un paramètre qui caractérise toutes les méthodes de clustering floues. Cette fuzzifieur contrôle à quel niveau les clusters peuvent se chevauchent ? Cependant, il y'a d'autres fonctions des degrés d'appartenance que celles de u_{ij}^{m} qui résultent des partitions floues. Des investigations récentes offrent un cadre plus général de mise en œuvre du fuzzifieur. Cela permet de surmonter certains effets négatifs et problèmes dus au : cluster avec une densité variable de données, données bruitées, et grandes bases de données avec un nombre de clusters important [87], [88].

Chapitre IV

Une nouvelle approche de clustering : résolution du problème bruit-coïncidence

IV.1 Introduction

Le but principal de l'approche possibiliste est de trouver la solution aux problèmes associés à la contrainte imposée sur les degrés d'appartenances [12] qui sont utilisés dans les algorithmes de classification probabiliste tels que les c-moyennes floues (FCM); la contrainte génère dans les FCM des degrés d'appartenance qui peuvent être interprétés comme des degrés de partage mais, pas comme des degrés de typicalité. Ainsi, dans un cluster donné constitué d'un ensemble de points, si deux points sont équidistants du prototype, alors leurs degrés d'appartenance peuvent être sensiblement différents; par contre s'ils sont répartis autrement, leurs degrés d'appartenance peuvent être égaux. Ces deux cas affaiblissent les performances des FCM en présence de bruit. L'introduction du terme η [90] dans la fonction objectif des FCM, permet d'établir un nouvel algorithme de classification appelé PCM (possibilistic C-Means). Dans cette approche possibiliste, l'appartenance d'un point à un cluster représente la typicalité ou la possibilité de ce point d'appartenir à ce dernier. La relaxation de la contrainte imposée sur les degrés d'appartenance peut engendrer des appartenances qui représentent la typicalité des objets aux différents clusters, ce qui permet de réduire l'effet de bruit et d'améliorer les résultats du classifieur. En revanche, l'approche possibiliste génère des centres de gravité identiques [7]. Compte tenu des modifications introduites dans PCM, nous présentons une étude comparative entre les différents algorithmes de classification par les c-moyennes floues, ensuite, nous proposons une nouvelle approche [23] qui est basée sur la fusion des algorithmes probabiliste et possibiliste.

IV.2 Les c-moyennes floues: Fuzzy C-Means (FCM)

Le critère de minimisation de l'algorithme des c-moyennes floues de Bezdek [12] qui minimise la fonction des moindres carrés pondérée par le degré

d'appartenance est donné par :

$$J_f(X, \mathbb{U}_f, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \, d_{ij}^2 \qquad \text{(IV. 1)}$$

Où:

- $\mathbb{U}_f$ est une partition probabiliste de l'ensemble des objets dans X;
- $m \in]1, \infty[$: la valeur qui caractérise le flou dans la partition.
- $d_{ij}^2 = \|\vec{x}_j - \vec{c}_i\|$: La distance euclidienne entre l'objet $\vec{x}_j$ et le prototype $\vec{c}_i$

La mise à jour des prototypes et des degrés d'appartenances est donnée par:

$$\vec{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \, \vec{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \qquad \text{(IV. 2)}$$

et

$$u_{ij} = \frac{d_{ij}^{-2/(m-1)}}{\sum_{t=1}^{c} d_{tj}^{-2/(m-1)}} \qquad \text{(IV. 3)}$$

Algorithme des FCM

$X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\} \subset \Re^p$ est l'ensemble des données à partitionner.

1. *Initialisation*
   - Fixer le nombre de clusters $c \in]1, n]$, le fuzzifieur $m \in]1, \infty[$, la distance d , le nombre d'itérations $l = 0$, $\varepsilon > 0$;
   - Initialiser les C-partitions floues $\mathbb{U}_{f(0)}$;
2. *Adaptation des prototypes*
   - Evaluer les C-prototypes flous $\{c_{i(l)}\}$ à l'aide de l'équation (IV.2) ;
3. *Evaluation des clusters*
   - Calculer les degrés d'appartenance flous u_{ij} à l'aide de l'équation (IV.3);
4. *Test d'arrêt*
   - Comparer $u_{ij(l-1)}$ et $u_{ij(l)}$ si $\|u_{ij(l)} - u_{ij(l-1)}\| < \varepsilon$ alors **STOP**
     Si non $l \leftarrow l+1$; recommencer à l'étape é*valuation des clusters*
     Fin si

IV.3 Avantages des c-moyennes floues

Le principal avantage de l'algorithme des c-moyennes floues par rapport aux algorithmes plus classiques (par exemple hard c-means) provient de l'introduction des degrés d'appartenance. Grâce à eux, le processus d'optimisation itératif est rendu beaucoup plus robuste, notamment en permettant de prendre en compte les recouvrements entre les clusters. Il permet ainsi d'obtenir des partitions plus pertinentes et plus proches de la réalité. En outre, ces degrés permettent de

prendre des décisions nuancées pour l'assignation d'une forme à un cluster, ce qui s'avère très intéressant pour toute forme de clustering. Parmi les autres avantages de l'algorithme, on peut noter que sa complexité algorithmique est relativement réduite par rapport à d'autres algorithmes de clustering. Cela le rend plus facilement exploitable pour traiter des problèmes de taille importante (avec beaucoup de données). Outre, ces avantages très généraux, l'algorithme des FCM est aussi particulièrement adapté au cadre de notre étude. Il permet en effet d'extraire automatiquement des sous-ensembles flous décrivant d'une façon robuste et synthétique une structure des formes incomplètement définie. Ces sous-ensembles flous peuvent être définis à partir de l'équation (IV.3) donnant directement la fonction d'appartenance. Ils ne dépendent alors que de la connaissance des centroïdes *C,* du degré de flou *m* et de la métrique *d.* Les paramètres nécessaires à leur représentation sont donc peu nombreux, ce qui les rend peu coûteux à utiliser. Malgré tout, l'algorithme possède aussi quelques inconvénients. On peut citer par exemple le problème de la sensibilité à l'initialisation (différentes initialisations peuvent aboutir à différentes partitions), la nécessité d'imposer le nombre de clusters *c* a priori, ou encore le manque de flexibilité sur la forme des clusters qu'il peut détecter. La plupart de ces difficultés peuvent cependant être contournées.

IV.4 Inconvénients des c-moyennes floues

Dans les FCM, la contrainte probabiliste imposée sur les degrés d'appartenance ; fait que les prototypes et les clusters sont établis les uns par rapport aux autres. La conséquence directe est la forme particulière des fonctions d'appartenance déduites de l'algorithme. Elles sont définies relativement les unes par rapport aux autres (cf. équation (IV.3)) et traduisent la notion de partage des objets entre les clusters. La Fig. (IV.1) montre l'impact sur le comportement des fonctions d'appartenance lorsque celles-ci sont déterminées en une dimension ou bien en deux dimensions (matérialisées par leurs lignes de niveaux) (Fig. (IV.2)).

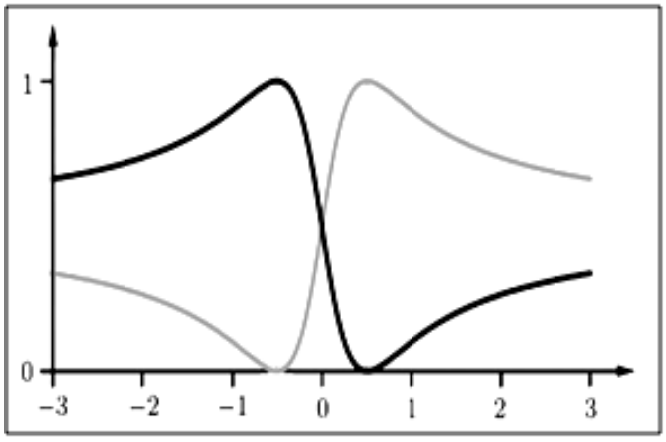


Fig. IV.1 : Forme et comportement des fonctions d'appartenance issues des FCM en une dimension.

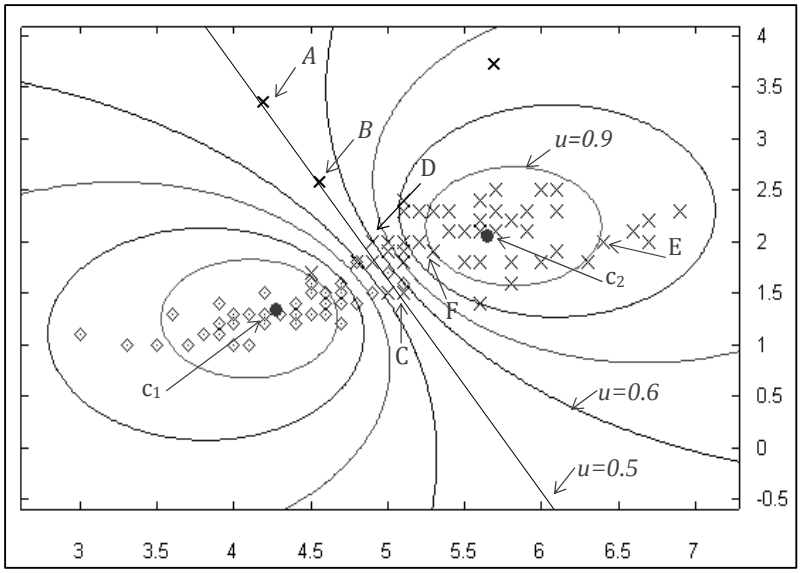


Fig. IV.2: Forme et comportement des fonctions d'appartenance issues des FCM lorsque deux clusters chevauchants sont recherchés en deux dimensions après projection des lignes de niveaux ($u = 0,9\ à\ u\ = 0,5$).

Les deux observations notées $B\ et\ C$ sont toutes les deux à une distance égale des deux centroïdes ($(d(B,c_1) = d(B,c_2)$ et $d(C,c_1) = d(C,c_2))$ .Ce qui leur permet d'avoir le même degré d'appartenance à chacun d'eux ($u_{B1} = u_{B2} = u_{C1} = u_{C2} = 0.5$), malgré que, l'objet C est un bon membre et le point B est un pauvre membre dans les deux clusters : ($d(B,c_i) > d(C,c_i)$). Même si cette particularité tend à s'estomper quand le nombre de clusters recherchés c augmente puisque les degrés tendent vers $1/c$, celle-ci est particulièrement gênante et ne correspond pas du tout à une caractérisation réelle des données. En effet, ce que l'on désire, c'est que les objets éloignés d'un centroïde possèdent des degrés d'appartenance à celui-ci qui soient faibles. C'est la notion de typicalité telle qu'elle est utilisée dans [90] et qui s'oppose à celle de partage des FCM.

Les deux observations A, B sont éloignées des deux prototypes et elles sont concédées comme objets bruités. Intuitivement, le point A ne doit pas avoir un degré d'appartenance élevé (dans le sens de typicalité) dans chaque cluster et le point B doit avoir un degré d'appartenance plus petit que celui de A dans chaque cluster, du fait qu'il est très éloigné des deux clusters. Cependant, les FCM assignent aux deux points A et B le même degré d'appartenance égal à 0.5 aux deux clusters. Dans cet exemple, les degrés d'appartenance ne représentent pas la compatibilité, mais aussi ils ne peuvent pas distinguer entre un membre atypique modéré et un membre atypique extremum. Cette situation est apparue à cause de la contrainte probabiliste qui ne différencie pas entre l'égalité de l'évidence et ignorance. Pour surmonter cette difficulté, on fait recours à des théories récentes telles que la théorie de croyance [114] ou la théorie des possibilités [50], [89] qui sont capables d'apporter quelques améliorations.

Les FCM produisent des degrés d'appartenance différents pour les observations D et E dans le cluster 2 ($u_{2D} = 0.6 < u_{2E} = 0.9$), alors qu'ils ont la même typicalité

dans ce dernier (i.e., ils sont équidistants du prototype). Ce problème est apparu à cause de la contrainte probabiliste qui partage l'appartenance du point D entre les deux clusters. Similairement, les objets E et F ont le même degré d'appartenance dans le cluster 2, alors que l'objet F est plus typique que E, ce qui signifie que le degré d'appartenance flou d'un objet à un cluster est un degré relatif, c'est à dire il dépend des autres appartenances aux différents clusters. L'appartenance du D reflète le partage entre les 2 clusters.

La mauvaise conséquence de cette notion de partage est que l'algorithme reste assez sensible aux bruits. Comme une donnée appartient nécessairement à un ou plusieurs clusters, la présence de données non représentatives influe sur la position et la forme des sous-ensembles flous. La figure IV.3 illustre ce point: la donnée notée A induit un « décalage » des centroïdes $c_1$et c_2 qui ne représentent plus la valeur la plus typique des données associées à chacun des clusters.

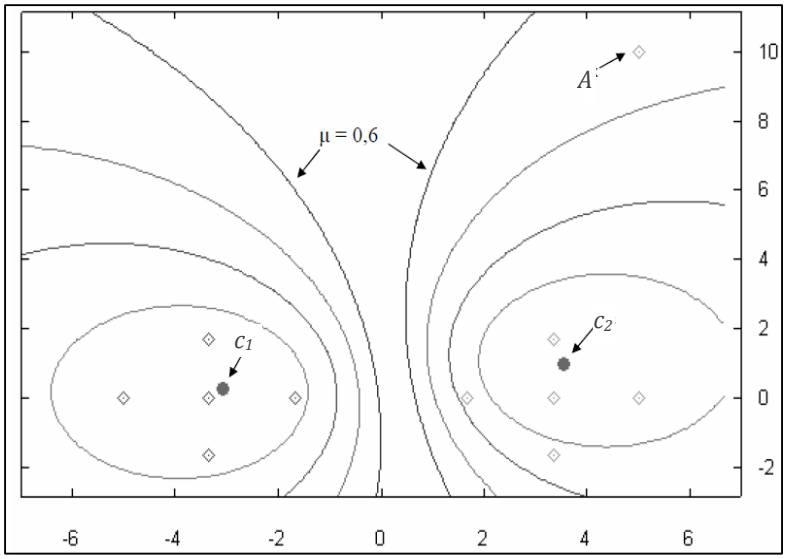


Fig. IV.3: Problèmes liés à la définition relative des fonctions d'appartenance dans les FCM: la donnée 'A' est du bruit mais elle déforme et déplace les sous-ensembles flous.

Pour toutes ces raisons, l'algorithme des FCM est inapte pour l'extraction de connaissances intrinsèques aux clusters. Même s'il peut être utilisé pour extraire une structure de sous-clusters, il est incapable de déterminer et de représenter les valeurs les plus typiques.

IV.5 Quelques variantes des FCM pour le traitement du bruit

Le bruit dû aux erreurs de nature statistique des appareils de mesure n'est pas en général source de problèmes. En revanche, les points arbitrairement présents dans les données et ne faisant partie d'aucun cluster biaisent lourdement les algorithmes de regroupement. La recherche et l'élimination de tels points avant l'application des algorithmes de regroupement peut être une solution. Jolion et Rosenfeld [81] ont, par exemple, proposé d'attribuer, à

chaque point, un coefficient, ou poids, proportionnel à la densité du nuage au voisinage de ce point. Les éléments appartenant à des groupes "naturels" se verront, alors, attribuer des poids plus élevés que ceux attribués à des points atypiques. Ce prétraitement peut, dans certains cas, réduire la nuisance due au bruit. Néanmoins, pour des distributions de bruit fortement non uniformes, cette tâche est souvent difficile à accomplir, voire impossible [35]. D'où l'intérêt de disposer d'algorithmes susceptibles de discriminer les bonnes données en présence du bruit.

Les approches que nous considérons ici visent à la manipulation des données bruitées avec des modèles de clustering flou (probabiliste). Leur objective est de définir des variantes robustes d'algorithmes de clustering flou, c'est à dire, des algorithmes dont les résultats ne dépendent pas de la présence ou l'absence de points bruités ou des valeurs aberrantes dans l'ensemble de données.

Deux approches sont mentionnées ici : la première est basée sur l'introduction d'un cluster amorphe bruité pour représenter les points bruités et la deuxième est basée sur l'utilisation d'estimateurs robustes.

IV.5.1 Clustering du bruit (NC)

L'algorithme NC qui a été initialement proposé par Davé [35] a été, ensuite, étendu par : Davé et Sen [36], [37]. Il consiste à ajouter, à côté des c clusters recherchés dans un ensemble de données, ce qu'on appelle le cluster bruité ; ce dernier vise à grouper les points qui sont mal représentés dans des clusters habituels, comme les points de données bruités ou aberrantes. Ils ne sont pas explicitement associés à un prototype, mais directement en fonction de la distance entre un prototype implicite et les points de données : le centre du cluster de bruit est considéré comme étant à une distance constante δ de tous les points de données. Cela signifie que tous les points ont a priori la même « probabilité » d'appartenir au cluster de bruit. Au cours du processus d'optimisation, cette « probabilité » est ensuite adaptée comme une fonction de probabilité selon laquelle les points appartiennent à des clusters habituels. Le cluster de bruit est alors introduit dans la fonction objectif, comme tout autre cluster, conduisant à :

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} + \sum_{k=1}^{n} \delta^{2}\left(1 - \sum_{i=1}^{c} u_{ik}\right)^{m} \qquad \text{(IV. 4)}$$

Le terme additionné est similaire à des termes dans la première somme : la distance au prototype de cluster est remplacée par δ et le degré d'appartenance à ce cluster est défini comme le complément à 1 de la somme de tous les degrés d'appartenance aux clusters standards. Cela implique, en particulier, que les valeurs aberrantes peuvent avoir des faibles degrés d'appartenance aux clusters standards, et un degré élevé au cluster bruité, ce qui permet de réduire leur influence sur le cluster standard : l'approche du clustering de bruit assouplit la contrainte de normalisation, selon laquelle la somme des degrés d'appartenance aux bons clusters doit égale à 1.

La fonction objectif (IV. 4) demande la fixation du paramètre δ. Dans l'algorithme initial NC, il a été fixé à :

$$\delta^2 = \lambda \frac{1}{c.n} \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \qquad \text{(IV. 5)}$$

C'est à dire sa valeur au carré est une proportion de la moyenne des carrés des distances entre les points et les autres prototypes des clusters, avec λ est un paramètre défini par l'utilisateur: plus λ est faible, plus la proportion des points qui sont considérés comme aberrantes est élevée. Dans le modèle de clustering de bruit, les points aberrants sont identifiés par leurs appartenances élevées au cluster bruité. Le poids restant du point de donnée qui peut être distribué aux autres clusters est donc réduit, ce qui conduit à une meilleure détection de la forme et de la position de bons clusters.

IV.5.2 Estimateurs robustes

Une autre approche permettant de gérer les ensembles de données bruités est basée sur l'exploitation des estimateurs robustes: comme nous avons indiqué dans le chapitre précédent, l'approche probabiliste des FCM est basée sur le moindre carré de la fonction objectif. Il est bien connu que l'approche du moindre carré est très sensible aux points aberrants, et c'est pourquoi le clustering probabiliste basé sur la fonction objectif donne des résultats insatisfaisants lorsqu'il est appliqué aux ensembles de données contaminés par le bruit et les valeurs aberrantes. Par conséquent, Frigui et Krishnapuram [66] ont proposé d'introduire un estimateur robuste des fonctions objectives traditionnelles (voir Eq. (IV.6)), conduisant à envisager :

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \rho_i d_{ij} \qquad \text{(IV. 6)}$$

Où ρ_i sont des fonctions robustes symétriques positives qui admettent un minimum en 0. Dans le cadre du M-estimateur robuste, ρ doit être choisie telle que $\rho(z) = log(J(z)^{-1})$ représente la contribution de l'erreur z dans la fonction objectif et J la distribution de ces erreurs. Le choix du $\rho(z) = z^2$, comme c'est habituellement le cas, revient à supposer une distribution normale des erreurs z et conduit à des fonctions de pondération constantes. Autrement dit, les grandes erreurs ont le même poids que les petites erreurs, et jouent ainsi un rôle trop important dans la correction appliquée aux paramètres, ce qui rend la fonction objectif sensible aux valeurs aberrantes. Par conséquent, il est proposé d'utiliser d'autres ρ, dont les fonctions de pondération tendent vers 0 pour les grandes valeurs de z. Frigui et Krishnapuram [66] désignent leur propre estimateur robuste pour qu'il s'adapte au comportement désiré, en définissant l'algorithme robuste des C -prototypes (RCP). Dans le cas où les clusters ne sont représentés que par des centres et une partition probabiliste est recherchée (prise en compte de la contrainte probabiliste), les équations de mise à jour des degrés d'appartenance et des prototypes dérivés de l'Eq. (IV. 6) deviennent alors [66] :

$$\vec{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m f_{ij}\, \vec{x}_j}{\sum_{j=1}^{n} u_{ij}^m f_{ij}} \text{ et } u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left(\rho\left(d_{ij}^2\right)\right)}{\rho\left(d_{kj}^2\right)} \right)^{1/(m-1)}} \quad \text{(IV. 7)}$$

Où : $f_{ij} = f(d_{ij})$ et $f = d\rho(z)/dz$. Il est à noter que les valeurs aberrantes ont toujours des degrés d'appartenance $u_{ij} = 1/c$ pour tous les clusters. La différence et l'avantage par rapport aux FCM viennent de leur influence sur le centre, qui est réduit par l'intermédiaire du coefficient f_{ij} (voir [66] pour l'expression de f_{ij}).

Autres méthodes robustes en présence du bruit

Afin de pouvoir traiter l'ensemble de données bruitées, d'autres algorithmes de clustering robustes sont proposés, par exemple Wu et Yang [125] considèrent une modification différente dans la fonction objectif (IV. 6).

Cebron et Berthold (2008) ont proposé un algorithme de classification adaptatif actif qui est une extension de l'approche de Davé [35]. L'algorithme des FCM avec détection du bruit minimise itérativement la fonction objectif en respectant $\vec{v}$ et u :

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \left(\overrightarrow{v_i} - \overrightarrow{x_j}\right)^2 + \delta^2 \sum_{i=1}^{n} \left(1 - \sum_{j=1}^{c} u_{ij} \right)^2 \quad \text{(IV. 8)}$$

Le premier terme correspond à la fonction objectif normale des FCM, cependant le second survient du cluster bruité. δ est la distance entre chaque point de donnée et le cluster bruité c. Cette distance peut être soit fixée soit adaptée à chaque itération selon la distance moyenne entre les points de données. Les objets qui ne sont pas proches de n'importe quel centre du cluster $\overrightarrow{v_i}$ sont alors considérés comme ayant des degrés d'appartenance élevés au cluster bruité. J_m est minimiser sous la contrainte : $\forall i: 0 \leq \sum_{j=1}^{c-1} u_{ij} \leq 1$.

Nasibov et Ulutagay (2007) ont proposé la méthode « Fuzzy Joint Points :FJP » qui associe les valeurs aberrantes dans une classe indépendante. La méthode est robuste dans les problèmes de biologie, médecine, information géographique, « mapping », etc.

En fin, une méthode pour améliorer les performances des FCM et ses dérivatives en présence du bruit, consiste à tenir compte de la validité du cluster. Dans l'exemple de la Fig. (IV.2), si on utilise une mesure de validité convenable, on peut conclure qu'une bonne classification est obtenue en choisissant un nombre de clusters égal à 3. Dans ce cas, les objets bruités sont englobés dans un seul cluster séparé des deux autres. Cette manière d'agir donne un meilleur sens aux degrés d'appartenances des points bruités relativement aux deux autres clusters. Cependant, les mesures de validité restent difficiles à définir et le nombre des clusters qui optimise une validité particulière n'est pas toujours correct. Bin que

les approches qui utilisent la validité de la classification sont efficaces dans certaines situations, mais elles ne peuvent pas corriger le problème de degré d'appartenance relative. Afin de remédier à ce problème, il faut modifier le mode de fonctionnement de l'algorithme. L'algorithme des c-moyennes possibilistes correspond à une telle évolution.

IV.6 Les c-moyennes possibilistes: Possibilistic C-Means (PCM)

Le principe de l'algorithme des c-moyennes possibilistes (PCM) [90], [91], [92] est de caractériser les clusters non plus de manière relative, les uns par rapport aux autres, mais au contraire de façon « absolue », ce qui correspond exactement à la définition d'un caractère intrinsèque. Pour cela, l'idée consiste à relâcher la contrainte sur la sommation à 1 des degrés d'appartenance de l'algorithme des FCM (cf. équation IV.3). Mais si cette contrainte est simplement supprimée, la minimisation de la fonction objectif initiale (équation (IV.1)) a comme solution triviale l'attribution de degrés d'appartenance nuls pour chaque donnée. Pour éviter ce problème, Krishnapuram et Keller [90] ont proposé d'ajouter un second terme au critère de minimisation de l'algorithme des FCM. La fonction objectif des c-moyennes possibilistes est définie comme suit :

$$J_p\big(X, \mathbb{U}_p, C\big) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \, d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} \big(1 - u_{ij}\big)^m \qquad \text{(IV. 9)}$$

Où les paramètres $\eta_i > 0,\ i \in \{1, \dots, c\}$ sont des nombres positifs homogènes à des distances carrées. Plus précisément, η_i est le carré de la distance séparant le centre du cluster *i* de l'ensemble des points dont le degré d'appartenance à ce même cluster est égal à 0.5. La minimisation du deuxième terme implique des degrés d'appartenance les plus élevés possibles, évitant ainsi les solutions triviales. Comme les degrés d'appartenance aux différents clusters sont décorrélés, un centre donné peut être évolué selon n'importe quelle trajectoire indépendamment de tous les autres centres. On se ramène alors, dans ce cas, à la minimisation de c critères partiels indépendants. Un tel critère, $J_i(X, \mathbb{U}_i, c_i)$ correspondant au cluster *i* est de la forme :

$$J_i(X, \mathbb{U}_i, c_i) \ = \sum_{j=1}^{n} u_{ij}^m \, d_{ij}^2 + \eta_i \sum_{j=1}^{n} \big(1 - u_{ij}\big)^m \qquad \text{(IV. 10)}$$

Signalons que (IV.10) n'est, évidemment, qu'un exemple possible de critère pouvant servir à un cluster de type possibiliste. Sauf indication contraire, c'est celui-ci qui sera discuté dans la suite. À titre d'exemple, Krishnapuram et Keller donnent un autre critère possible :

$$J_{pm}\big(X, \mathbb{U}_{pm}, C\big) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij} \, d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} \Big(u_{ij}\big(log\big(u_{ij}\big) - 1\big)\Big) \qquad \text{(IV. 11)}$$

Cependant, c'est l'équation (IV.10) qu'elle est essentiellement appliquée et

commentée. La convergence de l'algorithme est encore assurée [90] à la suite d'un certain nombre d'itérations dans lesquelles les degrés d'appartenance sont mis à jour à l'aide de la formule suivante :

$$u_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{1/(m-1)}} \qquad \text{(IV. 12)}$$

Dans le cas où la distance euclidienne est utilisée, une condition nécessaire (mais non suffisante) d'avoir un minimum (au moins local) du critère défini dans (IV.10), est que la dérivée par rapport à u_{ij} soit nulle :

$$\frac{\partial J_p}{\partial u_{ij}} = 0 \Rightarrow d_{ij}^{2/(m-1)}\, u_{ij} = \eta_i^{1/(m-1)}(1 - u_{ij}) \qquad \text{(IV. 13)}$$

On tire aisément l'expression (IV.12) à partir de l'égalité (IV.13). Si l'espace des données est de dimension supérieure à 1, on peut faire appel aux multiplicateurs de Lagrange (cf. par exemple [13] ou [71]. On peut vérifier également que le critère défini dans (IV.11) peut être minimisé si l'on a :

$$u_{ij} = \frac{1}{exp\left(\frac{d_{ij}^2}{\eta_i}\right)} \qquad \text{(IV. 14)}$$

Les centres des clusters sont toujours calculés de la même façon que dans le cas des FCM. Comme on peut le voir sur l'équation (IV.12), à chaque itération la valeur de u_{ij} ne dépend donc plus que de la distance, d_{ij}, séparant le vecteur x_j du centre c_i, ce qui est conforme à l'esprit de l'approche possibiliste. La position de x_j relativement aux centres des clusters autres que c_i n'interfère donc plus avec le calcul de u_{ij}. La relation précédente (IV.12), peut être étendue à toutes les valeurs (continues) possibles de la distance, et on définit, ainsi, une distribution de possibilité associée au cluster *i* (ayant pour centre c_i) sur l'ensemble de discours (IV.15).

$$u_i(x) = \frac{1}{1 + \left(\frac{d^2(x, c_i)}{\eta_i}\right)^{1/(m-1)}} \qquad \text{(IV. 15)}$$

Cette distribution de possibilité atteint, évidemment son maximum, lorsque la distance *d* est nulle, donc au centre du cluster *i.* Cette loi ressemble à la forme des fonctions d'appartenance préconisée par Zimmermann et Zysno [133]. Pour un cluster donné, l'influence du paramètre *m* sur la fonction d'appartenance est illustrée (dans le cas de la distance euclidienne) sur la figure IV.4 en choisissant comme variable la quantité :

$$t = \frac{d^2(x, c_i)}{\eta_i} \qquad \text{(IV. 16)}$$

On remarque, comme dans le cas des FCM, que la partition est d'autant plus floue que m est élevé. Lorsque la valeur de m tend vers 1, on se rapproche d'une fonction d'appartenance rectangulaire. Lorsque le paramètre m varie, toutes les courbes passent par un point fixe de coordonnées : $t = 1$ et $u = 0.5$.

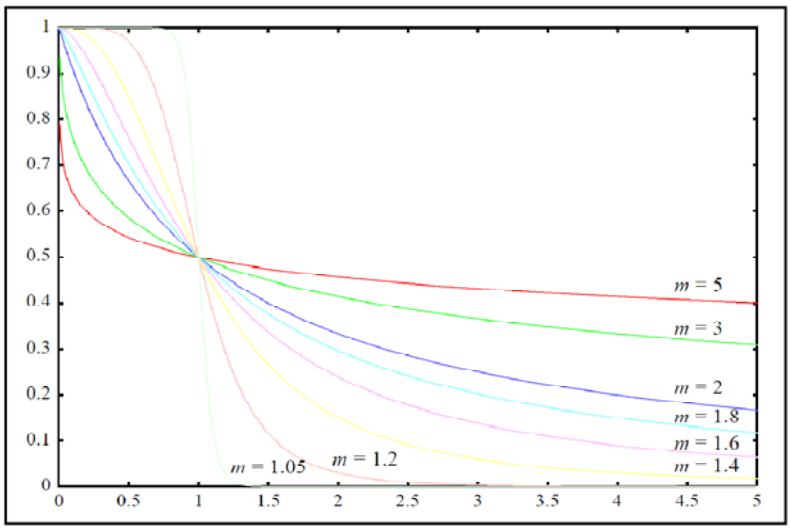


Fig. IV.4: Fonctions d'appartenance pour différentes valeurs de m.

Si la valeur de m est égale à 2, la fonction d'appartenance présente une allure intermédiaire, et cette valeur a l'avantage de correspondre à des calculs plus simples que dans le cas d'une autre valeur quelconque. Le schéma algorithmique général est le suivant :

Algorithme des PCM

$X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\} \subset \Re^p$ est l'ensemble des données à partitionner.

5. *Initialisation*
   - Fixer le nombre de clusters $c \in]1, n]$, $m \in]1, \infty[$, la distance d , le nombre d'itérations $l = 0$ *et* $\varepsilon > 0$;
   - Initialiser la C-partition possibiliste $\mathbb{U}_{p(0)}$ à l'aide de (IV.3);
   - Estimer les $\eta_{i(0)}$ en utilisant (IV.17);

6. *Evaluation des clusters*
   - Calculer u_{ij} à l'aide de l'équation (IV.12) ;

7. *Adaptation des prototypes*
   - Evaluer les C-prototypes possibilistes $\{c_{i(l)}\}$ à l'aide de l'équation (IV.2) ;

8. *Test d'arrêt*
   - Comparer $u_{ij(l-1)}$ et $u_{ij(l)}$ si $\|u_{ij(l)} - u_{ij(l-1)}\| < \varepsilon$ alors **STOP**
     Si non $l \leftarrow l+1$; recommencer à l'étape é*valuation des clusters*
     Fin si

Dans le cas où l'ensemble des données est relativement exempte de bruit, les résultats d'une partition floue (les FCM ou ses dérivatives) donnent une excellente estimation des η_i; par exemple on peut estimer les η_i par:

$$\eta_i = \frac{\sum_{j=1}^{n} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m} \quad \text{(III. 17)}$$

En pratique la valeur de k est choisie égale à 1. Lorsque les données sont bruités, la partition initiale produite par les FCM est mauvaise, ce qui implique une mauvaise estimation des η_i. Dans cette situation, on doit estimer les valeurs des η_i basées sur une première partition floue selon (IV.3), et après la convergence de l'algorithme, on doit recompter les valeurs des η_i avec précision en utilisant l'équation suivante :

$$\eta_i = \frac{\sum_{x_j \in (\Pi_i)_\alpha} d_{ij}^2}{|(\Pi_i)_\alpha|} \quad \text{(III. 18)}$$

Où $(\Pi_i)_\alpha$ représente l'α-coupe associée au sous-ensemble flous définit par u_{ij} et $|(\Pi_i)_\alpha|$ est le nombre d'objets de cette α-coupe. Le second passage au moyen de l'algorithme avec des valeurs de raffinage pour les η_i permet aux appartenances résultantes dans un environnement bruité d'être presque identiques à ceux obtenues en état d'absence de bruit. N'importe quelle valeur d'alpha entre 0,1 et 0,4 semble donner des résultats compatibles [92]. De par son mode de fonctionnement l'algorithme est très sensible à l'initialisation. D'autres algorithmes comme celui de Gath et Geva [67] (cf. paragraphe III.3.4) sont donc très souvent utilisés dans un premier temps pour obtenir la partition initiale.

IV.6.1 Propriétés de l'algorithme des PCM

Avec les modifications apportées aux FCM, l'algorithme des PCM permet d'obtenir des fonctions d'appartenance aux différents clusters qui ne sont plus dépendants de leurs interactions mais uniquement de la distance d'une donnée aux centres. Cela se traduit au niveau de l'algorithme, lors des itérations par une mobilité plus réduite des prototypes qui n'ont qu'un champ d'action limité par η_i. C'est une des raisons principales qui rend la phase d'initialisation de l'algorithme très importante. En revanche, cela permet de distinguer clairement les objets proches des centroïdes de ceux qui en sont éloignés. La forme des fonctions d'appartenance est illustrée sur la Fig. IV.5.

Le résultat de l'algorithme possibiliste présenté sur la Fig. IV.6 montre bien que cette fois, l'objet A possède un degré d'appartenance très nettement inférieur à celui de$B$. De plus, comme annoncé, l'algorithme est beaucoup moins sensible à la présence du bruit dans les données (ce qui est illustré par le point c sur la figure). Grâce à ces propriétés, l'algorithme des PCM permet d'extraire automatiquement des sous-ensembles flous caractérisant d'une manière intrinsèque l'organisation spatiale d'un ensemble de données.

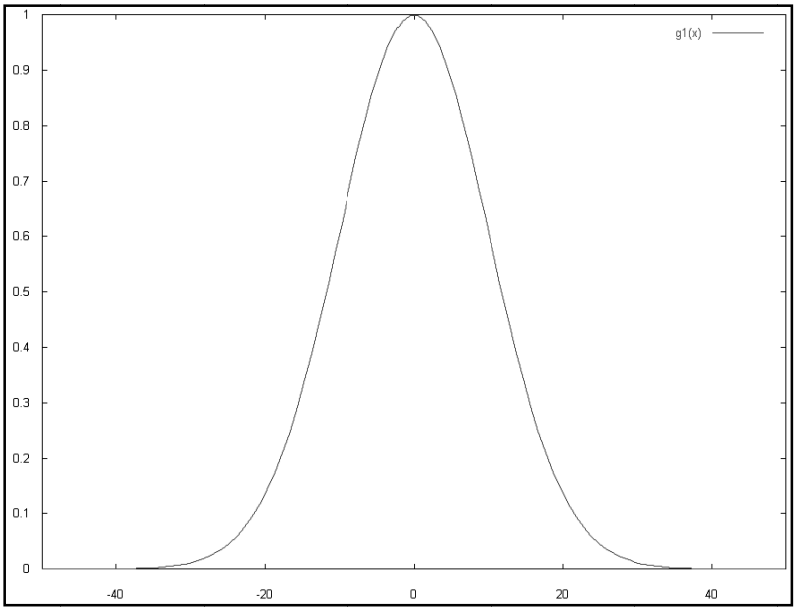


Fig. IV.5 : Forme d'une fonction d'appartenance issue des PCM en une dimension.

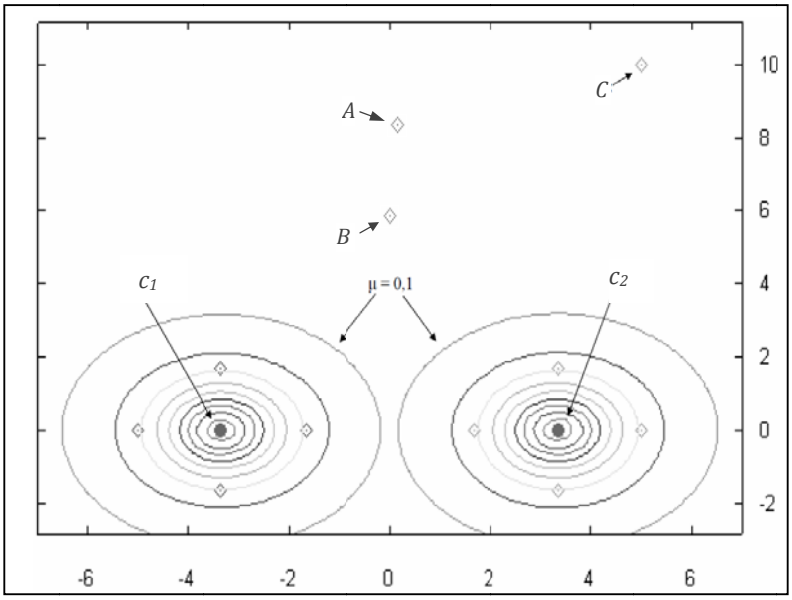


Fig. IV.6 : Immunité au bruit des prototypes issus des c-moyennes possibilistes

IV.6.2 Problème de coïncidence dans les PCM

La fonction objectif correspondante à un cluster i peut être formulée comme suit :

$$J_i(X, \mathbb{U}_i, c_i) = \sum_{j=1}^{n} u_{ij}^m \, d_{ij}^2 + \eta_i \sum_{j=1}^{n} \left(1 - u_{ij}\right)^m \qquad \text{(IV. 19)}$$

Les degrés d'appartenances générés avec les PCM ne sont pas liés par la contrainte d'inspiration probabiliste $\sum_{i=1}^{c} u_{ij} = 1 \ \ \forall j \in \{1, \ldots, n\}$. La mise à jour de l'équation qui calcule les degrés d'appartenance dans les PCM est donnée par (IV.12). À partir de l'équation (IV.12), on obtient :

$$d_{ij}^2 = \eta_i \left(\frac{1 - u_{ij}}{u_{ij}}\right)^{m-1} \qquad \text{(IV. 20)}$$

En éliminant d_{ij}^2, la fonction objectif définie dans (IV.19) devient :

$$J_i(X, \mathbb{U}_i, c_i) = \eta_i \sum_{j=1}^{n} \left(1 - u_{ij}\right)^{m-1} \qquad \text{(IV. 21)}$$

Pour une valeur donnée de η_i, minimiser $J_i(X, \mathbb{U}_i, c_i)$ est équivalent à maximiser:

$$J_i(X, \mathbb{U}_i, c_i) = \eta_i \sum_{j=1}^{n} \left(1 - \left(1 - u_{ij}\right)^{m-1}\right) = \eta_i \sum_{j=1}^{n} \acute{u}_{ij} \qquad \text{(IV. 22)}$$

Où :
$\acute{u}_{ij} = 1 - \left(1 - u_{ij}\right)^{m-1}$peut être interprété comme étant un degré d'appartenance modifié. Il est obtenu à partir de u_{ij} via un tracé monotone :

$$\frac{d\acute{u}_{ij}}{du_{ij}} = (m-1)\left(1 - u_{ij}\right)^{m-2} > 0 \text{ pour m} > 1$$

D'où, $\acute{u}_{ij}$ varie de la même façon que u_{ij} i.e., $u_{ij} = 0 \Rightarrow \acute{u}_{ij} = 0$; $u_{ij} = 1 \Rightarrow \acute{u}_{ij} = 1$, ces derniers sont des fonctions monotones décroissantes de d_{ij}^2. De plus, dans le cas spécial où $m = 2$, l'équation (V.22) se réduit à :

$$\acute{J}_i(X, \mathbb{U}_i, c_i) = \eta_i \sum_{j=1}^{n} \acute{u}_{ij} \qquad \text{(IV. 23)}$$

À partir des deux équations (IV.22) et (IV.23), on remarque que pour une valeur donnée de η_i , tous les C sous-fonctions objectives sont maximisées par le choix des positions des prototypes de tels sorte que la somme des degrés d'appartenance modifiés soit maximisée. Ceci est vérifié lorsque les prototypes sont localisés dans des régions denses, du fait que la fonction d'appartenance est une fonction

monotone et décroissante de la distance. S'il existe réellement C-régions denses, alors avec une bonne initialisation, chaque prototype converge vers une région dense. Dans une telle situation, même si tous les η_i sont égaux chacun d'eux aura C-distincts minimums correspondants aux C-régions denses.

Le mauvais comportement des PCM, est sa tendance à générer des clusters ayant des centres identiques, est dû au fait que les degrés d'appartenance générés par les PCM sont généralement très voisins les uns des autres pour tous les objets dans toutes les clusters. Cette situation engendre un glissement progressif des centres des clusters vers le centre de données à partitionner. Plusieurs modifications ont été appliquées sur les PCM pour contourner le problème de coïncidence.

IV.6.3 Surmonter les problèmes des c-moyennes possibilistes

Krishnapuram et Keller ont proposé une autre fonction objectif indépendante de *m* définie par :

$$J_{pm}(X, \mathbb{U}_{pm}, C) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}\, d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} \left(u_{ij}\left(log(u_{ij}) - 1\right)\right)$$

Alors la mise à jour des u_{ij} est donnée par :

$$u_{ij} = \frac{1}{exp\left(\frac{d_{ij}^2}{\eta_i}\right)}$$

L'équation de mise à jour des prototypes reste inchangeable. Dans ce cas, la fonction exponentielle descend plus rapidement pour les grandes valeurs de $d^2(x_j, c_i)$. Cette formulation est mieux adaptée lorsque les clusters deviennent probablement compacts. Il est à noté que $u_{ij}\left(log(u_{ij}) - 1\right)$ est une fonction monotone décroissante dans [0,1]. Si on utilise la distance de Mahalambis, alors on peut éliminer η_i (i.e., η_i =1).

Khodja [85] a proposé aussi d'autres modifications pour corriger ce problème. Il affaiblit la valeur de *m* en modifiant l'équation qui permet de calculer les degrés d'appartenance:

$$u_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{\frac{m}{m-1}}} \qquad \text{(IV. 24)}$$

Qu'est obtenue par dérivation de la fonction objectif suivante :

$$J_m(X, \mathbb{U}\ \ , C) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m\, d_{ij}^{2m} + \sum_{i=1}^{c} {\eta_i}^m \sum_{j=1}^{n} \left(1 - u_{ij}\right)^m \qquad \text{(IV. 25)}$$

Dans ce cas pour les valeurs de *m<2* les degrés d'appartenance des objets qui sont éloignées d'un cluster seront affaiblis et ceux qui lui appartiennent seront amplifiés.

Ouled Ahmedou [100] a proposé une approche appelée les PCMM (Possibilistic C-Means Modifier). L'algorithme consiste à trouver une partition de telle manière que le degré d'appartenance d'un objet à le cluster gagnante représente un degré d'appartenance renforcé par un élargissement de l'extension, alors que ce degré est abaissé en affaiblissant l'extension de chaque cluster restante. Il a proposé une nouvelle méthode de calcul de η_i :

$$\eta_{ik} = \begin{cases} \eta_1 & \text{si} \quad i = \arg\min_j \left\{ \left\| x_k - v_j \right\|^2 \right\} (\text{i} = \text{gagnant}) \\ \eta_2 & \text{sinon} \end{cases}$$

avec $\eta_1 \succ \eta_2$

Pour le calcul de η_1 et η_2, on peut choisir par exemple:

Où $\eta_1 = D^2 et \eta_2 = \frac{\eta_1}{2}$

$$D = \max_{i \neq j} \left\| v_i - v_j \right\|$$

Le choix de η_1 et η_2 est donné par:

$$\eta_1 = \begin{cases} D^2 & \text{si } D^2 \geq 1 \\ \sqrt{D} & \text{sinon} \end{cases}$$

$$\eta_2 = \begin{cases} \sqrt{D} & \text{si} \quad D^2 \geq 1 \\ D^2 & \text{sin } on \end{cases}$$

Dans les travaux récents de [102], un algorithme de clustering a été proposé optimise la fonction objectif suivante :

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} \left(a u_{ij}^m + b t_{ij}^\eta \right) d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{i=1}^{n} \left(1 - t_{ij} \right)^\eta \qquad \text{(III. 26)}$$

Où, a et b sont des paramètres définis par l'utilisateur. Dans le cas où la distance euclidienne est utilisée, les équations de mise à jour sont alors :

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(d_{ij}^2 / d_{ik}^2 \right)^{1/(m-1)}},$$

$$t_{ij} = \frac{1}{1 + \left((b/\eta_i) d_{ij}^2 \right)^{1/(\eta-1)}}, \quad \vec{c}_i = \frac{\sum_{j=1}^{n} \left(a u_{ij}^m + b t_{ij}^\eta \right) \vec{x}_j}{\sum_{j=1}^{n} \left(a u_{ij}^m + b t_{ij}^\eta \right)}$$

Ainsi, u_{ij} sont semblables aux degrés d'appartenance des FCM (voir Eq. (IV.3)), et t_{ij} aux coefficients possibilistes des PCM en remplaçant η_i avec η_i/b (voir l'équation (IV.12)). Les centres des clusters dépendent alors, à la fois, des coefficients : les valeurs a, b, m, et le paramètre η qui mène son influence relative. Cela montre que si b est supérieur à a, les centres seront plus influencés par les coefficients possibilistes que par les degrés d'appartenance. Ainsi, pour réduire l'influence des valeurs aberrantes, une valeur plus grande pour b que a doit être utilisée. Pourtant, il est à remarquer que quatre paramètres sont à définir par l'utilisateur, et que leur influence est corrélée, ce qui rend, la détermination de leurs valeurs optimales un peu difficile.

IV.7 L'approche proposée

IV.7.1 Présentation du problème

Nous avons vu que les FCM reposent sur l'appartenance relative en s'appuyant sur la contrainte d'inspiration probabiliste. Dans ce cas, les degrés d'appartenance sont interprétés comme des degrés de partage et non pas de typicalité. En effet, les degrés d'appartenance des deux points équidistants d'un prototype d'un cluster peuvent devenir différents. Réciproquement, on peut trouver le cas où les deux points ont le même degré d'appartenance alors que ces derniers sont arbitrairement éloignés l'un de l'autre. Ceci dégrade les performances des FCM en présence du bruit. Pour surmonter ce problème, les PCM relâchent la contrainte et la typicalité est remplacée par le partage ; signalons encore une fois que les degrés d'appartenance générés par les PCM sont très voisins et peuvent engendrer des centres identiques.

Pendant que les degrés possibilistes fassent leur possible pour réduire l'influence des valeurs aberrantes, les appartenances floues (probabilistes) assurent l'assignation nécessaire de tous les points dans un ensemble de données, ne laissant pas, ainsi aucune donnée non classifiée. Comme le résultat de clustering exige à la fois une bonne propriété de clustering des FCM (probabilistes), et une propriété robuste du mode de recherche des c-moyennes possibilistes, l'utilisation des deux concepts d'appartenance possibiliste et flou dans un même algorithme de clustering s'avère efficace pour la résolution du problème bruit-coïncidence.

Le clustering avec les deux assignations : typiques (ici:) et normalisés ( ) est réalisé grâce à l'optimisation de la fonction objectif suivante :

$$J_{fp}(X, \mathbb{U}_{fp}, C) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\, d_{ij}^{2} + \sum_{i=1}^{c}\sum_{j=1}^{n} t_{ij}^{m}\, d_{ij}^{2} + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - t_{ij})^{m} \qquad \text{(IV. 28)}$$

Où, les contraintes connues sont appliquées aux deux types respectifs des assignations graduels. Dans le cas où la distance euclidienne est utilisée, cherchant les équations de mise à jour implique une application du modèle AO. En effet, une condition nécessaire (mais non suffisante) d'avoir un minimum du critère défini

dans (IV.28), est que les dérivées par rapport à u_{ij}, par rapport à t_{ij} et par rapport à v_i appliquées d'une manière alternative soient nulles. Dans le cas très simple de données monodimensionnelles, il vient :

$$\frac{\partial J_{fp}}{\partial t_{ij}} = 0 \Rightarrow t_{ij} = \frac{1}{1 + \left({d_{ij}^2}/{\eta_i} \right)^{1/(m-1)}} \quad \text{(IV. 29)}$$

$$\frac{\partial J_{fp}}{\partial u_{ij}} = 0 \Rightarrow u_{ij} = \frac{d_{ij}^{-2/(m-1)}}{\sum_{t=1}^{c} d_{tj}^{-2/(m-1)}} \quad \text{(IV. 30)}$$

$$\frac{\partial J_{fp}}{\partial v_i} = 0 \Rightarrow \begin{cases} \vec{c}_i = \dfrac{\sum_{j=1}^{n} u_{ij}^m \, \vec{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \text{ si } d_{ij}^2 \leq \eta_i & \text{(IV. 31)} \\ \vec{c}_i = \dfrac{\sum_{j=1}^{n} t_{ij}^m \, \vec{x}_j}{\sum_{j=1}^{n} t_{ij}^m} \text{ si } d_{ij}^2 > \eta_i & \text{(IV. 32)} \end{cases}$$

Si l'espace des données est de dimension supérieure à 1, on peut faire appel aux multiplicateurs de Lagrange.

IV.7.2 Principe

Pour résoudre le problème de coïncidence, il est convenable que les objets d'un même cluster doivent avoir des degrés d'appartenance plus forts que ceux des autres clusters. Dans ce cas, nous attribuons à chaque objet qui est susceptible d'être dans un cluster, au sens possibiliste, un degré flou et un degré possibiliste aux clusters restants. Le critère de minimisation des FPCM [23] est basé sur la fusion des deux concepts flou et possibiliste. Le concept d'appartenance des FPCM est défini par les règles suivantes :

- **R1.** Si un objet appartient à la zone d'influence d'un cluster alors, on l'associé à ce dernier avec un degré flou et aux autres avec des degrés possibilistes.
- **R2.** Si un objet appartient à l'intersection des deux ou plusieurs zones d'influences on lui assigne un degré flou à chacun des clusters chevauchants.
- **R3.** Si un objet n'appartient à aucune zone d'influence, on lui attribue un degré possibiliste à chacun des clusters.

Exemple

La Fig. IV.7 et le tableau IV.1 représentent respectivement quelques situations typiques.

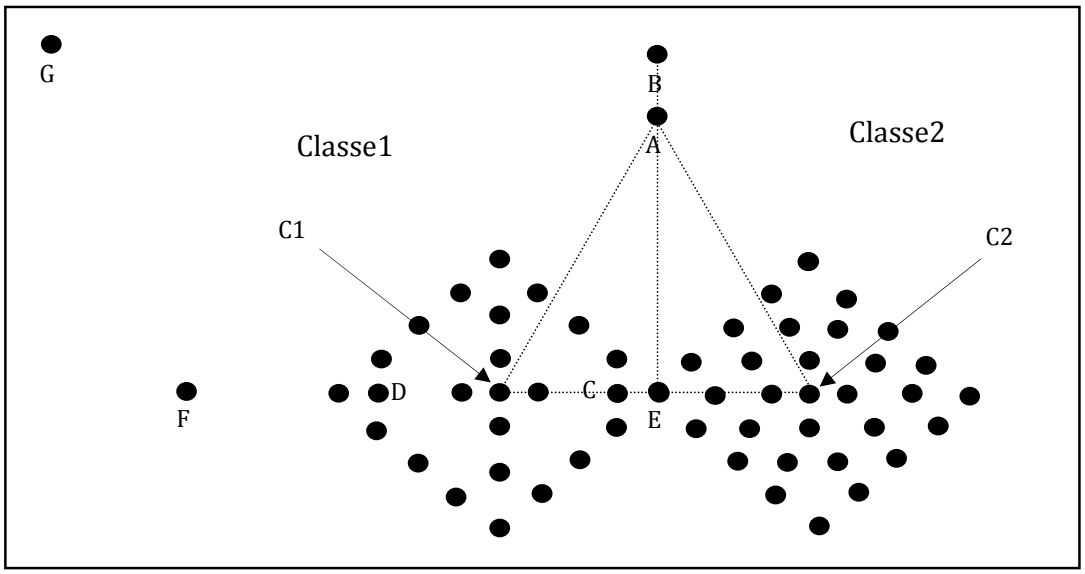


Fig. IV.7 : Représentation de quelques situations typiques.

Algorithme de classification	$u_A > u_B$	$u_{1C} = u_{1D}$	$u_{1C} > u_{1F}$	$u_{1E} = u_{2E}$	$u_{1E} > u_{1G}$
FCM	Non	Non	Non	Oui	Non
PCM	Oui	Oui	Oui	Non	Oui
FPCM	Oui	Oui	Oui	Oui	Oui

Tableau IV.1 : Satisfaction de quelques règles empiriques par les : FCM, PCM et les FPCM

1. $u_A > u_B$: À cause de la contrainte imposée sur les degrés d'appartenance, les FCM assignent les deux points A et B aux deux clusters (1 et 2) avec le même degré d'appartenance, néanmoins le point B est moins typique que le point A. Dans le cas des PCM, le point B est affecté aux deux clusters avec des degrés moins typiques que ceux du point A. Avec les FPCM, la situation est la même que celle dans les PCM du fait que les points A et B sont considérés comme des points hors de la zone d'influence de chaque cluster.

2. $u_{1C} = u_{1D}$: Avec les FCM, le degré d'appartenance au premier cluster du point C est inférieur à celui du point D, parce que le point C est partagé entre les deux clusters. Contrairement, les PCM attribuent aux deux points le même degré d'appartenance. Dans le cas des FPCM, le point D est affecté au premier cluster avec un degré flou, le point C est affecté au premier cluster avec un degré flou et à

la deuxième avec un degré possibiliste ce qui élimine l'opération de partage avec le deuxième cluster.

3. $u_{1C} = u_{1F}$: Les FCM assignent le point F au premier cluster avec un degré d'appartenance plus élevé que celui du point C. Avec les PCM, le point C prend un degré d'appartenance plus fort que celui du point F. Dans le cas des FPCM, le point C est assigné au premier cluster avec un degré flou plus fort que celui du point F qui est possibiliste.

4. $u_{1E} = u_{2E}$: Le point E est assigné avec le même degré d'appartenance aux deux clusters (1 et 2) avec les FCM, alors que, avec les PCM, les deux degrés sont différents. Ceci est dû à la variation des zones d'influence des deux clusters (les objets du premier cluster ont plus de la tendance de mobilité que ceux du deuxième cluster). Du fait que E appartient à la zone d'intersection des deux clusters, les FPCM appliquent la notion de partage et élimine ainsi le problème de chevauchement.

5. $u_{1E} = u_{1G}$: Les FCM assignent le point bruité G au premier cluster avec un degré plus élevé que celui du point E. Les PCM attribuent à chacun d'eux un degré en fonction des distances réelles. Tandis que les FPCM considèrent le point G comme étant un point étranger et ils lui attribut alors un degré possibiliste d'où l'élimination du problème du bruit.

IV.7.3 Etapes de l'algorithme des FPCM

$X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_n}\} \subset \Re^p$ est l'ensemble des données à partitionner.

Initialisation
- Fixer le nombre de clusters $c \in]1, n]$, $m \in]1, \infty[$, la distance d , le nombre d'itérations $l = 0$ *et* $\varepsilon > 0$;
- Initialiser la C-partition flou-possibiliste $\mathbb{U}_{p(0)}$ à l'aide de (IV.3);
- Estimer les $\eta_{i(0)}$ en utilisant (IV.17);

**Si* $d_{ij}^2 \leq \eta_i$ *alors :*

Evaluation des clusters
- Calculer les u_{ij} flous à l'aide de l'équation (IV.30);

Adaptation des prototypes
- Evaluer les C-prototypes flous $\{c_{i(l)}\}$ à l'aide de l'équation (IV.31) ;

Sinon

Evaluation des clusters
- Calculer les t_{ij} possibiliste à l'aide de l'équation (IV.29);
- $u_{ij} \leftarrow t_{ij}$;

Adaptation des prototypes
- Evaluer les C-prototypes possibilistes $\{c_{i(l)}\}$ à l'aide de l'équation (IV.32) ;

Fin si

Test d'arrêt

- Comparer $u_{ij(l-1)}$ et $u_{ij(l)}$ si $\|u_{ij(l)} - u_{ij(l-1)}\| < \varepsilon$ alors **STOP**
  Sinon $l \leftarrow l+1$; recommencer à *
  Fin si

IV.8 Conclusion

Dans ce chapitre, une étude comparative entre les algorithmes de clustering flous de la famille des c-moyennes a été présentée, suivie de la proposition d'une nouvelle approche [23]. Sur le plan théorique, l'approche que nous avons proposée (FPCM) qui se base sur la fusion des deux concepts d'appartenance probabiliste et possibiliste permet de:

- Résoudre le problème de chevauchement entre deux ou plusieurs clusters en appliquant un concept probabiliste à l'intérieur des zones d'influences des clusters,
- Éliminer le problème de bruit en utilisant un concept possibiliste en dehors des zones d'influences des clusters,
- Contourner au problème de coïncidence des centres de clusters, grâce à la diversité des degrés d'appartenances générés.

Chapitre V

Evaluation des résultats du clustering

V.1 Introduction

Malgré la popularité des méthodes de clustering, récemment, peu d'attention a été accordée à ce qu'on prévoit par la sortie d'un algorithme de clustering. Un classifieur prend un ensemble de points de données et les partitionne en clusters. Mais quel est le sens du résultat ? Existe-t-il plus qu'une simple image ? Peut-on argumenter qu'une procédure de clustering est meilleure que d'autre ? Toutes ces questions pointent sur la base épistémologique du clustering [25]. Le clustering est essentiellement un outil de visualisation subjective. Jain et al. [78] ont écrit, « Le clustering est un processus subjectif, le même ensemble de données doit souvent être partagée différemment pour différentes applications. Cette subjectivité rend le processus de regroupement difficile ». Historiquement, une grande variété de mesures de "validité" ont été proposées pour évaluer les résultats de clustering [62], [119], [70], [73], [96] et [18]. Les mesures de validité proposées pour les algorithmes de clustering se divisent en trois classes. Le premier type est basé sur le calcul des propriétés résultantes du cluster, comme la compacité et la séparation. Cette approche est appelée la *validation interne*, car elle ne nécessite pas une information additionnelle sur les données [73], [96] et [53]. Une deuxième approche est basée sur la comparaison des partitions générées par le même algorithme avec des paramètres différents. Celle-ci est appelée *validation relative,* et ne comprend pas une information additionnelle [73], [10] et [129]. La troisième méthode, dite *validation externe,* est également basée sur la comparaison des partitions ; les partitions à comparer se composent de la partition générée par l'algorithme de clustering et de la partition donnée (ou d'un sous-ensemble de données) [96] et [46]. La validation externe correspond à un type de mesure d'erreur, soit directement, soit indirectement. Fig. (V.1) montre une hiérarchie des techniques de validation.

Dans ce chapitre, nous examinerons un certain nombre de mesures de validité proposées dans la littérature pour évaluer les résultats du clustering. Ensuite, nous présenterons une étude expérimentale pour examiner les performances des algorithmes des c-moyennes flous, c-moyennes possibilistes et des c-moyennes flous-possibilistes. Pour déterminer la tendance de chaque algorithme à

reconnaitre les groupes existants, des tests ont été réalisés sur trois bases de données (Iris, cuisse humaine et image des texturés).

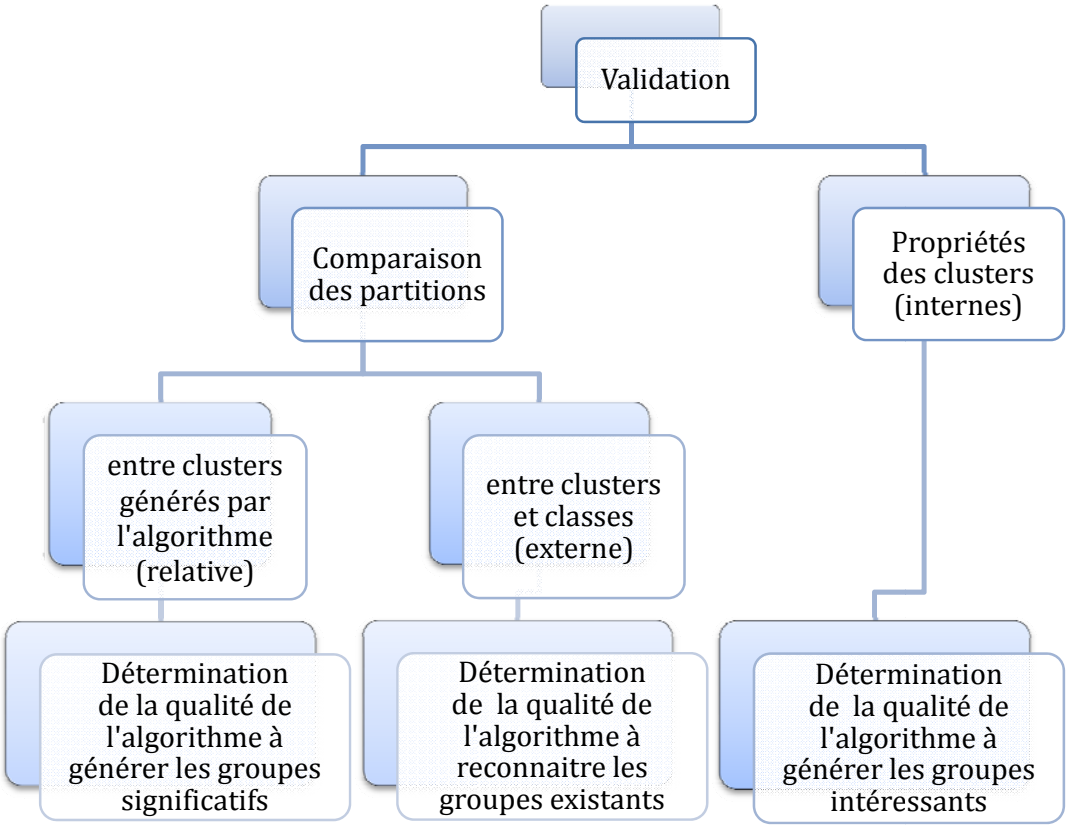


Fig.V.1: Une classification simplifiée des techniques de validation.

V.2 Indices de validation Interne

Pour la validation interne, l'évaluation des résultats des clusters est fondée sur les clusters eux-mêmes, sans informations supplémentaires ou de répétitions de la procédure de clustering. Cette famille de techniques est basée sur l'hypothèse que les algorithmes devraient chercher les clusters, dont les membres sont proches les uns des autres et loin d'être des membres d'autres clusters. Ci-dessous, nous décrivons quelques indices de validation interne.

V.2.1 Indices de Dunn

L'indice de validation de Dunn est défini comme le rapport entre la distance minimale entre les deux clusters et la taille du plus grand cluster [2], [3] et [22]. Si $\varphi = \{c_1, \dots, c_K\}$ est une partition de n points en K clusters, alors l'indice est défini par :

$$V(\varphi) = \frac{\min_{h,k=1,\dots,K,h\neq k} d_c(c_k, c_h)}{\max_{k=1,\dots,K} \Delta(c_k)} \qquad (V.1)$$

Où $d_c(c_k, c_h)$ est la distance entre les deux clusters et $\Delta(c_k)$ est la taille du cluster c_k. La valeur de $V(\varphi)$ dépend de la sélection des mesures de distance. Plusieurs mesures de distances entre les clusters (ou lien) sont proposées dans Ref. [22]: *simple, complet, moyenne, moyenne au centroïde et métriques de Haussdorff*. Le tableau (V.1) présente la définition de chacune de ces mesures de distance. La taille du cluster peut être définie de manières multiples. Certaines des mesures définies dans [3] sont c*omplètes, moyennes* et *centroïdes*. Le tableau (V.2) montre la définition de chacune de ces mesures. Chaque combinaison de mesure de distance et de taille de cluster définit un indice de Dunn différent.

Lien	**Equation**
simple	$d_c(c_i, c_j) = \min_{x\in c_i, y\in c_{j,}} d(x,y)$
complet	$d_c(c_i, c_j) = \max_{x\in c_i, y\in c_{j,}} d(x,y)$
moyenne [a]	$d_c(c_i, c_j) = \frac{1}{n_i n_j} \sum_{x\in c_i, y\in c_{j,}} d(x,y)$
centroïde	$d_c(c_i, c_j) = d(\bar{x}, \bar{y})$
moyenne au centroïde [b]	$d_c(c_i, c_j) = \frac{1}{n_i+n_j}\left[\sum_{x\in c_i} d(x,\bar{y}) + \sum_{y\in c_j} d(x,\bar{y})\right]$
Métriques de Hausdorff [c]	$d_c(c_i, c_j) = \max(d_H(c_i, c_j), d_H(c_j, c_i))$

Tableau V.2. Méthodes de linkage entre deux clusters

[a] n_i et n_j sont les nombres d'échantillons dans les clusters c_i et c_j, respectivement.
[b] $\bar{x}$ et $\bar{y}$ sont les centroïde des clusters c_i et c_j, respectivement.
[c] $d_H(A,B) = \max_{x\in A} \min_{y\in B} d(x,y)$

Mesure	**Equation**
Complète	$\Delta(\mathrm{c}) = \max_{\mathrm{x,y}\in\mathrm{c}} \mathrm{d}(\mathrm{x},\mathrm{y})$
moyenne [a]	$d_c(c_i, c_j) = \frac{1}{n*(n-1)} \sum_{x,y\in c} d(x,y)$
Centroïde [b]	$\Delta(c) = \frac{2}{\lvert c\rvert} \sum_{x\in c} d(x,\bar{x})$

Tableau V.3. Mesures de la taille de cluster

[a] n est le nombre d'échantillons en clusters c.
[b] $\bar{x}$ est le barycentre des clusters c.

V.2.2 Indice de Silhouette

La *silhouette* est la moyenne, sur tous les clusters, de la *largeur de silhouette* de leurs points [3] et [22]. Si x est un point dans le cluster, c_k et n_k est le nombre de points dans c_k, alors, la *largeur de la silhouette* de x est définie par le rapport :

$$S(x) = \frac{b(x) - a(x)}{\max[b(x), a(x)]} \qquad (V.2)$$

Où $a(x)$ est la distance moyenne entre x et tous les autres points dans c_k,

$$a(x) = \frac{1}{n_k - 1} \sum_{y\in c_k, y\neq x} d(x,y) \qquad (V.3)$$

Et $b(x)$ est le minimum de la moyenne des distances entre x et les points dans les autres clusters,

$$b(x) = \min_{h=1,\ldots,K, h\neq k} \left[\frac{1}{n_h} \sum_{y\in c_h} d(x,y) \right] \qquad (V.4)$$

Enfin, l'indice de silhouette global est défini par :

$$S = \frac{1}{K} \sum_{k=1}^{K} \left[\frac{1}{n_k} \sum_{x\in c_k} S(x) \right] \qquad (V.5)$$

Pour un point donné x, sa largeur de silhouette varie de -1 à 1. Si la valeur est proche de -1, cela signifie que le point est plus proche, en moyenne, à un autre

cluster auquel il appartient. Si la valeur est proche de 1, cela signifie que la distance moyenne à son propre cluster est nettement plus petite qu'à tout autre cluster. Une valeur augmentée de la silhouette, correspond aux clusters les plus compacts et les plus séparés.

V.2.3 Corrélation de Hubert avec une matrice de distance

Soit $\varphi = \{c_1, \dots, c_K\}$ une partition de l'ensemble de n objets en K clusters, et P une matrice de similarité entre les n objets tels que $P(i,j)$ est une mesure de similarité entre $x_i$et x_j. La relation entre les deux vecteurs, qu'ils appartiennent au même cluster ou non, peut être représentée par une matrice de similarité D définie par $D(i,j)$=1 si x_i et x_j appartiennent au même cluster, et $D(i,j) = 0$ s'ils appartiennent aux clusters différents. La corrélation entre les deux matrices Γ_D donne une mesure de similarité entre eux :

$$\Gamma_D = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} D(i,j)P(i,j) \qquad (V.6)$$

Avec $M = n(n-1)/2$, le nombre de paires de points différents.

L'indice Γ_D est classé comme un indice interne, car il est fondé uniquement sur la partition φ définie par l'algorithme de clustering et de la similarité entre les points à regrouper.

V.3 Indices de validation externe

Dans la validation externe, la qualité d'un algorithme est évaluée en comparant les clusters résultants avec l'information pré-spécifique.

V.3.1 Corrélation de Hubert

Supposons qu'il existe deux partitions du même ensemble de n objets en K clusters $\varphi^A = \{c_1^A, \dots, c_K^A\}$, définies par des informations supplémentaires sur le problème (appelée la *vraie* partition), et $\varphi^B = \{c_1^B, \dots, c_K^B\}$ obtenu par l'application d'un algorithme de clustering (appelé le *clustering* de la partition). Les ensembles c_k^A sont appelés des *classes* et les ensembles c_k^B sont appelés *clusters*. Pour chaque partition φ, la relation entre les deux vecteurs, qu'ils appartiennent au même cluster ou non, peut être représentée par une matrice de similarité définie par $d(i,j) = 1$ si x_i et x_j appartiennent au même cluster, et $d(i,j) = 0$ s'ils appartiennent aux différents clusters.

Si d^A et d^B sont les matrices de similarité induites par les deux partitions, φ^Aet φ^B, alors deux indices de similarité sont calculés comme des fonctions de corrélations et des covariances de ces matrices, le Hubert Γ statistique :

$$\Gamma = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d^A(i,j)\, d^B(i,j) \qquad (V.7)$$

Et Γ^*normalisé statistique:

$$\Gamma^* = \frac{1}{M\sigma^A\sigma^B}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} (d^A(i,j)-\mu_A)\,(d^B(i,j)-\mu_B), \qquad (V.8)$$

Où $M=n(n\text{-}1)/2$ est le nombre de paires de points différents, μ^A, μ^B, σ^A et σ^B sont les moyens de l'échantillon et les écarts-types des valeurs dans les matrices d^A et d^B. L'Hubert statistique est basé sur le fait que plus les partitions sont similaires, plus les matrices pouvaient être similaires, et cette similarité peut être mesurée par leur corrélation.

V.3.2 Rends statistiques, coefficient de Jaccard et indices de Folkes et Mallows

Etant donné la vraie partition $\varphi^A = \{c_1^A, \dots, c_K^A\}$ et le clustering de la partition φ^B, pour chaque paire d'échantillons x, y $(x \neq y)$, il y a quatre situations possibles :

(a) x et y tombent dans le même cluster à la fois dans φ^A et φ^B,
(b) x et y tombent dans le même cluster dans φ^A, mais à différents clusters dans φ^B,
(c) x et y tombent dans les différents clusters dans φ^A, mais dans le même cluster dans φ^B,
(d) x et y tombent dans les différents clusters à la fois dans φ^A et φ^B,

La mesure de désagrément entre φ^A et φ^Best quantifiée par le nombre de paires de vecteurs qui sont dans les situations (b) et (c). Soit a, b, c et d les nombres de paires de différents vecteurs qui appartiennent à des situations (a), (b), (c) et (d), respectivement, et soit $M=n(n\text{-}1)/2$ le nombre de paires de différents vecteurs. Les indices dans le tableau 4, mesure l'agrément conclu entre les deux partitions [73] : le *Rend statistique, coefficient de Jaccard et Folkes Mallow* Le Rend statistique mesure la proportion de paires de vecteurs qui sont d'accords, d'appartenir au même cluster (a) ou aux différents clusters (d) dans les deux partitions.. Le coefficient de Jaccard mesure la proportion des couples qui appartiennent au même cluster (a) dans les deux partitions, par rapport à tous les couples qui appartiennent au même cluster dans au moins l'une des deux partitions $(a+b+c)$. L'indice de Folkes et Mallow (FM) indice mesure la moyenne géométrique de la proportion de couples qui appartiennent au même cluster dans les deux partitions (a), par rapport aux couples qui appartiennent au même cluster, pour chaque partition $((a+b)$ pour φ^A et $(a+c)$ pour $\varphi^B)$.

Indice	**Equation**
Rend statistique	$R = \frac{a+d}{M}$
coefficient de Jaccard	$J = \frac{a}{a+b+c}$
Indice de Folkes et Mallow	$FM = \sqrt{\frac{a}{a+b}\frac{a}{a+c}}$

Tableau V.4 : Indices d'agreement entre les partitions

V.3.3 Le taux de reconnaissance et le taux d'erreurs

Finalement, une méthode plus simple pour déterminer les performances en termes de taux de reconnaissance consiste à présenter au classifieur chacun des exemples de la base et à comparer le cluster donné c^B en résultat à la vraie classe c^A. En considérant que la base contient n objets et que sur ceux-ci $n_{corrects}$ sont biens classés par le système, le taux *de reconnaissance* est simplement défini par:

$$\tau_{reco} = \frac{n_{correct}.100}{\text{n}} \qquad \text{(V. 9)}$$

$$\tau_{err} = \frac{n_{err}.100}{\text{n}} \qquad \text{(V. 10)}$$

Le *taux d'erreur* τ_{err} est défini à partir du nombre d'individus n_{err} mal classés.

V.4 Indices de validation relative

La validation relative est basée sur la mesure de la consistance des algorithmes, en comparant les clusters obtenus par le même algorithme dans des conditions différentes.

V.4.1 Stabilité

La mesure de stabilité a été introduite pour évaluer la validité de la partition trouvée par les algorithmes de clustering et de choisir le nombre de clusters [94] et [108]. La stabilité mesure l'habilité d'un ensemble de données classifié à prévoir le clustering d'un autre ensemble de données échantillonnées à partir de la même source. Nous supposons qu'il existe une partition d'un ensemble S de n objets en K groups, $\varphi = \{c_1, \dots, c_K\}$, et une partition d'un autre ensemble S' de n' objets en K groups $\acute{\varphi} = \{\acute{c}_1, \dots, \acute{c}_K\}$.

Soient les étiquettes α et α' définies par $\alpha(x) = i \; si \; x \in c_i$, pour $x \in S$, et $\alpha'(x) = i$ si $x \in \acute{c}_i$, respectivement. L'ensemble étiqueté (S, α) peut être utilisé pour former un classifieur $f: \Re^n \rightarrow L$, qui induit un étiquetage $\bar{\alpha}$ sur S' par $\bar{\alpha}(x) = f(x)$. La consistance des paires (S, α) et (S', α') est mesurée par la similarité entre l'étiquetage original α et l'étiquetage induit $\bar{\alpha}$ dans S':

$$d_S(\varphi, \varphi')=\min_\pi d_\alpha \left(\alpha', \pi(\bar{\alpha})\right) \qquad (V.11)$$

sur toutes les permutations possibles des étiquettes K' pour ′, avec :

$$d_\alpha(\alpha^1, \alpha^2) = \frac{1}{n' \sum_{x \epsilon S}{}' \delta\left(\alpha^1(x), \alpha^2(x)\right)} \qquad (V.12)$$

Avec $\delta(u, v) = 0$ si $u = v$ et $\delta(u, v) = 1$ si $u \neq v$.

La stabilité d'un algorithme de clustering est définie par l'espérance E de la stabilité pour des paires d'ensembles tirés de la même source:

$$\xi = E_{(S,\varphi)\,(S\prime,\varphi\prime)}[d(\varphi, \varphi')] \qquad (V.13)$$

Dans la pratique, il y'a seulement un ensemble S de points avec lesquels on estime la stabilité d'un algorithme de clustering. Estimation de la stabilité est obtenue par le biais d'un schéma de ré-échantillonnage [94]: l'ensemble S est partitionné en deux sous-ensembles disjoints S_1 et S_2, l'algorithme est appliqué pour obtenir deux partitions, $\varphi_1$et φ_2 , $d(\varphi_1,\varphi_2)$ est calculée et le processus se répète et les valeurs moyennes sont calculées pour obtenir une estimation de ξ. L'indice de stabilité est dépond du nombre de clusters, et doit donc être normalisé lorsqu'il est utilisé pour le modèle de sélection [94] et [108]. La normalisation est obtenue en le divisant par la stabilité obtenue en utilisant un estimateur aléatoire comme classificateur. La sélection de la règle de classification peut influencer la capacité de cet indice pour évaluer la qualité de l'algorithme, car si la règle est aussi simple que de partitionner l'espace de la même manière que l'algorithme de clustering, alors, il peut introduire une instabilité fausse et dégrade l'algorithme [108].

V.5 Expérimentation

Pour estimer les performances de la nouvelle approche des FPCM par rapport aux FCM et aux PCM, nous avons réalisé des tests sur les bases de données : Iris, Texture et la cuisse humaine avec le FCM, PCM et le FPCM.

Base de données Iris : La base de données Iris est constituée de 150 fleurs décrites par 4 variables (longueur et largeur de sépales, et de pétales), le nombre de classes est égal à 3, les objets sont uniformément répartis en trois classes, les classes 2 et 3 sont facilement séparables de la classe 1, mais difficilement séparables entre elles.

Image des textures : La Fig.(V.2) montre une image qui est constituée des deux microtextures différentes. Un prétraitement (le calcul des différentes corrélations locales) de l'image initiale a donné naissance à une série de 8 images dont chacune est le résultat d'une détection d'un attribut particulier. Un pixel est alors décrit par un vecteur à 8 attributs. Ainsi, l'image des deux textures de la Fig.(V.2) est représentée par deux classes dont chacune est décrite par 8 fichiers contenant chacun les valeurs de pixels pour l'une des 8 composantes. L'échantillon obtenu est constitué de 400 pixels de chaque classe.

La cuisse humain : L'image de la Fig.(V.3) est acquise par la photographie couleur de cryosection .Cette image est mise sous le format TIFF. Elle a une taille de 670*415 pixels. Une classification manuelle a été faite par un expert, 4 tissus (classes) ont été identifiés: Graisse, Os, Moelle Muscle. Chaque classe est présentée par 300 pixels, soit 1200 pixels pour les quatre classes. Chaque pixel est caractérisé par 5 attributs ; la composante couleur RVB plus la position géométrique (x, y).

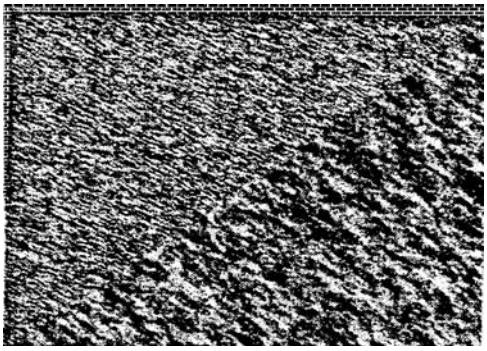


Fig.V.2 : Image des textures.

Fig.V.3 : Image d'une cryosection de cuisses humaines RVBXY.

V.5.1 Classification de la base de données Iris

Les résultats obtenus avec les trois algorithmes des FCM, PCM et des FPCM changent selon le choix de l'initialisation de l'algorithme. On distingue deux méthodes d'initialisation :

- Initialisation par centres de gravité;
- Initialisation par matrice d'appartenance.

V.5.1.1 Initialisation par centres de gravité

L'initialisation par centres de gravité consiste à exécuter plusieurs fois l'algorithme en partant des centres initiaux différents. Lorsqu'on obtient de façon répétée des centres stables d'une répétition à l'autre, alors on peut les considérer comme fiables. En appliquant cette procédure, les centres de gravité qui servent à l'initialisation de chaque algorithme sont les suivants :

$$V_{FCM} = \begin{pmatrix} 5.11 & 5.22 & 1.71 & 1.41 \\ 5.53 & 6.51 & 4.63 & 2.15 \\ 4.80 & 3.53 & 5.40 & 7.50 \end{pmatrix}$$

Les différentes partitions de la base de données Iris générées par les trois algorithmes de classification sont représentées dans la Fig. (V.4).

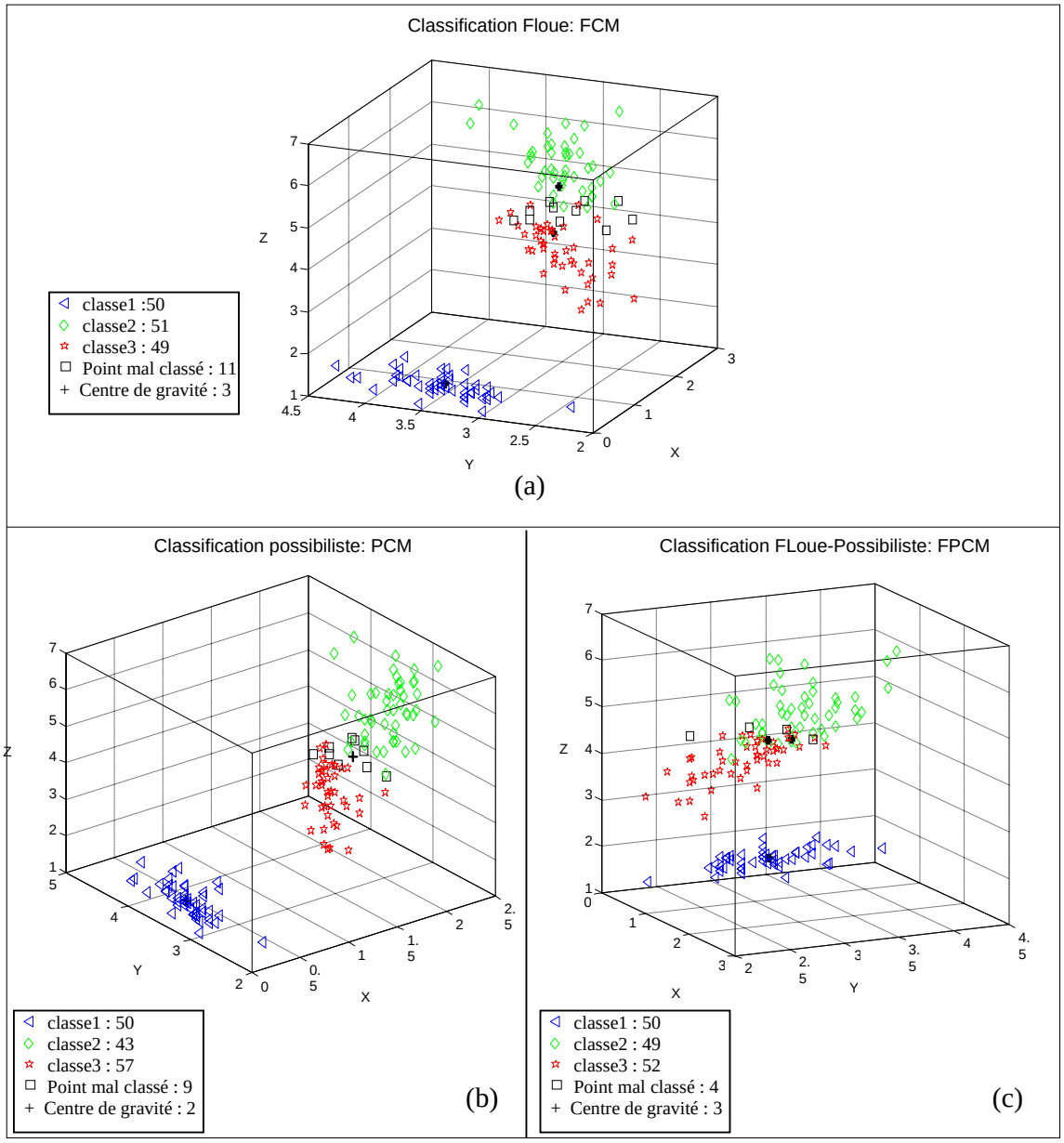


Fig.V.4 : a), b), c) Représentation dans l'espace des attributs des différentes partitions de la base de données Iris générées respectivement par les : FCM, PCM et les FPCM.

Le taux de réussite ainsi que l'indice de satisfaction et le nombre d'itérations correspondants à chaque algorithme de classification sont donnés dans le tableau suivant :

Algorithme de classification	Taux de réussite (%)	Indice de satisfaction (J_m)	Nombre d'itérations pour l'initialisation	Nombre d'itérations Pour la convergence	Nombre d'itérations total
FCM	92.66	$06.91.10^{-8}$	352	22	374
PCM	94.00	$15.10.10^{-1}$	352	42	394
FPCM	97.33	$01.80.10^{-3}$	352	13	365

Tableau V.5 : Comparaison entre les résultats des classifications obtenus avec les FCM, PCM et les FPCM dans le cas d'initialisation par centres de gravité.

À la lecture du tableau (V.5), on remarque que le taux de classification obtenu avec les FPCM est égal à 97.33% après 365 itérations, cependant les PCM atteignent 94.00% après 394 itérations, tandis que les FCM atteignent un taux de 92.00 % pendant 374 itérations.

V.5.1.2 Initialisation par matrice d'appartenance

En initialisant les algorithmes avec la matrice d'appartenance, les résultats de classification de la base de données Iris obtenus sont donnés dans le tableau (V.6).

Algorithme de classification	Taux de réussite (%)	Indice de satisfaction (J_m)	Nombre d'itérations pour l'initialisation	Nombre d'itérations Pour la convergence	Nombre d'itérations total
FCM	92.66	$06.98.10^{-8}$	00	34	34
PCM	94.00	$15.10.10^{-1}$	00	51	51
FPCM	97.33	$01.80.10^{-3}$	00	13	13

Tableau V.6 : Comparaison entre les résultats des classifications obtenus avec les FCM, PCM et les FPCM dans le cas d'initialisation par matrice d'appartenance.

En analysant les résultats du tableau (V.6) , on remarque que les taux et les coûts de la classification obtenus par les trois algorithmes, dans le cas de l'initialisation par matrice d'appartenance, restent inchangeables comparativement au cas de l'initialisation par centres de gravité, cependant le nombre d'itérations a nettement diminué.

a) *Matrice de confusion*

La matrice de confusion correspondante aux résultats de classification pour chaque algorithme, dans le cas d'initialisation par matrice d'appartenance, est donnée par le tableau (V.7).

	C1	C2	C3
C1	50	0	0
C2	0	45	8
C3	0	6	44

FCM
Erreur totale 11

	C1	C2	C3
C1	50	0	0
C2	0	42	8
C3	0	1	49

PCM
Erreur totale 9

	C1	C2	C3
C1	50	0	0
C2	0	47	3
C3	0	1	49

FPCM
Erreur totale 4

Tableau V.7 : Matrices de confusion

Les matrices de confusion qui sont représentées par le tableau (V.7), montrent que les trois algorithmes reconnaissent à 100% la première classe (C1) et commettent des erreurs lors de la classification des objets des deux classes restantes (C2 et C3). L'erreur totale qui correspond respectivement aux FCM, PCM, et aux FPCM est de 11, 9 et 4.

b) Centres finaux des partitions obtenus

Les centres finaux générés par les FCM, PCM et les FPCM, dans le cas de l'initialisation par matrice d'appartenance, sont donnés par le tableau (V.8).

Centres	V1	V2	V3
FCM	[5.02 3.39 1.51 0.25]	[6.02 2.87 4.48 1.46]	[6.49 2.99 5.24 1.89]
PCM	[5.06 3.43 1.46 0.24]	**[6.17 2.88 4.76 1.60]**	**[6.17 2.88 4.76 1.60]**
FPCM	[5.10 3.45 1.45 0.20]	[6.10 2.95 4.65 1.40]	[6.05 3.00 4.85 1.80]

Tableau V.8 : Comparaison entre les centres de gravité obtenus avec les FCM, PCM et les FPCM.

À partir du tableau (V.8), on remarque d'une part la coïncidence des centres des deux dernières classes si on effectue la classification avec les PCM (i.e. V_2=V_3), alors qu'avec les FCM et les FPCM on aboutit à une bonne séparation des centres.

c) Choix du fuzzifieur : m

Les courbes représentées par la Fig.(V.5), illustrent la variation du taux de réussite de la classification et de l'indice de satisfaction en fonction du facteur de fuzzification *m* pour les FCM, PCM et les FPCM avec l'initialisation par matrice d'appartenance.

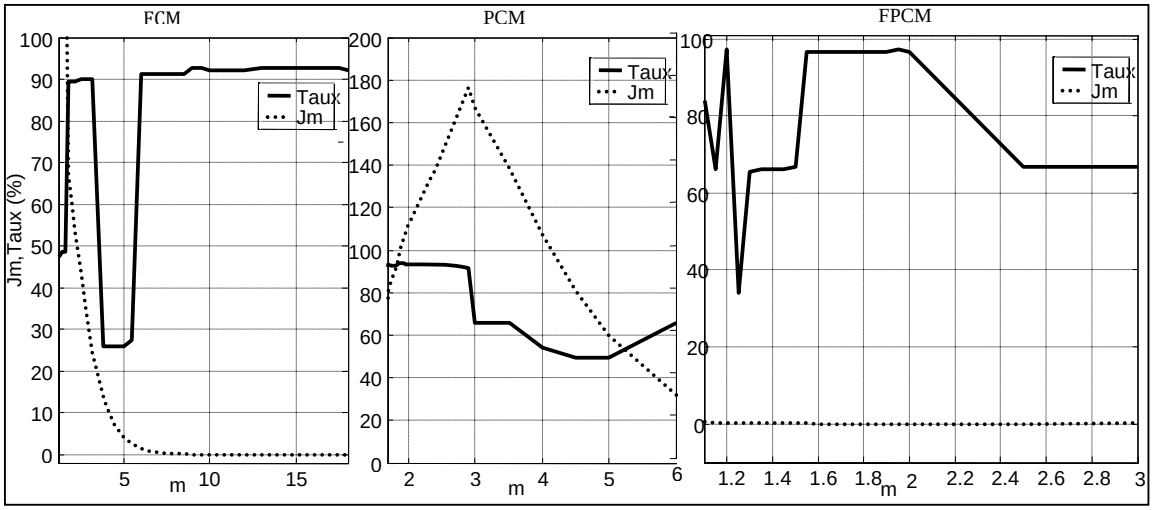


Fig. V.5 : a), b), c) Variation du taux de réussite de la classification et de l'indice de satisfaction en fonction du facteur de fuzzification *m* pour les FCM, PCM et les FPCM.

La valeur de *m* est choisie de telle sorte que le taux de classification soit maximal est le coût soit minimal. En analysant les courbes de la Fig.(V.5), les valeurs de *m* qui correspondent à un indice de satisfaction minimal (J_m) et un taux de réussite maximal pour chaque algorithme de classification sont données dans le tableau (V.9).

Algorithme	m	Taux	J_m
FCM	17.697	92.66	$06.91.10^{-8}$
PCM	1.823	94.00	$15.10.10^{-1}$
FPCM	1.230	97.33	$01.80.10^{-3}$

Tableau V.9: Valeurs optimales de *m* pour les FCM, PCM et les FPCM.

V.5.2 Classification de l'image des textures et de la cuisse humaine

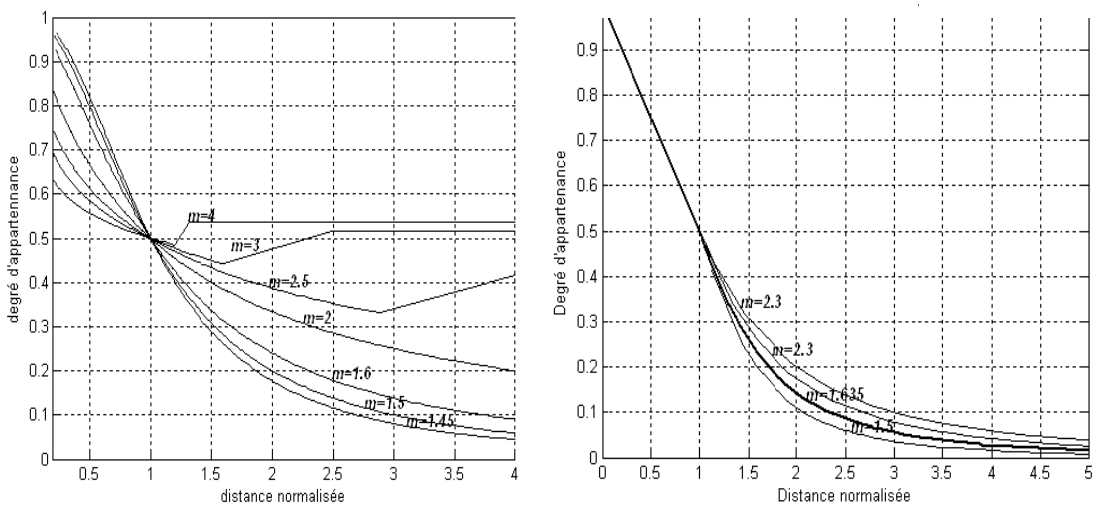
En utilisant l'initialisation par matrice d'appartenance, et en appliquant les différents algorithmes sur les bases de données texture et cuisse humaine, les résultats obtenues sont données dans le tableau (V.10).

Base de données	Algorithme	Taux de réussite (%)	J_m	nombre d'itérations
Image des textures	FCM	99.00	27.0452	14
	PCM	99.50	07.5140	40
	FPCM	99.75	00.1843	09
Cuisse humaine	FCM	88.83	66.40	23
	PCM	85.50	23.46	41
	FPCM	90.83	$4.68.10^{-5}$	18

Tableau V.10 : Résultats de la classification de l'image des textures et de la cuisse humaine obtenus avec les trois algorithmes les FCM, PCM et les FPCM.

On remarque, la supériorité des FPCM en termes de taux de réussite et nombre d'itérations : 99.75%, 09 itérations avec le texture et 90.83% ,18 itérations avec la cuisse humaine.

Les courbes représentées par la Fig. (V.6) illustrent la variation des degrés d'appartenance en fonction de la distance normalisée $\frac{d_{ik}}{\eta_i}$ pour différentes valeurs de *m* pour les PCM et les FPCM.



(a) PCM (b) FPCM

Fig. V.6 : a), b): Variation des degrés d'appartenance en fonction de la distance normalisée $\left(d_{ik}^2/\eta_i\right)$ correspondante à la classification de l'image des textures pour les PCM et les FPCM.

En analysant les courbes de la Fig. (V.6), on remarque que les courbes présentent deux parties différentes : la première partie correspond à des degrés d'appartenance compris entre 0.5 et 1 et la deuxième partie correspond à des degrés compris entre 0.5 et 0. La première partie des courbes correspond à des valeurs de $\left(d_{ik}^2/\eta_i\right)\leq 1$, tandis que la deuxième correspond à des valeurs de $\left(d_{ik}^2/\eta_i\right)>1$. De plus lorsque $\left(d_{ik}^2/\eta_i\right)$ tend vers 0, u_{ik} tend vers 1, et lorsque $\left(d_{ik}^2/\eta_i\right)$ tend vers ∞, u_{ik} tend vers 0. Dans la Fig.(V.6.a), on remarque que, la plus faible valeur de *m* (1.45) est celle qui présente la plus grande valeur de la pente de la tangente de la fonction d'appartenance dans le point (1,0.5). Dans le cas des FPCM, la Fig. (V.6.b) présente des pentes plus élevées (spécifiquement pour *m*=1.5) comparativement à celles obtenues avec les PCM.

Dans la Fig. (V.7.a) et pour une valeur de *m* de 1.45, les PCM atteignent son maximum des taux de réussite (99.5%) et a un coût minimal (7.51). Pour les FPCM, la Fig. (V.6.b) présente une valeur de *m* de 1.65 qui correspond à un taux maximal de 99.75% et à un coût minimal de 0.18.

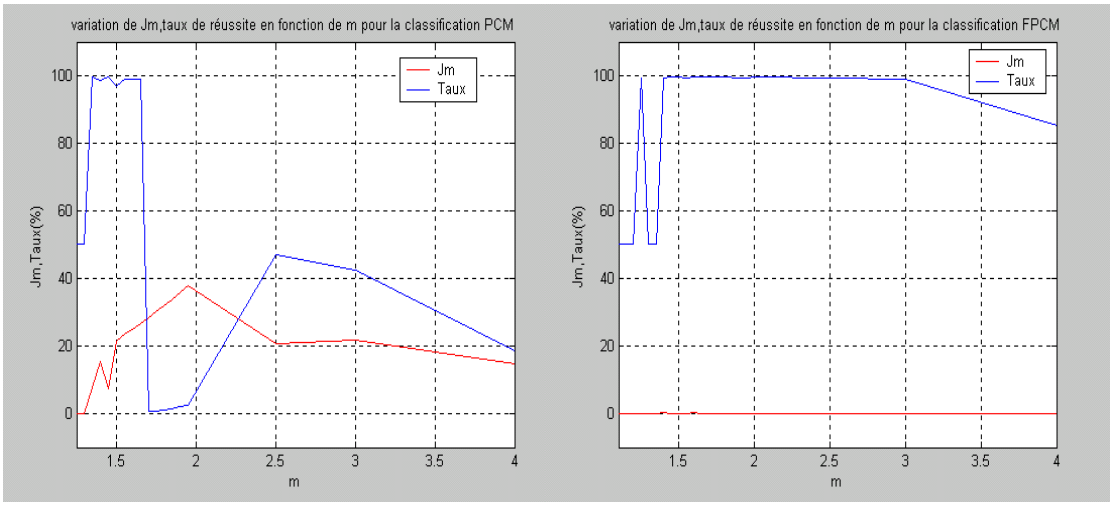


Fig. V.7 a), b): Variation du taux de réussite de la classification et de l'indice de satisfaction en fonction du facteur de fuzzification respectivement pour les PCM et les FPCM lors de la classification de l'image des textures.

À partir des deux figures précédentes, on remarque la compatibilité des résultats obtenus avec les PCM et aussi avec les FPCM. Dans le cas des PCM, une valeur de *m* estimée à 1.45 d'une part, permet, aux points qui sont proches de la première classe d'appartenir à cette dernière avec des degrés forts, alors que ceux qui sont éloignés sont assignés à cette même classe avec des degrés faibles (Fig.V.6.a), et d'autre part, elle correspond au maximum des taux de réussite et au minimum des coûts (Fig.V.6.a). Avec les FPCM, la situation est encore mieux ; une valeur de *m* estimée à 1.63 donne le meilleur taux de réussite (Fig.V.7.b) et pour cette même valeur, la fonction d'appartenance décroît rapidement, ce qui permet aux points situés au voisinage de la première classe d'appartenir à cette dernière avec des degrés plus forts comparativement aux points éloignés qui sont assignés à cette dernière avec des degrés plus faibles (Fig.V.6.b).

V.6 Conclusion

Dans ce chapitre, nous avons présenté trois familles de méthodes de validation : internes, externe et relative. Pendant que les indices internes mesurent la qualité d'un algorithme à générer les groupes intéressants en s'appuyant sur leurs propriétés internes, les indices de validité externes (relatives) comparent les clusters générés avec les vraies classes (les clusters générés) pour déterminer la qualité à reconnaître (à générer) les groupes existants (significatifs). Pour la simulation (ou lorsque l'information additionnelle sur les vraies classes est connue), le choix de l'indice de validité est pour les indices externes. Dans le cas d'absence de cette information, intuitivement, il semble que les indices relatifs devraient être plus désirables que les indices internes puisqu'ils essaient d'exploiter les données d'une manière redondante.

Les résultats de la classification des trois bases de données Iris, texture et cuisse humaine nous ont permis de conclure que les FPCM sont nettement supérieurs aux FCM et aux PCM. Le traitement de ces bases de données par les méthodes classiques qui se base essentiellement sur la détermination d'une partition dure, se trouve confronté à des classes chevauchantes et conduit à une mauvaise classification. Pour surmonter ce type de contrainte, les FCM proposent la notion de partage relatif d'un objet aux différentes classes en respectant la contrainte probabiliste. Ces nouvelles notions apportent des améliorations considérables, mais demeurent sensibles aux bruits, ce qui dégrade les performances du classifieur. Pour résoudre le problème des données bruitées, les PCM proposent la substitution de la notion de partage relatif par celle de typicalité absolue ; dans ce cas, on aboutit à des classes ayant les mêmes centres de gravité. L'algorithme des FPCM qui se base sur les FCM et les PCM, est robuste en présence du bruit, génère des centres séparables, résout le problème de chevauchement et converge rapidement. Malgré que l'initialisation par centres de gravité donne des résultats satisfaisants, elle reste une méthode aléatoire est très coûteuse. La mise en application de cette approche est simple et non coûteuse. L'avantage que présente l'initialisation par matrice d'appartenance est l'élimination du temps de recherche de la partition initiale et par conséquent la réduction du temps de la classification.

Conclusion générale

Dans le cadre de cette thèse, le travail a été principalement articulé autour des points suivants : au départ, on s'est situé à dédier une vue générale du problème de la reconnaissance des formes, et en suite on s'est attaché à établir un état de l'art du clustering pour acquérir les préalables nécessaires au domaine. En pratique, des applications différentes ont des buts différents et peuvent s'accompagner de contraintes qui imposent le choix de la méthode de clustering. À cet effet, les techniques apportées en solution au divers problème de clustering sont nombreuses. Si on se trouve confronté à un problème de reconnaissance des formes incomplètement définies, intuitivement, il semble que les techniques de clustering floues devraient être plus commodes que celles qui sont hard puisqu'elles sont basées sur un concept d'appartenance robuste face aux imprécisions.

Sur le plan pratique, les techniques des c-moyennes hard qui sont basées essentiellement sur la partition dure ont conduit à une mauvaise classification des données en particulier dans le cas où les frontières sont mal définies. Contrairement à l'approche hard, les c-moyennes flous qui sont basés sur la notion de partage probabiliste sont robustes en présence de clusters chevauchants, mais sensibles aux bruits. Pour surmonter le problème du bruit, les approches des c-moyennes possibilistes utilisent la notion de typicalité ; cependant, il génère des centres identiques. Pour contourner les problèmes rencontrés dans le contexte de l'application des c-moyennes, nous avons proposé une nouvelle approche basée sur la fusion des concepts du flou et des possibilités avec une nouvelle méthode d'initialisation des algorithmes. La nouvelle approche permet de résoudre d'une part le problème de chevauchement, de bruit, de la coïncidence et d'autre part d'accélérer le temps de la classification. Toutefois, une étude plus approfondie resterait nécessaire pour l'évaluer par rapport aux autres techniques de clustering flous.

Sur la base de notre analyse de différentes méthodes de clustering, une combinaison de plusieurs techniques peut être la meilleure façon pour remédier aux problèmes rencontrés. En résumé, nous devrions essayer de concevoir un système hybride combinant des modèles multiples pour être applicable aux diverses situations.

Concernant le problème du choix du fuzzifieur, une approche appelée FLVQ propose le lien entre les FCM et une autre méthode de compétition vectorielle (LVQ) permet de trouver une solution au problème du choix de la meilleure valeur de fuzzifieur. Mais cette version reste sensible au bruit. Sur le plan théorique, l'introduction du concept d'appartenance flou-possibiliste permet de rendre

l'algorithme FLVQ robuste en présence du bruit avec maintenance des performances de FLVQ.

Notre travail futur concernera :

– L'adaptation de notre approche pour prendre en compte la différence de volume ou de forme des clusters recherchés, en utilisant, par exemple, des matrices de covariance.

– L'étude des différents indices de validité pour chercher le nombre de clusters optimal.

Annexe A

Notions de similarité

A.I Similarité entre objets

Typiquement, la similarité entre objets est évaluée par une fonction de distance définie entre paire d'objets. En outre, Il est évident que le type des données influence la manière de mesurer le rapprochement potentiel de deux objets. On distingue deux types de mesure de distances :

A.I.1 Cas d'attributs purement quantitatifs

Dans ce cas, on considère des objets dont tous les attributs prennent des valeurs numériques, peu importe que celles-ci soient continues, discrètes ou par intervalles. Les mesures de distance les plus courantes sont les suivantes :

A.I.1.1 Distance de Manhattan ou city-bloc

$$D(X_i, X_j) = \sum_{k=1}^{M} \left| x_{ik} - x_{jk} \right|$$

A partir de cette définition, on peut s'interroger sur la forme des clusters qu'il est possible de détecter dans l'espace des données. Pour cela, considérons le cas simple d'un ensemble d'objets bidimensionnels (c'est-à-dire décrits par deux attributs). Le lieu des points situés à égale distance γ d'un centre (c,d) dans le plan vérifie alors l'équation

$$\left| x - c \right| + \left| y - d \right| = \gamma$$

On montre assez facilement que ce lieu correspond à la représentation de la figure A.1, la diagonale du carré étant égale à 2.

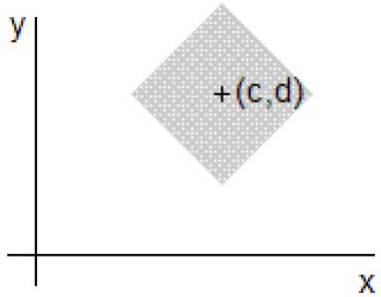


Figure A.1: Visualisation d'un cluster par distance de Manhattan.

A.1.2 Distance Euclidienne

$$D(X_i, X_j) = \sqrt{\sum_{k=1}^{M} \left| x_{ik} - x_{jk} \right|^2}$$

La distance euclidienne correspond à la distance la plus couramment utilisée. Afin de s'intéresser à la forme des clusters détectables par l'utilisation d'une telle distance, on reprend le même exemple que précédemment. On trouve l'équation du lieu des points :

$$\sqrt{(x-c)^2 + (y-d)^2} = \gamma$$

Qui correspond à l'équation d'un cercle de centre (c, d) et de rayon γ dans le plan, représenté à la figure A.2.

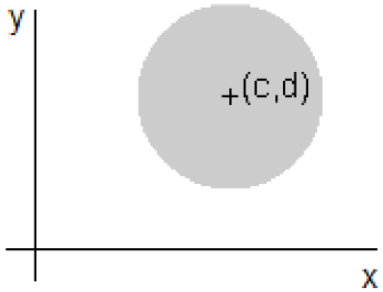


Figure A.2: Visualisation d'un cluster par distance euclidienne

On voit donc que cette distance nous permettra de détecter des clusters ronds dans un espace bidimensionnel, plus généralement des clusters M-sphériques dans un espace à M dimensions.

A.1.3 Distance de Minkowski :

$$D(X_i, X_j) = \left(\sum_{k=1}^{M} \left| x_{ik} - x_{jk} \right|^R \right)^{1/R}$$

Cette distance généralise les deux précédentes. La forme des clusters détectables pour différentes valeurs de R est reprise à la figure A.3, toujours pour un exemple bidimensionnel.

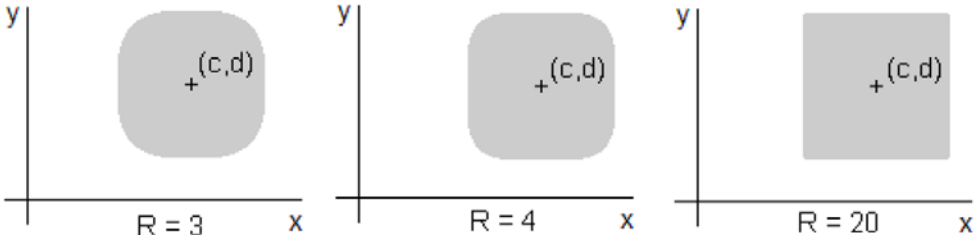


Figure A.3: Visualisation de clusters par distance R

On constate que plus la valeur de R augmente, plus les clusters détectables se rapprochent de carrés parfaits. De manière générale, la distance de Minkowski (et donc les distances de Manhattan, euclidienne, ...) fournissent de bons résultats lorsque les clusters à détecter sont compacts et bien isolés.

A.1.3 Distance de Chebychev :

La distance de Chebychev est la limite de Minkowski pour R tendant vers l'infini.

$$D(X_i, X_j) = \lim_{R\to\infty}\left(\sum_{k=1}^{M}\left|x_{ik} - x_{jk}\right|^R\right)^{1/R}$$

Les clusters détectables correspondent donc à des carrés parfaits.

A.1.4 Cosinus (Ochini coefficient)

$$\cos(X_i, X_j) = \frac{\sum_{k=1}^{M} x_{ik}\, x_{jk}}{\sqrt{\sum_{k=1}^{M} x_{ik}^2}\cdot\sqrt{\sum_{k=1}^{M} x_{jk}^2}} = \frac{\vec{X}_i \,.\, \vec{X}_j}{\left\|\vec{X}_i\right\|.\left\|\vec{X}_j\right\|}$$

Dans ce cas, le cosinus donne une mesure de similarité entre les deux objets [26]. Une mesure de distance peut être simplement obtenue en prenant l'arc cosinus de cet indice de similarité.

A.2 Cas d'attributs qualitatifs (catégoriels)

Dans le cas de données décrites par des attributs catégoriels, la similarité entre deux objets est généralement fonction du nombre d'attributs pour lesquels ils partagent la même catégorie.

A.2.1 Attributs qualitatifs nominaux

Dans ce cas, les attributs de l'objet reprennent typiquement de l'information telle que la couleur d'une voiture, le sexe d'une personne, sa nationalité,... Une mesure courante pour déterminer la similarité entre deux objets décrits par des attributs uniquement catégoriels nominaux.

A.2.1.1 Indice de similarité de Jaccard

Une mesure courante pour déterminer la similarité entre deux objets décrits par des attributs uniquement catégoriels nominaux est l'indice de similarité de Jaccard [26]:

$$Jaccard(X_i, X_j) = \frac{cardinal(attributs\_communs)}{cardinal(attributs)}$$

Cet indice très simple détermine la similarité entre deux objets en effectuant le quotient du nombre d'attributs que les deux objets ont en commun (même modalité) par le nombre d'attributs total décrivant les objets. Pour reprendre l'exemple que nous avons présenté, la similarité entre un français et un belge est donc nulle, si l'on se base uniquement sur la nationalité. En pratique, on sait qu'ils pourront parler la même langue, qu'ils auront les mêmes habitudes culinaires,... Ceci fera donc augmenter leur similarité. On constate rapidement qu'on ne peut pas utiliser l'indice de Jaccard pour des attributs catégoriels ordonnés. En effet, aucune insertion de nuance n'est prévue dans le calcul, c'est-à-dire que le résultat de la mesure est en quelque sorte binaire. Par exemple, pour un attribut donné, les mentions « bien » et « satisfaisant » sont jugées équivalentes par rapport à la mention « très bien »: la similarité est nulle dans les deux cas.

A.2.1.2 Conversion en strings de bits

Lorsque l'on dispose d'attributs qualitatifs nominaux pour mesurer le rapprochement de deux objets, on peut aussi convertir les valeurs non numériques des attributs en strings de bits. Pour cela, chaque attribut possède un certain nombre de modalités (possibilités) qui détermine la longueur de la chaîne sur laquelle on va coder l'attribut. Par exemple, un attribut pouvant prendre les couleurs «rouge »,« bleu »,« jaune » et « vert » sera codé sur quatre bits. Ainsi, un objet de couleur bleue serait codé selon 0100. On voit donc qu'il n'y aura dans chaque code d'attribut qu'un seul bit à 1, correspondant à la modalité caractérisant l'objet pour cet attribut. Ensuite, toutes ces valeurs codées (différents attributs) sont mises bout à bout en chaîne. Un objet sera donc maintenant représenté par un mot binaire dont la longueur sera égale à la somme du nombre de modalités sur tous les attributs.

Sur base de ce codage, une première option consiste à mesurer directement une dissimilarité. On peut en effet caractériser la dissimilarité entre deux objets à partir du nombre de bits discordants dans leurs chaînes respectives. Le tableau ci-dessous fournit un exemple de ce type de démarche. La dernière ligne du tableau correspond à un mot binaire dont les bits valent 1 lorsque les bits associés sont discordants dans les deux chaînes comparées et 0 si ces bits sont concordants.

z_{ik}	0100111010111001
z_{jk}	1001101111011101
	1101010101100100

On définit alors l'indice de dissimilarité selon [38] :

$$d(X_i, X_j) = \frac{1}{Z}\sqrt{\sum_{k=1}^{Z}(z_{ik} - z_{jk})^2}$$

Où Z représente le nombre de bits dans les chaînes manipulées (somme des nombres de modalités sur tous les attributs). On a ajouté ici la division par Z dans la formule pour satisfaire la convention adoptée selon laquelle un indice de

dissimilarité a une valeur comprise entre 0 et 1. On constate que cette méthode ne peut pas non plus être utilisée pour des attributs ordonnés. En effet, on remarque facilement que la valeur de l'indice est égale à la racine carrée (divisée par *Z*) du double du nombre de variables qualitatives initiales (attributs) qui n'ont pas la même modalité entre les deux objets. Il s'agit donc à nouveau d'une comparaison « en tout ou rien » au niveau d'un attribut.

Une autre option, plus flexible et courante, consiste à s'intéresser à des mesures de similarités et/ou de dissimilarités « nuancées ». Dans ce cas, on s'attache à observer les concordances et discordances dans les chaînes de bits traitées, en faisant la nuance entre les concordances 1-1 et 0-0 et entre les discordances 0-1 et 1-0. Ceci est représenté au tableau ci-dessous, tableau des occurrences des concordances et discordances entre deux objets.

	Objet *i*		
		0	1
Objet *j*	0	a	b
	1	c	d

Sur base de ce tableau, on peut définir diverses mesures de (dis) similarité. Seuls quelques indices de similarités relativement simples sont présentés ci-dessous, à titre d'exemples :

- Pourcentage de concordances [38] :

$$s = \frac{a+d}{Z}$$

- Concordances 1-1 et Concordances 0-0 [38] :

$$s = \frac{d}{Z} \qquad \text{et} \qquad s = \frac{a}{Z}$$

Notons que l'on peut « traduire » l'indice de similarité de Jaccard à partir de ce formalisme. En effet, on remarque assez facilement que :

$$Jaccard(X_i, X_j) = \frac{d}{cardinal(attributs)}$$

Remarquons encore que de manière générale, la notion de centroïde est remplacée par celle de médoïde lorsque l'on travaille avec des chaînes de bits. Ceci est une conséquence directe du fait que la moyenne entre deux valeurs binaires différentes n'existe pas toujours en binaire. Ainsi, la moyenne entre 0 et 1 n'existe pas, de même que celle entre 001 et 010, … Le médoïde est alors défini comme le string de bits qui minimise la somme des écarts à tous les objets du groupe.

Enfin, d'autres types de mesures peuvent encore être appliqués :

- Mesure du cosinus (Ochini coefficient) : cette mesure s'applique en effet aussi bien à des attributs numériques que non numériques. Le principe est ici de prendre le cosinus de l'angle défini par les deux vecteurs binaires à Z composantes. On peut bien évidemment toujours passer à une mesure de distance en prenant l'arccosinus de l'indice obtenu.
- Distance $\chi 2$: cette distance vise à diminuer naturellement la contribution d'une modalité k (ou bit) si sa fréquence d'observation est élevée, puisqu'elle n'est alors pas déterminante pour la formation des clusters. Les modalités rares ont donc un impact élevé.

$$D_{\chi^2}(X_i, X_j) = \sqrt{\sum_{k=1}^{Z} \frac{n}{n_k} \left(\frac{z_{ik} - z_{jk}}{M}\right)^2}$$

Où M correspond au nombre de variables qualitatives (attributs), n est le nombre d'objets et n_k est le nombre d'objets ayant la modalité k.

A.2.2 Attributs qualitatifs ordonnés

Dans ce cas, les attributs de l'objet reprennent typiquement de l'information telle que le rang militaire d'un officier, une appréciation, ...

Considérons, par exemple, un attribut pouvant prendre les modalités suivantes : « faible », « moyen » et « bon ». Pour garder leur caractère ordonné, la technique la plus courante consiste à convertir les attributs en valeurs numériques. Ainsi, on associe des valeurs numériques aux échelons de l'échelle ordinale. Il existe une multitude de possibilités pour le codage, par exemple :

Modalités	Codage 1	Codage 2	Codage 3
FAIBLE	1	1	1
MOYEN	2	3	7
BON	3	5	10

Les codages 1 et 2 présentent la particularité que les modalités sont séparées par un écart constant en valeurs numériques. Par contre, le codage 3 est fondamentalement différent. On supposera ici que les échelles qualitatives sont « bien faites » en ce sens que des modalités successives sont équidistantes en termes de préférences de l'utilisateur. Ainsi, on pourra procéder facilement à la conversion des attributs en valeurs numériques (sans nécessiter d'information supplémentaire).

Néanmoins, cette manière de procéder est discutable. En effet, ces codages ne tiennent pas compte de l'aspect qualitatif des données de départ, ils dénaturent les données. A partir du moment où l'utilisateur ne faxerait pas forcément les modalités équitablement, le codage reviendrait à devoir demander l'information directement en valeurs numériques (classes, intervalles). De plus, des problèmes

peuvent se poser en termes d'interprétations : selon le codage 2, par exemple, que signifie le fait que « bon » soit jugé comme 5 fois meilleur que « faible »?

Pourtant, ces transformations des modalités en valeurs numériques sont d'usage fréquent pour permettre de calculer des distances, similarités ou dissimilarités. Dans le cadre de problèmes multicritères où il n'est pas nécessaire de définir de mesures de distances (ou autre), on peut envisager d'autres techniques [59].

A.3 Cas d'attributs mixtes

Il est bien évidemment rare qu'un problème ne comprenne que des attributs quantitatifs ou que des attributs qualitatifs. La plupart du temps, les données sont mixtes et il est donc nécessaire de généraliser les considérations précédentes. Plusieurs approches sont possibles : une première méthode consiste à convertir les variables catégorielles en variables numériques, ou inversement, on peut procéder à une conversion des variables numériques en variables catégorielles par la découpe préalable de l'intervalle de variation en sous-intervalles correspondant alors aux différentes modalités. Finalement, une approche consiste à traiter chaque variable selon son type et à rassembler tous les résultats de comparaisons dans une même mesure commune. Par exemple, un modèle de dissimilarité pourrait se construire selon [38]:

$$d^2(X_i, X_j) = \frac{1}{P}\sum_{k=1}^{P}\sigma_k(X_i, X_j)$$

$$\text{Avec} \begin{cases} \sigma_k(X_i, X_j) = \sigma_k(X_j, X_i) \\ 0 \le \sigma_k(X_i, X_j) \le 1 \\ \sigma_k(X_i, X_j) = 0 \Leftrightarrow x_{ik} = x_{jk} \end{cases}$$

Où : $\sigma_k(X_i, X_j)$ représente la contribution de la K^e variable parmi les P variables.

On peut brièvement discuter sur le type de variable :

- pour des variables qualitatives nominales, on peut se baser sur la notion exploitée par l'indice de Jaccard (transposée en indice de dissimilarité ici), c'est-à-dire que :

$$\sigma_k(X_i, X_j) = \begin{cases} 0 & si \quad x_{ik} = x_{jk} \\ 1 & si.non \end{cases}$$

- pour des variables qualitatives ordonnées, on peut convertir les données en valeurs numériques et ensuite appliquer tout indice de dissimilarité pour des attributs numériques.
- pour des variables quantitatives, on peut se baser directement sur tout indice de dissimilarité pour des attributs numériques.

A.II Similarité de deux clusters

Contrairement à la similarité entre objets, la similarité de deux clusters, ne nécessite pas de calcule complexe et ne fait intervenir que des concepts physiques. Supposons avoir choisi une manière de mesurer le rapprochement de deux objets (peu importe laquelle). Le problème de déterminer celui de deux clusters revient alors simplement à déterminer quels objets prendre dans chacun des clusters pour définir la mesure. Les possibilités les plus courantes sont reprises ci-dessous.

A.II.1 Lien simple (SLINK)

Cette approche est encore nommée « *nearest neighbor approach* », ce qui traduit peut-être mieux son principe. Il s'agit donc de définir la distance entre deux clusters comme étant la plus petite distance parmi celles entre toutes les paires d'objets entre les deux clusters. Mathématiquement, la distance entre le cluster C_p et le cluster C_q est la plus petite distance entre un élément de C_p et un élément de C_q :

$$D(C_p, C_q) = \min\{dist(X_i, X_j), X_i \in C_p, X_j \in C_q\}$$

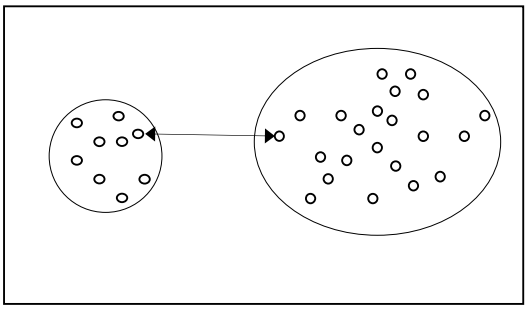


Fig. A.4: Distance entre deux clusters par SLINK

A.II.2 Lien complet (CLINK)

Cette approche est encore nommée « *farthest neighbor approach* », ce qui traduit peut-être mieux son principe. Il s'agit donc de définir la distance entre deux clusters comme étant la plus grande distance parmi celles entre toutes les paires d'objets entre les deux clusters. Mathématiquement, la distance entre le cluster C_P et le cluster C_q est la plus grande distance entre un élément de C_P et un élément de C_q :

$$D(C_p, C_q) = \max\{dist(X_i, X_j), X_i \in C_p, X_j \in C_q\}$$

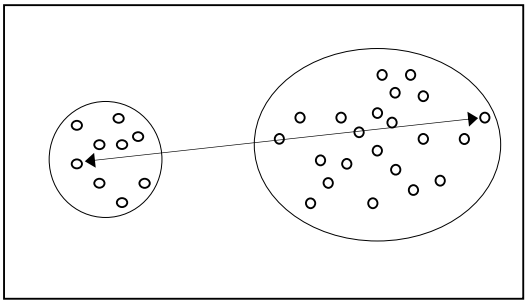


Fig. A.5 : Distance entre deux clusters par CLINK

A.II.3 Lien moyen (ALINK)

Cette approche définit la distance entre deux clusters en faisant intervenir tous les objets présents dans ces clusters. Mathématiquement, la distance entre le cluster C_P et le cluster C_q est la moyenne des distances entre un élément de C_P et un élément de C_q :

$$D(C_p, C_q) = \frac{\sum_i \sum_j \{dist(X_i, X_j), X_i \in C_p, X_j \in C_q\}}{cardinal(C_p).cardinal(C_p)}$$

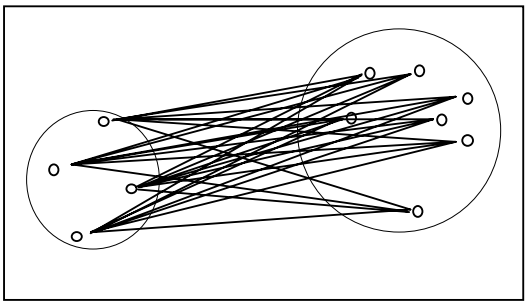


Fig. A.6: Distance entre deux clusters par ALINK

A.II.4 Lien moyen de groupe (GALINK)

Cette approche est encore nommée «méthode des centroïdes», ce qui traduit peut-être mieux son principe. Il s'agit donc de définir la distance entre deux clusters comme étant la distance entre les centroïdes de ces clusters. Mathématiquement, si G_P est le centroïde du cluster C_p et si G_p est le centroïde du cluster C_p alors la distance entre le cluster C_p et le cluster C_p est la distance entre leurs centroïdes :

$$D(C_p, C_p) = dist(GP, Gq)$$

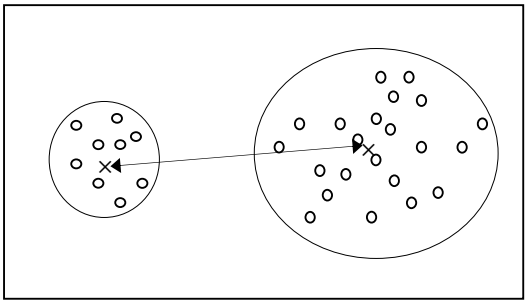


Fig. A.7: Distance entre deux clusters par GALINK

Remarquons qu'il n'est pas toujours nécessaire d'avoir recours à des mesures de rapprochements de clusters. Comme nous le verrons au chapitre suivant, ceci est surtout utile dans les algorithmes de clustering hiérarchique ascendant.

A.III Rapprochement d'un objet et d'un cluster

Si l'on suppose avoir défini une mesure du rapprochement de deux objets, la définition de celui d'un objet et d'un cluster est presque immédiate puisqu'il suffit de choisir un point représentatif du cluster. On se ramène donc à la mesure du rapprochement de deux objets : l'objet en question et le représentatif du cluster, pouvant être son centroïde (ou médoïde), l'objet le plus proche, l'objet le plus loin..

Annexe B

B.1 Modification de l'espace de description pour la prise en compte du contexte

Afin de prendre en compte le fait que certains attributs décrivant les données ont plus au moins d'importance pour la classification, une première possibilité consiste à utiliser une méthode de sélection, d'extraction ou de pondération d'attributs en prétraitement de l'apprentissage. La sélection d'attributs est le processus qui consiste à identifier le sous-ensemble des attributs le plus efficace à utiliser pour le clustering. L'extraction d'attributs correspond à l'utilisation d'une ou plusieurs transformations des attributs initiaux pour produire de nouveaux attributs pertinents. Et la pondération d'attributs consiste à donner un poids plus important à certains attributs considérés comme plus pertinents pour le clustering.

a. Sélection d'attributs

Il existe trois grandes techniques pour sélectionner le sous-ensemble des attributs le plus approprié pour le clustering : exhaustive, aléatoire ou heuristique. Étant donné un critère d'intérêt d'un sous-ensemble d'attributs pour effectuer le clustering d'un ensemble d'objets, il s'agit dans le premier cas de considérer l'ensemble des sous-ensembles d'attributs possibles et de sélectionner celui qui optimise ce critère. Une telle approche n'est cependant pas utilisable en pratique à cause de sa complexité exponentielle $O(2^p)$ en fonction du nombre de dimensions p. Une approche aléatoire consiste à sélectionner aléatoirement puis évaluer différents sous-ensembles d'attributs, son avantage étant alors d'être capable de fournir un résultat à n'importe quel moment de son exécution. Enfin, des heuristiques souvent utilisées en sélection d'attributs consistent à effectuer une sélection ascendante ou descendante. En sélection ascendante, l'attribut considéré comme le plus pertinent selon le critère utilisé et parmi l'ensemble des attributs de la base est d'abord sélectionné, puis le second attribut le plus pertinent est sélectionné, et ainsi de suite. L'opération inverse est exécutée en sélection descendante : l'ensemble des dimensions de description est initialement considéré, puis à chaque étape, la dimension évaluée comme la moins pertinente est supprimée. Dans [33], le critère de la pertinence d'un sous-ensemble d'attributs pour le clustering est basé sur l'observation qu'un ensemble de données contenant des clusters a un histogramme des distances entre objets très différent par rapport à celui d'un ensemble de données ne contenant pas de clusters. En effet, si un ensemble de données peut être partitionné, alors la majorité des distances intra-clusters sera bien inférieure à la majorité des distances inter-clusters. Au contraire, si les données sont uniformément réparties, alors les distances entre objets le seront également. La mesure proposée est dérivée de la mesure d'entropie, qui

représente typiquement le taux de désordre d'une configuration, *D* dénotant la distance normalisée entre les objets et dans le sous-espace considéré :

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N}[D_{ij} \times \log D_{ij} + (1 - D_{ij}) \times \log(1 - D_{ij})]$$

Une autre mesure basée sur la *Category Utility* a également été proposée dans [116] pour évaluer la pertinence d'attributs catégoriels, l'hypothèse sous-jacente à cette mesure étant de considérer que si une dimension n'est pas hautement corrélée aux autres, alors cette dimension ne jouera pas un rôle important dans le processus de clustering, et peut dès lors être considérée comme peu pertinente :

$$\frac{\sum_{l=1}^{M}\sum_{m\in Modalites_l} P(x_{il} = m)\sum_{m2\in Modalites_d}\left[P(x_{id} = m2 / x_{il} = m)^2 - P(x_{id} = m2)^2\right]}{M - 1}$$

b. <u>Extraction d'attributs</u>

La technique d'extraction d'attributs la plus utilisée est sûrement *l'Analyse en Composantes Principales* (ACP) [82]. Cette technique consiste à définir un ensemble de *m* nouvelles variables, combinaison linéaire des variables de l'espace initial, qui feraient perdre le moins d'information possible. Ces *m* variables sont appelées *composantes principales* et les axes qu'elles déterminent *axes principaux.*

Il s'agit pour cela d'analyser la structure de la matrice variance-covariance, qui rend compte de la variabilité, de la dispersion des données, l'objectif de l'ACP étant de décrire à l'aide d'un minimum de composantes un maximum de cette variabilité. Cette technique permet alors de réduire efficacement la taille des données considérées. Elle peut aussi être utilisée afin de visualiser les données en deux ou trois dimensions. La première étape de la méthode consiste à standardiser les données, c'est-à-dire les centrer et les réduire. Ensuite, il s'agit de constituer la matrice de corrélation entre les variables, puis de trouver les vecteurs propres de cette matrice, ainsi que leur valeur propre associée. Les vecteurs propres donnent alors les axes factoriels. Ils sont déterminés de façon à rendre compte le mieux possible de la dispersion des données présentes dans la matrice. Les valeurs propres, quant à elles, sont proportionnelles à la variance associée à ces axes. La dernière étape de la méthode consiste finalement à calculer les coordonnées des individus sur les nouveaux axes sélectionnés, c'est-à-dire les *m* vecteurs propres dont les valeurs propres associées sont maximales.

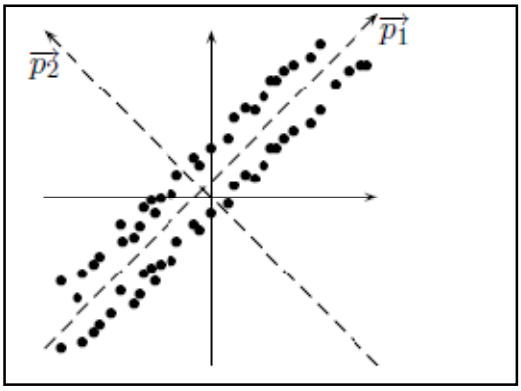


Figure B.1 : Exemple d'Analyse en Composantes Principales

La figure B.1 montre sur un exemple les deux axes principaux et *p* calculés à partir d'un ensemble d'objets définis sur deux dimensions.

c. Pondération d'attributs

Dans [64], les auteurs proposent d'assigner à chaque objet un poids sur les dimensions de l'espace. Basée sur l'utilisation des plus proches voisins, la technique est la suivante :

1. démarrer avec toutes les dimensions également pondérées pour chaque objet.
2. cibler les k plus proches voisins de chaque objet; plus la dispersion du groupe des k plus proches voisins sur une dimension est faible plus le poids assigné à cette dimension est important;
3. ces poids sont -ensuite utilisés pour calculer les nouvelles distances, qui sont
4. Réutilisées pour cibler les k nouveaux plus proches voisins;
5. et le processus est ainsi répété jusqu'à ce que les poids se stabilisent.

L'algorithme fournit alors en sortie une matrice de distances entre objets, qui peut être utilisée par n'importe quel algorithme de clustering, et permettra ainsi de refléter le fait que les dimensions à prendre en compte pour les calculs de distance entre objets peuvent varier d'un couple à l'autre.

B.2 Solution pour traiter le problème des grandes bases de données

Pour être applicable à des bases de données contenant un grand nombre d'objets ou d'attributs, les méthodes de clustering développées doivent porter une attention particulière à leur complexité. Typiquement, la meilleure solution est de développer des méthodes qui dépendent de façon linéaire du nombre d'objets et d'attributs. D'autres solutions peuvent également être envisagées :

1. Minimiser le nombre de parcours de l'ensemble des données;

2. Réduire le nombre de données examinées pendant l'exécution;
3. Réduire la taille de la structure des données fournie à l'algorithme.

Plus concrètement, les solutions les plus souvent utilisées dans ce cadre sont les suivantes [76] :

1. L'échantillonnage aléatoire permet de sélectionner un sous-ensemble représentatif des données, permettant ainsi de considérer ensuite moins d'objets lors du déroulement de la méthode;
2. L'utilisation de bornes permet de fixer un nombre maximum d'itérations de la méthode, lorsque celle-ci est capable de fournir une solution à chaque instant, comme c'est le cas pour les méthodes stochastiques ou statistiques;
3. Le partitionnement permet de diviser le problème en plusieurs sous problèmes de plus petite taille, puis d'utiliser ensuite les résultats de ces sous problèmes pour résoudre le problème général;
4. Et la transformation des données permet de réduire la taille des données fournies en entrée de la méthode, afin de travailler ensuite sur une base de données moins importante : il s'agit dans ce cas de transformer les données sous une représentation plus compacte, ou bien de se placer dans un espace plus réduit que l'espace original; ce dernier point sera détaillé dans le chapitre suivante.

Annexe C

Caractéristiques des méthodes de clustering

Caractéristique	**Hiérarchique**	**K-means**	**Statistique**	**Stochastique**	**Basé sur la densité**	**Basé sur la grille**	**Basé sur les graphes**
connaissances a priori	Nombre de clusters ou seuils	Nombre de clusters K	Nombre de clusters c	nombre de clusters c	critère de densité du voisinage	taille de grille et critère de densité	distance minimale entre clusters
Présentation des résultats	hiérarchie	K centroïdes	c centroïdes et matrices de covariances	partition	partition	ensembles de cellules connectées	partition sous forme de graphe
complexité	$O(p.n^2)$	$O(p\ .n.\ k)$	$O(p^2\ .n.\ c)$	$O(p.\ n.\ c)$	$O(p.n^2)$	$O(p.$ *taille de grille*$)$	$O(p.n^2 \log n)$
déterministe	oui	non	non	non	oui	oui	oui
incrémental	non	oui	oui	oui	non	non	non
any-time	non	oui	oui	oui	non	non	non
hard	oui	oui	non	oui	oui	oui	oui
prise en compte du contexte	non	non	non	non	non	non	non
tolérance au bruit	non	non	oui	oui	oui	oui	non
tolérance à l'effet de chaîne	non	oui	oui	oui	non	non	non
tolérance aux clusters de tailles variées	oui	non	oui	oui	oui	oui	oui
tolérance aux clusters de densités variées	oui	oui	oui	oui	non	non	oui
tolérance aux clusters de forme quelconque	oui	non	non	non	oui	oui	oui
tolérance aux clusters concentriques	oui	non	non	non	oui	oui	oui

Tableau C.1 : Caractéristiques associées aux méthodes de clustering classiques.

Bibliographie

[1] S. Aksoy et R. M. Haralick, Probabilistic vs. geometric similarity measures for image retrieval. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition,* volume 2, pp. 357- 362, 2000.

[2] F. Azuaje, A cluster validity framework for genome expression data, *Bioinformatics* 18 (2002), pp. 319–320.

[3] F. Azuaje, N. Bolshakova, Clustering genomic expression data, in: D. Berrar, W. Dubitzky, M. Granzow (Eds.), Design and Evaluation Principles, A Practical Approach to Microarray Data Analysis, Copyright 2002, Kluwer Academic Publishers, Boston, Dordrecht, London.

[4] G.P. Babu and M.N. Murty, Clustering with evolutionary strategies, *Pattern Recognition* 27 (1994), pp. 321–329.

[5] R. Babuska, P.J. Van der Veen, U. Kaymak, Improved covariance estimation for Gustafson–Kessel clustering. in: Proc. 2002 *IEEE* Internat. Conf. on Fuzzy Systems, vol. 2, Honolulu, HI, 2002, pp. 1081–1085.

[6] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data", *Behavioral Science,* vol.12, pp. 153-155, 1967.

[7] M.Barni, V. Cappellini, end A. Mecocci, "Comments on 'A Possibilistic Approach to Clustering,'"*IEEE* Conf, Fuzzy Syst., Orlando, FL, july 1994, pp.902-908.

[8] M. Barni, V. Cappellini and A. Mecocci, Comments on a possibilistic approach to clustering, *IEEE Trans. Fuzzy Systems* 4 (1996), pp. 393–396.

[9] N. Beck, Application de méthodes de clustering traditionnelles et extension au cadre multicritère, mémoire d'Ingénieur, Université libre de Bruxelle, 2006.

[10] A. Ben-Dor, R. Shamir and Z. Yakhini, Clustering gene expression patterns, *J. Comput. Biol.* 6 (1999) (3/4), pp. 281–297.

[11] P. Bertier et J.M Bouroche, *Analyse des Données Multidimensionnelles,* Presses Universitaires de France, 1975.

[12] J.C. Bezdek, Fuzzy mathematics in pattern classification. Ph.D. Thesis, Applied Mathematics Center, Cornell University, Ithaca.

[13] J.C. Bezdek, A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms, *IEEE* Trans. Pattern Anal. Machine Intell., vol. PAMI-2, no. 1, pp. 1-8, January 1980.

[14] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York (1981).

[15] J.C. Bezdek and Pal S. K., eds. (1992) *Fuzzy Models for Pattern Recognition Methods that Search for Strucure in Data, IEEE* Press, New York.

[16] J.C. Bezdek, Numerical taxonomy with fuzzy sets, *Journal of Mathematical Biology,* 1, pp. 57-71, 1974.

[17] J.C. Bezdek, J. Keller, R. Krisanpuram, N.R. Pal, Fuzzy Models and Algorithms for Pattern Recognition and Image Processing, Kluwer, Norwell (USA), 1999.

[18] J.C. Bezdek, J. Keller, R. Krishnapuram and N.R. Pal, Fuzzy Models and Algorithms for Pattern Recognition and Image Processing, Kluwer, Boston, London (1999).

[19] J.C. Bezdek, J. Keller, R. Krishnapuram and N.R. Pal, Cluster analysis for relational data, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer, Boston, London (1999), pp. 137–182 (Chapter 3).

[20] H.H. Bock, Automatische Klassifikation, Vadenhoeck & Ruprecht, Göttingen, Zürich (1974).

[21] C. Borgelt, R. Kruse, Fuzzy and probabilistic clustering with shape and size constraints. In: Proceedings of the 11th International Fuzzy Systems Association World Congress, IFSA'05, Beijing, China (2005), pp. 945–950.

[22] N. Bolshakova, F. Azuaje, Cluster validation techniques for genome expression data, Technical Report TCD-CS-2002-33, Computer Science Department, The University of Dublin.

[23] H. Boudouda, M. Nemissi, H. Seridi and H. Akdag, Fuzzy-Possibilistic Classification: Resolution of Initialization Problem, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 13, N°1, pp. 45-51, Fuji Technology Press Ltd., Japan, 2009, ISSN-0130,

[24] J-M Bouroche et G. Saporta, *L'analyse des données,* Presses Universitaires de France, sixième édition, Décembre 1994.

[25] E.R. Dougherty, U. Braga-Neto, Epistemology of computational biology: mathematical models and experimental prediction as the basis of their validity, Biol. Syst. 14(1) (2006) 65–90.

[26] L. Candillier, La classification non supervisée, Equipe GRAppA, Lille 3, 2004.

[27] L. Candillier, Contextualisation, Visualisation et Évaluation en Apprentissage Non Supervisé, thèse de doctorat à l'Université Charles de Gaulle - Lille 3, 2006.

[28] Francisco de A.T. de Carvalho et al. , Partitional fuzzy clustering methods based on adaptive quadratic distances, Fuzzy Sets and Systems 157 (2006) 2833 – 2857

[29] G. Celeux, and G. Govaert, A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis,* 14(1992): 315 – 332.

[30] S. Chatzis and T. Varvarigou, Factor Analysis Latent Subspace Modeling and Robust Fuzzy Clustering Using t-Distributions, *IEEE Trans.*, Fuzzy Systems,Volume: 17, page(s) : 505-517, 2009.

[31] R. Coppi and P. D'Urso, Three-way fuzzy clustering models for lr fuzzy time trajectories, *Comput. Statist. Data Anal.* 43 (2003) (2), pp. 149–177.

[32] R. Coppi and P. D'Urso, Fuzzy unsupervised classification of multivariate time trajectories with the Shannon entropy regularization, *Comput. Statist. Data Anal.* 50 (2006) (6), pp. 1452–1477.

[33] M. Dash, K. Choi, P. Scheuermann, and H. Liu, Feature selection for clustering – *a* filter solution. *IEEE International Conference on Data Mining,* (2002) pages 115–122.

[34] R.N. Davé and R. Krishnapuram, Robust clustering methods: a unified view, *IEEE Trans. Fuzzy Systems* 5 (1997), pp. 270–293.

[35] R. Davé, Characterization and detection of noise in clustering, *Pattern Recognition Lett.* 12 (1991), pp. 657–664.

[36] R. Davé and S. Sen, On generalising the noise clustering algorithms, *Proceedings of the Seventh IFSA World Congress, IFSA'97* (1997), pp. 205–210.

[37] R. Davé and S. Sen, Generalized noise clustering as a robust fuzzy c-m-estimators model, *Proceedings of the 17th Annual Conference of the North*

American Fuzzy Information Processing Society: NAFIPS'98 (1998), pp. 256–260.

[38] C. Decaestecker, M. Saerens, *Clustering,* Unité de Systèmes d'Information, Université Catholique de Louvain-la-Neuve, 2005.

[39] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion), *J. Roy. Statist. Soc. Ser. B* 39 (1977), pp. 1–38.

[40] Y. De-smet, C. Lamboray, Multicriteria clustering : a few approaches, Service de Mathématiques de la Gestion, Université Libre de Bruxelles, 2005.

[41] E. Diday, La méthode des nuées dynamiques, *Revue de Statistique Appliquée,* vol. 19, n. 2, pp. 19-34, 1971.

[42] E. Diday, *Nouvelles méthodes et nouveaux concepts en classification automatique et reconnaissance des formes,* thèse de doctorat d'Etat, Université de Paris VI, 1972.

[43] E. Diday, Optimisation en Classification Automatique et Reconnaissances des formes, *Rev, Fr. Inf. Rech. Opér.,* 6e année, pp. 61-95, novembre 1972.

[44] C. Döring, C. Borgelt and R. Kruse, Effects of irrelevant attributes in fuzzy clustering, *Proceedings of the 14th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2005, Reno, Nevada, USA*, *IEEE* Press, Piscataway, NJ, USA (2005), pp. 862–866.

[45] C. Döring, M-J. Lesot, and R. Kruse, Data analysis with fuzzy clustering methods, Computational Statistics & Data Analysis 51 (2006) 192 – 214.

[46] E.R. Dougherty, J. Barrera, M. Brun, S. Kim, R.M. Cesar, Y. Chen, M.L. Bittner and J.M. Trent, Inference from clustering with application to gene-expression microarray, *J. Comput. Biol.* 9 (2002) (1), pp. 105–126.

[47] E.R. Dougherty, M. Brun, A probabilistic theory of clustering, Pattern Recognition 37 (2004) 917–925.

[48] E.R. Dougherty, U. Braga-Neto, Epistemology of computational biology: mathematical models and experimental prediction as the basis of their validity, Biol. Syst. 14(1) (2006) 65–90.

[49] D. Dubois and H. Prade, *Fuzzy Sets and Systems : Theory and Applications,* New York, Academic, 1980.

[50] D. Dubois and H. Prade, Possibility Theory, Plenum Press, New York, NY, USA (1988).

[51] R. Duda and P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, USA (1973).

[52] R. Duda, P. Hart, D. Stork, *Pattern classification* (2nd edition), Wiley, New York (2001), ISBN 0-471-05669-3

[53] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New York (2002).

[54] 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.

[55] J.C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact, well separated clusters, *J. Cybernet.* 3 (1974), pp. 95–104.

[56] P. D'Urso and P. Giordani, A weighted fuzzy c-means clustering model for fuzzy data, *Comput. Statist. Data Anal.* 50 (2006) (6), pp. 1496–1523

[57] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X, A densitybased algorithm for discovering clusters in large spatial databases with noise. In *2nd International Conference on Knowledge Discovery and Data Mining,* (1996), pages 226 231.

[58] B.S. Everitt and D.J. Hand, Finite Mixture Distributions, Chapman & Hall, London (1981).

[59] M. Ezzigaf, B. Mareschal, Utilisation d'échelles qualitatives dans les méthodes PROMETHEE, Institut de Statistique et de Recherche Opérationnelle, Université Libre de Bruxelles, 2000.

[60] B. Everitt, Cluster Analysis, Halsted, New York, 2001.

[61] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugenics* 7 (1936) (2), pp. 179–188.

[62] L. Fisher and J.W. Van-Ness, Admissible clustering procedures, *Biometrika* 58 (1971) (1), pp. 91–104.

[63] E. W. Forgy, Cluster Analysis of Multivariate Data : Efficiency Versus Interpretability of Classifications", *Biometrics,* 21, pp. 768-769, 1965.

[64] J. H. Friedman, and J. J. Meulman, Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society : Series B (Statistical Methodology),* (2004), 66(4) :1 25.

[65] H. Frigui, O. Nasraoui, Unsupervised learning of prototypes and attribute weights, Pattern Recognition 37 (2000) 567–581.

[66] H. Frigui and R. Krishnapuram, A robust algorithm for automatic extraction of an unknown number of clusters from noisy data, *Pattern Recognition Lett.* 17 (1996), pp. 1223–1232.

[67] I. Gath and A.B. Geva, Unsupervised optimal fuzzy clustering, *IEEE Trans. Pattern Anal. Mach. Intelligence* 11 (1989), pp. 773–781.

[68] A.D. Gordon, Classification, Chapman & Hall/CRC, Boca Raton, FL, 1999.

[69] P.J.F. Groenen, K. Jajuga, Fuzzy clustering with squared Minkowsky distances, Fuzzy Sets and Systems 120 (2001) 227–237.

[70] S. Guenter and H. Bunke, Validation indices for graph clustering. In: J.-M. Jolion, W. Kropatsch and M. Vento, Editors, *Proceedings of the 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition* (2001), pp. 229–238.

[71] D. Gustafson and W. Kessel, Fuzzy clustering with a fuzzy covariance matrix, Proc. *IEEE* CDC, San Diego, CA, pp. 761-766, January 10-12, 1979.

[72] E.E. Gustafson and W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, *Proceedings of the IEEE Conference on Decision and Control, San Diego, CA*, IEEE Press, Piscataway, NJ (1979), pp. 761–766.

[73] M. Halkidi, Y. Batistakis and M. Vazirgiannis, On clustering validation techniques, *Intell. Inf. Syst. J.* 17 (2001) (2–3), pp. 107–145.

[74] R.J. Hathaway and J.C. Bezdek, Optimization of clustering criteria by reformulation, *IEEE Trans. Fuzzy Systems* 3 (1995), pp. 241–245.

[75] R.J. Hathaway, et Yingkang Hu, Density-Weighted Fuzzy c-Means Clustering, *IEEE* Trans., Fuzzy Systems, Volume 17, Issue 1, Feb. 2009 Page(s) :243 – 252.

[76] A. Hinneburg, and D. A. Keim, Cluster discovery methods for large data bases - from the past to the future. In *ACM SIGMOD International Conference on Management of Data.* Tutorial Session, (1999).

[77] F. Höppner, F. Klawonn, R. Kruse and T. Runkler, Fuzzy Cluster Analysis, Wiley, Chichester, UK (1999).

[78] A.K. Jain, M.N. Murty and P.J. Flynn, Data clustering: a review, *ACM Comput. Surv.* 31 (1999) (3), pp. 264–323.

[79] A.K. Jain, A. Topchy, M. Law, J.M. Buhmann, Landscape of clustering algorithms, in: Pattern Recognition, 2004, ICPR 2004, Proceedings of the 17th International Conference on, vol. 1, iss., 23–26 August 2004, 2004, pp. 260–263.

[80] S. C. Johnson, Hierarchical Clustering Schemes, *Psychometrika,* vol. 32, pp. 241-254, 1967.

[81] J. Jolion, and A. Rosenfeld, Cluster detection in background noise, *Pattern Recognition,* vol. 22, n. 5, pp. 603-607, 1989.

[82] I. T. Jolliffe, *Principal Component Analysis.* Springer Verlag, New-York (1986).

[83] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, New York, 1990.

[84] G. Karypis, E.-H. S. Han and V.K. NEWS, Chameleon: Hierarchical clustering using dynamic modeling. Computer (1999), 32(8) : 68 75.

[85] L. Khodja, Contribution à la classification floue non supervisé, thèse de l'Université de Savoie, 1997.

[86] F. Klawonn and A. Keller, Fuzzy clustering with evolutionary algorithms, *Internat. J. Intelligent Systems* 13 (1998), pp. 975–991.

[87] F. Klawonn and F. Höppner, An alternative approach to the fuzzifier in fuzzy clustering to obtain better clustering results, *Proceedings of the Third Conference for Fuzzy Logic and Technology (Eusflat), Zittau/Goerlitz* (2003), pp. 730–734.

[88] F. Klawonn and F. Höppner, What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. In: Advances in Intelligent Data Analysis V, Lecture Notes in Computer Science, vol. 2810. Springer GmbH, Berlin, 2003, pp. 254–264, 3-540-40383-3.

[89] G. klir and T. Folger, *Fuzzy Sets, uncertainty, and information. Englewood Cliffs, NJ: Prentice-Hall* , 1988, chap.4.*Lett.*, vol 12, no. 11, pp. 657-664, 1992.

[90] R. Krishnapuram and J. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Systems* 1 (1993), pp. 98–110.

[91] R. Krishnapuram, Generation of membership functions via possibilistic clustering, in *Proceedings of the 3rd IEEE Conf. Fuzzy Syst.,* Orlando, FL, USA, July 1994, vol. 2, pp. 902-908.

[92] R. Krishnapuram and J. Keller, The possibilistic c-means algorithm: insights and recommendations, *IEEE Trans. Fuzzy Systems* 4 (1996), pp. 385–393.

[93] R. Krishnapuram, J. Kim, A note on the Gustafson–Kessel and adaptive fuzzy clustering algorithms, *IEEE Trans*. Fuzzy Systems 7 (4) (1999), 453–461.

[94] T. Lange, M. Braun, V. Roth, J.M. Buhmann, Stability-based model selection, Advances in Neural Information Processing Systems.

[95] L. Lebart, A. Morineau, M. Piron, Statistique exploratoire multidimensionnelle, Dunod, 2000.

[96] Z. Lubovac, B. Olsson, P. Jonsson, K. Laurio and M.L. Anderson, Biological and statistical evaluation of clusterings of gene expression profiles. In: C.E. D'Attellis, V.V. Kluev and N.E. Mastorakis, Editors, *Proceedings of Mathematics and Computers in Biology and Chemistry*, WSES Press (2001), pp. 149–155.

[97] J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, *Proc. 5th Berkeley Symp.,* 1965, pp. 281-297.

[98] B.R. Meijer, Règles et algorithmes pour la conception des modèles pour le modèle de correspondance, Pattern Recognition, 1992. Vol.1. Conférence A: Computer Vision and Applications, 11th IAPR International Conference on, pp: 760 - 763, août 1992.

[99] J. Christoph, SVM-base fonctionnalité de sélection par l'objectif direct minimisation , 2004.

[100] M. L. Ould Ahmedou, Amélioration des Méthodes de Classification Automatique Non Supervisée pour la Segmentation des Images Multi-Composantes, Thèse de l'Université de Reims, France, 1998.

[101] N. Pal, K. Pal and J. Bezdek, A mixed c-means clustering model, *Proceedings of the FUZZ-IEEE*(1997), pp. 11–21.

[102] N. Pal, K. Pal, J. Keller and J. Bezdek, A new hybrid c-means clustering model, *Proceedings of the FUZZ-IEEE'04* (2004), pp. 179–184.

[103] L. Parsons, E. Haque and H. Liu, Evaluating subspace clustering algorithms. In *Workshop on Clustering High Dimensional Data and its Applications, SIAM International Conference on Data Mining* (2004), pages 48-56.

[104] S. Pettie, and V. Ramachandran, An optimal minimum spanning tree algorithm. In *Automata, Languages and Programming*(2000), pages 49 60.

[105] M. Pirlot, *Métaheuristiques pour l'optimisation combinatoire: un aperçu général,* Faculté Polytechnique de Mons, 2004.

[106] N. Ragot, Reconnaissance de formes par modélisation mixte Intrinsèque /discriminante à base de systèmes d'inférence floue hiérarchisés, thèse de doctorat à l'Université de Rennes, 2003.

[107] Rifqi et S. Monties. Fuzzy prototypes for fuzzy data mining. In O. Pons, A. Vila, et J. Kacprzvk (édité par), Knowledge Management in Fuzzy Databases. Phvsica-Verlag, 2000.

[108] V. Roth, M. Braun, T. Lange and J.M. Buhmann, Stability-Based Model Order Selection in Clustering with Applications to Gene Expression Data, Springer, Berlin (2002).

[109] T.A. Runkler and J.C. Bezdek, Race: relational alternating cluster estimation and the wedding table problem. In: Brauer, W. (Ed.), Fuzzy-Neuro-Systems '98, München, Proceedings in Artificial Intelligence (1998), vol. 7, pp. 330–337.

[110] T.A. Runkler and J.C. Bezdek, Web mining with relational clustering, *Internat. J. Approx. Reasoning* 32 (2003) (2–3), pp. 217–236.

[111] E. R. Ruspini, A New Approach to Clustering, *Inform. Control,* vol. 15, no. 1, pp. 22-32, July 1969.

[112] G. Saporta, *Probabilités, analyses des données et statistique,* editions Technip, 1990.

[113] G. S. Sebestyen, *Decision Making Processes in Pattern Recognition,* New York : Macmillan, 1962.

[114] G. Shafer, *A mathematical theory of evidence*. Princeton, NJ: Princeton University Press. 1979.

[115] H. Spaeth, Cluster Analysis Algorithms, Wiley, New York, 1980.

[116] L. Talavera,. Dependency-based feature selection for clustering symbolic data. *Intelligent Data Analysis* (2000) 4 :19- 28.

[117] W.-C. Tjhi, et Lihui Chen, Dual Fuzzy-Possibilistic Coclustering for Categorization of Documents, *IEEE Trans.* , Fuzzy Systems, Volume: 17, Issue: 3, page(s) : 532-543, 2009.

[118] H. Timm, C. Borgelt, C. Döring and R. Kruse, An extension to possibilistic fuzzy cluster analysis, *Fuzzy Sets and Systems* 147 (2004), pp. 3–16.

[119] J.W. Van-Ness, Admissible clustering procedures, *Biometrika* 60 (1973) (2), pp. 422–424.

[120] Ph. VINCKE, cours de Recherche opérationnelle, Service de Mathématiques de la Gestion, Université Libre de Bruxelles, 2004.

[121] Verma, D. and Meila, M. (2003), A comparison of spectral clustering algorithms.

[122] R. Wagner et M. Fisher, The string-to-string correction problem. Journal of the AGM, 12(1) :168 173, 1974.

[123] W. Wang, J. Yang, and R. R. Muntz, STING: A statistical information grid approach to spatial data mining. In Jarke, M., Carey, M. J., Dittrich, K. R., Lochovsky, F. H., Loucopoulos, P., and Jeusfeld, M. A., editors, 23rd International Conference on Very Large Data Bases (1997), pages 186 - 195.

[124] Yaonan Wang, Chunsheng et Li Yi Zuo, A Selection Model for Optimal Fuzzy Clustering Algorithm and Number of Clusters Based on Competitive Comprehensive Fuzzy Evaluation, *IEEE Trans.*, Fuzzy Systems, Volume: 17, Issue: 3, June 2009, page(s) : 568-577.

[125] K. Wu and M. Yang, Alternating c-means clustering algorithms, *Pattern Recognition* 35 (2002), pp. 2267–2278.

[126] R. Xu, et D. Wunsch, (2008) *Clustering* (Edition: 1), Wiley-*IEEE* Press ,*IEEE* Press Series on Computational Intelligence.

[127] M.S. Yang, A survey of fuzzy clustering, Math. Comput. Modelling 18 (11) (1993) 1–16.

[128] L. Ye, and M. Spetsakis, Clustering on unobserved data using mixture of gaussians. Technical report, York University, Toronto, Canada (2003).

[129] K.Y. Yeung, D.R. Haynor and W.L. Ruzzo, Validating clustering for gene expression data, *Bioinformatics* 17 (2001), pp. 309–318.

[130] L.A. Zadeh, Fuzzy sets, *Inform. Control* 8 (1965), pp. 338–353.

[131] L. A. Zadeh, Fuzzy Sets as a Basis for a Theory of Possibility, *Fuzzy Sets and Systems,* vol. 1, pp. 3-28, 1978.

[132] Lihong Zheng and Xiangjian He, Classification Techniques in Pattern

Recognition, *Conference proceedings ISBN 80-903100-8-7 WSCG'2005, January 31-February 4*, 2005.

[133] H. J. Zimmermann and P. Zysno, Quantifying vagueness in decision models, European J. Operational Res., vol. 22, pp. 148-158, 1985.

[134] H. J. Zimmerman, *Fuzzy Set Theory and Its Applications,* Dordrecht, Holland : Kuwer, 1991.

Table des figures

Liste des tableaux