THÈSE DE DOCTORAT DE L'UNIVERSITÉ CLERMONT AUVERGNE

Laboratoire d'Informatique, de Modélisation
et d'Optimisation des Systèmes

Ecole Doctorale Sciences pour l'ingénieur
Spécialité de doctorat : Informatique

Soutenue publiquement le 12/01/2024, par :

Benoit ALBERT

Méthodes d'optimisation avancées pour la classification automatique

---

Devant le jury composé de :

BARRA Vincent Professeur, Université Clermont Auvergne	**Président**
DENŒUX Thierry Professeur, Université de Technologie de Compiègne	**Rapporteur**
MERCIER David Professeur, Université d'Artois	**Rapporteur**
JEHAN-BESSON Stéphanie Chercheuse, CNRS	**Examinatrice**
LESOT Marie-Jeanne Professeure, Sorbonne Université	**Examinatrice**
ANTOINE Violaine MCF HDR, Université Clermont Auvergne	**Co-Directrice de thèse**
Et la présence de :	
KOKO Jonas MCF HDR, Université Clermont Auvergne	**Directeur de thèse**

«Si l'on veut obtenir quelque chose que l'on n'a jamais eu, il faut tenter quelque chose que l'on n'a jamais fait.»
Périclès

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude envers mes directeurs de thèse, Docteurs Antoine Violaine et Jonas Koko, pour leur confiance et leur soutien, leur expertise et leur accompagnement tout au long de ces trois années.

Un merci sincère aux membres du jury, en particulier aux rapporteurs, les Professeurs Thierry Denœux et David Mercier, pour leurs précieux commentaires, leur expertise approfondie et leur contribution significative à l'évaluation de ma thèse. À tous les membres du jury, les examinatrices Docteur Stéphanie Jehan-Besson et Professeure Marie-Jeanne Lesot, et le président Professeur Vincent Barra, je suis reconnaissant pour le temps et l'effort qu'ils ont consacrés.

Un mot de reconnaissance à mes collègues de recherche, dont leur présence et le partage d'idées ont créé un environnement stimulant et dynamique. Je remercie mes collègues doctorants, anciens et nouveaux, pour les bons moments partagés. Je remercie particulièrement Shidi Deng pour son travail effectué durant son stage à mes côtés.

À mes amis, ma famille, mes parents, je souhaite exprimer ma gratitude pour leurs encouragements, leur soutien indéfectible. Cette réussite est aussi la leur.

Enfin, un merci à tous ceux qui ont contribué de près ou de loin à la réalisation de cette thèse. J'ai une pensée particulière pour chaque personne qui, sur ce long voyage de l'éducation et de la recherche, m'a fait grandir pour arriver à ce bel accomplissement.

Résumé

En partitionnement de données, l'objectif consiste à regrouper des objets en fonction de leur similarité. K-means est un des modèles les plus utilisés, chaque classe est représentée par son centroïde. Les objets sont assignés à la classe la plus proche selon une distance. Le choix de cette distance revêt une grande importance pour prendre en compte la similarité entre les données. En optant pour la distance de Mahalanobis au lieu de la distance euclidienne, le modèle est capable de détecter des classes de forme ellipsoïdale et non plus seulement sphérique. L'utilisation de cette distance offre de nombreuses opportunités, mais elle soulève également de nouveaux défis explorés dans ma thèse.

L'objectif central concerne l'optimisation des modèles, en particulier FCM-GK (variante floue de k-means) qui est un problème non convexe. L'idée est d'obtenir un partitionnement de meilleure qualité, sans créer un nouveau modèle en appliquant des méthodes d'optimisation plus robustes. À cet égard, nous proposons deux approches : ADMM (Alternating Direction Method of Multipliers) et la méthode du gradient accéléré de Nesterov. Les expériences numériques soulignent l'intérêt particulier de l'optimisation par ADMM, surtout lorsque le nombre d'attributs dans le jeu de données est significativement plus élevé que le nombre de clusters.

L'incorporation de la distance de Mahalanobis dans le modèle requiert l'introduction d'une mesure d'évaluation dédiée aux partitions basées sur cette distance. Une extension de la mesure d'évaluation de Xie et Beni est proposée. Cet index apparaît comme un outil pour déterminer la distance optimale à utiliser.

Enfin, la gestion des sous-ensembles dans ECM (variante évidentielle) est traitée en abordant la détermination optimale de la zone d'imprécision. Une nouvelle formulation des centroides et des distances des sous-ensembles à partir des clusters est introduite. Les analyses théoriques et les expérimentations numériques mettent en évidence la pertinence de cette nouvelle formulation.

Mots clés : Classification automatique floue, classification automatique évidentielle, distance de Mahalanobis, optimisation non convexe, mesure interne, zone d'imprécision.

Vous avez la possibilité de consulter les codes de mes travaux sur ma page GitHub : https://github.com/BenoitAlbertScientist.

Abstract

In data partitioning, the goal is to group objects based on their similarity. K-means is one of the most commonly used models, where each cluster is represented by its centroid. Objects are assigned to the nearest cluster based on a distance metric. The choice of this distance is crucial to account for the similarity between the data points. Opting for the Mahalanobis distance instead of the Euclidean distance enables the model to detect classes of ellipsoidal shape rather than just spherical ones. The use of this distance metric presents numerous opportunities but also raises new challenges explored in my thesis.

The central objective is the optimization of models, particularly FCM-GK (a fuzzy variant of k-means), which is a non-convex problem. The idea is to achieve a higher-quality partitioning without creating a new model by applying more robust optimization methods. In this regard, we propose two approaches : ADMM (Alternating Direction Method of Multipliers) and Nesterov's accelerated gradient method. Numerical experiments highlight the particular effectiveness of ADMM optimization, especially when the number of attributes in the dataset is significantly higher than the number of clusters.

Incorporating the Mahalanobis distance into the model requires the introduction of an evaluation measure dedicated to partitions based on this distance. An extension of the Xie and Beni evaluation measure is proposed. This index serves as a tool to determine the optimal distance to use.

Finally, the management of subsets in ECM (evidential variant) is addressed by determining the optimal uncertainty zone. A new formulation of centroids and distances for subsets from clusters is introduced. Theoretical analyses and numerical experiments underscore the relevance of this new formulation.

Keywords : Fuzzy clustering, evidential clustering, Mahalanobis distance, non-convex optimization, internal measure, uncertainty zone.

You can review the code for my work on my GitHub page :
https://github.com/BenoitAlbertScientist.

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Liste de symboles et abréviations

Modèle et méthodes

ADMM	Alternating Direction Method of Multpliers *(directions alternées)*
AO	Alternating Optimisation *(optimisation alternée)*
AO-APG	Alternating Optimisation associée à APG
APG	Accelerated Projected Gradiant *(gradient proximal accéléré)*
ECM	Evidential C-Means
ECM+	Evidential C-Means amélioré
FCM	Fuzzy C-Means
GK,FCM-GK	Fuzzy C-Means extension Gustafon et Kessel
HCM	Hard C-Means

Paramètres et indices

α	Paramètre contrôle des sous-ensembles pour ECM
β	Paramètre de fuzzification pour ECM (équivalent à m)
δ	Distance à l'ensemble $\emptyset$ pour ECM, Inverse de la constance de Lipschitz pour APG
ℓ	Indice des classes
c	Nombre de classes
i	Indice des objets
j	Indice des sous-ensembles pour ECM et des classes pour FCM
m	Paramètre de fuzzification pour FCM (équivalent à β)
n	Nombre d'objets
n_d	Nombre d'attributs
r	Paramètre de pénalité pour ADMM

Variables

$\boldsymbol{M}$	Partition crédale
$\boldsymbol{S}, \boldsymbol{\mathcal{S}}$	Matrice de la distance de Mahalanobis, Ensemble de ces matrices
$\boldsymbol{U}$	Partition floue
$\boldsymbol{V}, \boldsymbol{\mathcal{V}}$	Centroïdes, Ensemble des centroïdes
$\boldsymbol{X}$	Matrice du jeu de données
$\mathcal{A}, 2^{\Omega}$	Sous-ensemble, Ensemble puissance
ω, Ω	Classe, Cadre de discernement-Ensemble des classes

Chapitre 1

Introduction

Depuis l'Antiquité, l'homme a toujours cherché à extraire des connaissances à partir des données dont il disposait afin de mieux comprendre le monde qui l'entoure. L'exploration de données est une branche de l'informatique qui regroupe les méthodes visant à extraire de la connaissance à partir de grandes quantités de données [1]. Plus précisément, les méthodes de classification automatique, appelées clustering en anglais, visent à regrouper des objets en fonction de leur similarité. Lorsque ces groupes sont formés sans aucune connaissance préalable, on parle de classification non supervisée [2,3]. Les applications de la classification non supervisée sont nombreuses et variées. Ghosal et al. présente une liste non exhaustive des applications possibles [4]. Le non supervisé est, entre autre, utilisé pour la détection de fraude. Dans le domaine de la santé, il est employé pour l'analyse de données biologiques et pour l'imagerie médicale. Dans le secteur financier, il sert à l'analyse des marchés et à la segmentation de la clientèle. En urbanisme, il est utilisé pour la répartition et la disposition des services.

Selon leurs caractéristiques propres, les différents modèles de classification non supervisée sont capables de détecter des structures cachées dans les données. Parmi les plus utilisés, le modèle *k-means* se base sur la représentation de chaque classe par un centroïde, un objet "type" et affecte chaque objet à la classe dont le centroïde est le plus proche en utilisant une mesure de distance. Le choix de la distance est primordial pour permettre à *k-means* de détecter des structures cachées spécifiques. La distance euclidienne, usuellement employée, détecte des classes de forme sphérique exclusivement.

Dans cette étude, nous nous sommes intéressés à l'utilisation d'une distance adaptative à chaque classe qui forme des ellipses, en l'occurrence la distance de Mahalanobis. De plus, nous avons considéré les variantes de *k-means* qui génèrent des partitions floues afin de modéliser l'incertitude [5]. Dans ce contexte, les classes à forme ellipsoïdale peuvent se chevaucher, ce qui rend leur détection complexe en l'absence de métrique commune. L'application de cette distance en classification automatique soulève donc de nouveaux défis.

Dans cette thèse, nous souhaitons aborder les trois points clés d'une méthode de classification non supervisée : le modèle, son optimisation et sa validation par des mesures d'évaluation. Valider un modèle suppose être capable de prendre en compte sa spécificité. Le premier défi auquel nous sommes confrontés est le manque de mesures de validité pour une distance adaptative telle que la distance de Mahalanobis. Nous avons choisi d'adapter la mesure de Xie-Beni [6] .

Afin de tirer le meilleur bénéfice d'un modèle, il est essentiel de se focaliser sur la minimisation de son problème associé. L'idée est d'obtenir un partitionnement de meilleure qualité sans créer un nouveau modèle en appliquant des méthodes d'optimisation plus robustes. Cependant, le problème que nous étudions est non convexe, et la minimisation est rendue encore plus difficile avec les variables supplémentaires liées à la distance

de Mahalanobis. Nous proposons deux alternatives à la méthode d'optimisation alternée actuellement utilisée : ADMM (Alternating Direction Method of Multipliers) [7,8] et la méthode du gradient projeté accéléré (APG) de Nesterov [9,10]. ADMM est une méthode de décomposition-coordination simple mais puissante. Elle décompose le problème en sous-problèmes, tout en coordonnant les solutions obtenues localement. APG permet une relaxation d'une contrainte par la projection sur un ensemble associé.

En ce qui concerne la variante évidentielle de *k-means*, ECM [11], qui modélise l'imprécision en plus de l'incertitude de l'affectation d'un objet à une classe, la définition des sous-ensembles de ce modèle avec la distance de Mahalanobis ne peut pas être réalisée avec les mêmes formulations que celles de la distance euclidienne. Nous proposons une nouvelle approche pour y remédier.

Dans la première partie de cette thèse, nous introduisons dans le chapitre 2 les fondements théoriques de *k-means* et ses variantes, suivi dans le chapitre 3 des notions clés d'optimisation et des méthodes employées dans cette étude. Dans la seconde partie, nous présentons nos contributions : le chapitre 4 aborde notre nouvel indice d'évaluation, XBMW, le chapitre 5 traite de l'optimisation du modèle flou de *k-means* avec les méthodes d'optimisation ADMM et APG et le chapitre 6 présente la formulation que nous proposons pour la variante évidentielle de *k-means*, ECM+.

Première partie

État de l'art

Chapitre 2

Classification non supervisée

Contents

Le partitionnement de données consiste à partager les n objets d'un jeu de données dans des groupes, classes ou sous-ensembles, dans le but de regrouper les objets similaires et d'obtenir des groupes bien distincts les uns des autres [12]. Les modèles de classification non supervisée doivent définir trois concepts : le regroupement, la similarité (ou la dissemblance) et l'affectation des objets dans les groupes. Il existe plus de 100 méthodes de classification non supervisée différentes qui peuvent être classées en 4 grandes familles de modèles [4] : le regroupement hiérarchique [13, 14], le regroupement basé sur les partitions [15, 16], le regroupement basé sur la densité [17, 18] et le regroupement basé sur un "modèle" [19].

Pour la notion d'affectation, nous distinguons :
— Hard clustering *(partitionnement dur)* lorsque les objets appartiennent à un et un seule groupe (ou aucun) avec certitude.
— Soft clustering *(partitionnement souple)* lorsqu'en l'absence de certitude d'appartenance à un groupe, on propose un degré de croyance/certitude/appartenance à un groupe.

Les données traitées par ces méthodes peuvent être de nature variée : booléenne, catégorielle, numérique (continue ou non). La définition de la similarité ou de la dissimilarité dépend fortement de la nature des données et du modèle choisi.

Dans mon étude, je me suis intéressé aux méthodes basées sur les partitions appliquées à des données numériques et continues. Parmi les méthodes de regroupement basé sur les partitions, *k-means* est l'une des plus célèbres formant une partition dure. Chacun des k groupes est représenté par un centroïde (ou centre de gravité) qui correspond à un objet type, et le concept de similarité est défini par la distance entre les objets et les centroïdes. Grâce à sa faible complexité et sa simplicité, cette méthode est très utilisée [20, 21]. Il existe de nombreuses applications dans des domaines divers : environnemental pour l'analyse des risques [22, 23], médical pour la détection des cancers [24], financier pour la détection de fraudes, la segmentation des clients [25, 26]... L'algorithme *k-means* génère une partition dure, c'est-à-dire qu'il prend une décision stricte quant à l'affectation d'un objet dans une unique classe.
Cependant, il existe de nombreuses situations où une décision stricte est inadéquate car forcée. En effet, l'indécision dans l'affectation d'une classe pour un objet n'étant pas permise, il peut arriver que *k-means* ait à prendre une décision alors que l'algorithme manque d'informations pour réaliser ce choix. Le partitionnement souple quant à lui permet de modéliser l'incertitude. Le cas échéant, c'est un expert qui prendra la décision, la méthode de classification non supervisée est de ce point de vue un outil d'aide à la décision [27] comme en analyse d'imagerie médicale [28].
La prise en compte de l'incertitude décisionnelle permet d'extraire davantage d'informations. L'analyse peut-être améliorée par l'application de distances adaptatives pour chaque classe qui modélise leur variabilité.

Dans ce chapitre, nous introduisons dans un premier temps le modèle de *k-means* (section 2.1). Du point de vue de la gestion de l'incertitude, nous présentons des variantes de ce modèle et leur fondement théorique (sections 2.2 et 2.3). La dernière section fait référence aux mesures permettant d'évaluer les partitions de ces modèles (section 2.5).

2.1 *k-means* (HCM)

2.1.1 Hard C-Means : classification non supervisée dure

Bien que popularisé sous le nom *k-means*, par soucis d'homogénéité avec les autres modèles, nous utiliserons le nom de **Hard C-Means** (**HCM**) dans la suite de cette thèse.
L'objectif de HCM est de regrouper en c classes, $\Omega = \{\omega_1, ..., \omega_c\}$, les données $\boldsymbol{X} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_n)$ contenant n objets ayant n_d attributs, $\boldsymbol{x}_i \in \mathbb{R}^{n_d}, \forall i \in \{1, n\}$. Le nombre de classes est compris entre deux et le nombre d'objets, $2 \leq c < n$.

Dans ce modèle, le regroupement est défini comme une prise de décision correspondant au fait d'associer à chaque objet $\boldsymbol{x}_i, \forall i \in \{1, n\}$, une unique classe $\omega_j, \forall j \in \{1, c\}$.

Définition 2.1.1: Partition dure

Soit la matrice booléenne $\boldsymbol{H}$ de dimension $(n \times c)$ définie :

$$\forall i \in \{1, n\}, \forall j \in \{1, c\}, \quad h_{ij} = \begin{cases} 1 & \text{si } \boldsymbol{x}_i \in \omega_j, \\ 0 & \text{sinon.} \end{cases}$$

h_{ij} est la valeur booléenne d'appartenance de l'objet $\boldsymbol{x}_i$ dans la classe ω_j.
$\boldsymbol{H}$ est une **partition dure** si elle respecte les deux contraintes :

$$\sum_{j=1}^{c} h_{ij} = 1 \quad \forall i \in \{1, n\}, \tag{2.1}$$

$$\sum_{i=1}^{n} h_{ij} > 0 \quad \forall j \in \{1, c\}. \tag{2.2}$$

Chaque objet appartient à une unique classe (2.1), et chaque classe possède au moins un objet (2.2).
L'**ensemble des partitions dures** est noté :

$$M_{hp} = \left\{ \boldsymbol{H} \in \mathbb{R}^{n \times c} | h_{ij} = \{0, 1\}; \sum_{j=1}^{c} h_{ij} = 1 \quad \forall i; \sum_{i=1}^{n} h_{ij} > 0 \quad \forall j \right\}. \tag{2.3}$$

Chaque classe est représentée par son centre de gravité, ce qui donne le nom au modèle : *k-means* [29]. Les centres de gravité sont notés $\boldsymbol{\mathcal{V}} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_c\}$.
Pour $\boldsymbol{H}$, les objets sont affectés à la classe dont le centre de gravité est le plus "proche" selon une distance définie d.

Exemple 2.1.1: Partitionnement dure

Dans le but d'illustrer la classification non supervisée dure, nous considérons un exemple avec deux classes. La figure 2.1.1 met en avant la frontière nette qui résulte de la décision booléenne. Les fonctions des degrés d'appartenance correspondantes sont données dans les figures 2.1.2-2.1.3 .

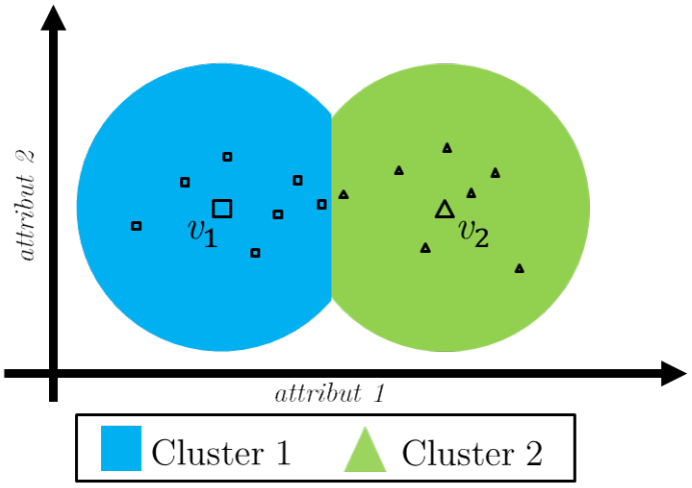


FIGURE 2.1.1 – Exemple de deux classes.

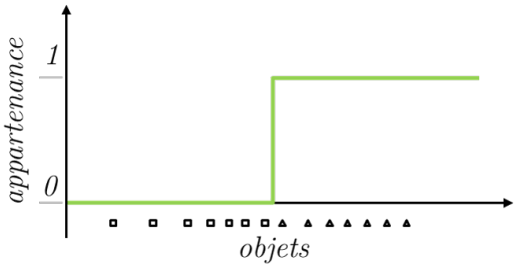


FIGURE 2.1.2 – Fonction d'appartenance de la classe 1.

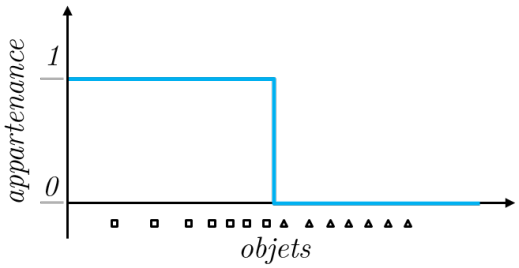


FIGURE 2.1.3 – Fonction d'appartenance de la classe 2.

La méthode HCM cherche $(\boldsymbol{H}, \boldsymbol{\mathcal{V}})$ minimisant les distances intra-classes :

$$J_{HCM}(\boldsymbol{H}, \boldsymbol{\mathcal{V}}) = \sum_{i=1}^{n} \sum_{j=1}^{c} h_{ij} d_{ij}^2, \tag{2.4}$$

$$\text{sous les contraintes (2.1)-(2.2).} \tag{2.5}$$

Avec les variables binaires,

$$h_{ij} = \begin{cases} 1 & \text{si } \boldsymbol{x}_i \in \omega_j, \\ 0 & \text{sinon.} \end{cases}$$

2.1.2 Algorithme de Lloyd

L'heuristique couramment utilisée pour résoudre ce problème est l'algorithme de Lloyd, retranscrit par l'algorithme 1. Cette méthode d'optimisation a été introduite pour la première fois par Lloyd [30], dans le traitement du signal pour la modulation par impulsions codées. Dans son travail, il a utilisé l'erreur des moindres carrés correspondant à la distance euclidienne. L'algorithme est un processus itératif en deux étapes : calculer le barycentre des objets dans la classe et mettre à jour la partition en fonction de l'affectation de chaque objet dans la classe la plus proche.

Algorithme 1 HCM par l'algorithme de Lloyd.

Entrée : $\boldsymbol{X}$ les données, c le nombre de classes.
Sortie : $\boldsymbol{H}^k$,$\boldsymbol{\mathcal{V}}^k$

1: $err = 0, k = 0$
2: $\boldsymbol{H}^0$ initialisation aléatoire.
3: **tant que** $err > \varepsilon = 10^{-3}$ **faire**
4: $\quad k = k + 1$
5: $\quad$ calcul $\boldsymbol{\mathcal{V}}^k : \boldsymbol{v}_j^k = \frac{\sum_{i=1}^{n} h_{ij}^{k-1} \boldsymbol{x}_i}{\sum_{i=1}^{n} h_{ij}^{k-1}}, \quad \forall j \in \{1, c\}.$
6: $\quad$ calcul $\boldsymbol{H}^k : \begin{cases} h_{il}^k = 1 \quad \text{avec } l = \underset{j}{argmin}(d_{ij}^2), \\ h_{ij}^k = 0 \quad \forall j \in \{1, c\}, j \neq l, \end{cases} \quad \forall i \in \{1, n\}.$
7: $\quad err = \| \boldsymbol{H}^k - \boldsymbol{H}^{k-1} \|$
8: **fin tant que**

Dans l'algorithme 1, l'initialisation est réalisée par une généralisation aléatoire de la partition dure. Bien que cette pratique soit la plus courante, il est également possible d'initialiser l'algorithme par les centroïdes, soit en prenant des valeurs aléatoires dans l'espace défini $\mathbb{R}^{n_d}$, soit en tirant aléatoirement c objets de $\boldsymbol{X}$.

La condition d'arrêt $\| \boldsymbol{H}^k - \boldsymbol{H}^{k-1} \|$ concrétise la stabilité de la partition, lorsqu'il n'y a plus d'évolution à ε-près, l'algorithme s'arrête. La segmentation de l'espace forme une partition de Voronoï [31] : l'espace est divisé en régions autour de chaque centroïde, de sorte que tous les points à l'intérieur d'une région sont plus proches de ce centroïde que de tous les autres.

L'algorithme HCM est de la classe des problèmes NP-difficile [32]. Nous sommes seulement assurés d'une convergence vers un minimum local en raison de sa nature heuristique gloutonne [20].

2.1.3 Illustration et limites de HCM

En vue d'illustrer les différents modèles de *k-means* et leur gestion de l'incertitude, dans la suite de ce chapitre, nous étudions l'exemple suivant.

Exemple 2.1.2: Ensemble symbole en 2 classes - partition dure

Soit l'ensemble $\boldsymbol{X}$ constitué des symboles $\{|, \|, \equiv, \_, [, +, \perp, /, o\}$, nous souhaitons regrouper les objets de $\boldsymbol{X}$ en deux classes $\Omega = \{\omega_1, \omega_2\}, c = 2$.
Considérons que la première classe ω_1 contienne les barres verticales "|" et la deuxième ω_2 les barres horizontales " − ".
En observant les objets $\{|, \|, \equiv, \_\}$, il est évident d'affecter les deux premiers symboles à la classe des barres verticales ω_1 et pour les deux autres à la classe des barres horizontales ω_2. Le tableau 2.1 présente la partition résultante.

Objets	ω_1	ω_2
$\|$	1	0
$\|\|$	1	0
$\equiv$	0	1
$\_$	0	1

TABLEAU 2.1 – Partition dure à deux classes.

Pour les objets étant constitués des barres verticales et horizontales $\{[, +, \perp\}$, la prise de décision est difficile. Pour l'objet [, la barre verticale étant plus grande que les barres horizontales, l'affectation dans la classe ω_1 semble cohérente. En revanche, l'affectation des objets $\{+, \perp\}$, est arbitraire voir le tableau 2.2.

Objets	ω_1	ω_2
$[$	1	0
$+$	?	?
$\perp$	?	?

TABLEAU 2.2 – Problématique liée à la partition dure.

Cet exemple illustre le paradoxe entre la prise de décision, qui implique un choix, et la modélisation de l'incertitude, qui nécessite de conserver le maximum d'informations. Face à la logique booléenne incapable de représenter l'incertitude, de nouveaux concepts doivent être introduits.

2.2 Variantes floues

2.2.1 Ensemble flou

En 1965, dans l'intention de représenter l'incertitude, Zadeh [33] a développé la théorie des ensembles flous servant de base à la logique floue. Elle étend la logique booléenne classique : les valeurs de décision ne sont plus soit vraies (1), soit fausses (0) mais sont

des variables comprises entre 0 et 1.

Définition 2.2.2: Fonctions d'appartenances

L'ensemble des décisions possibles $E = \{e_1, \ldots, e_c\}$ est appelé le **cadre de discernements**. En théorie classique des ensembles la **fonction caractéristique** associée à A est I_A (E $\rightarrow \{0, 1\}$) modélise l'appartenance ou non d'un élément x au sous-ensemble A de E,

$$I_A(x) = \begin{cases} 1 \text{ si } x \in A, \\ 0 \text{ sinon.} \end{cases}$$

La **fonction d'appartenance** U_A (E $\rightarrow [0, 1]$) représente le degré de validité que la proposition x appartient à A ($x \in A$), nous disons que $U_A(x)$ est le **degré d'appartenance** de x à A. Cette fonction caractérise le sous-ensemble flou, ou la partie floue A de E.

Définition 2.2.3: Notions d'une partie floue

Le **noyau** d'une partie floue A est l'ensemble des éléments qui appartiennent totalement à A : $n(A) = \{x \in E | U_A(x) = 1\}$.
Le **support** d'une partie floue A est l'ensemble des éléments ayant un degré d'appartenance à A non nul : $supp(A) = \{x \in E | U_A(x) > 0\}$.
Une **coupe** α d'une partie floue A est le sous-ensemble des éléments ayant un degré d'appartenance supérieur ou égal à α : $A^{\geq \alpha} = \{x \in E | U_A(x) \geq \alpha\}$. La coupe est dite stricte lorsque le degré d'appartenance est strictement supérieur $A^{>\alpha}$. En particulier, nous avons $A^{\geq 1} = n(A)$ et $A^{>0} = supp(A)$.

Définition 2.2.4: Opérateurs

Soit deux parties floue A, B de E et $x \in E$.
L'opérateur **complémentaire**, correspondant à l'opérateur booléen "non", est défini :

$$U_{\neg A}(x) = 1 - U_A(x). \tag{2.6}$$

L'opérateur d'**intersection**, correspondant à l'opérateur booléen "et", est défini :

$$U_{A \cap B}(x) = min(U_A(x), U_B(x)). \tag{2.7}$$

L'opérateur d'**union**, correspondant à l'opérateur booléen "ou", est défini :

$$U_{A \cup B}(x) = max(U_A(x), U_B(x)). \tag{2.8}$$

2.2.2 Fuzzy C-Means (FCM)

Bezdek et Dunn [5,34] ont adapté la logique des ensembles flous à HCM pour modéliser l'incertitude. Le cadre de discernements $\Omega = \{\omega_1, ..., \omega_c\}$ est l'ensemble des classes. Les ensembles flous étudiés sont les c classes. La partition décrit la probabilité que chaque objet, $\boldsymbol{x}_i, \forall i \in \{1, n\}$, appartient à la classe $\omega_j, \forall j \in \{1, c\}$. Ainsi, la partition dure $\boldsymbol{U}$ est remplacée par une partition floue (probabiliste).

Définition 2.2.5: Partition floue

Soit la matrice réelle $\boldsymbol{U}$ de dimension $(n \times c)$ définie :

$$\forall i \in \{1, n\}, \forall j \in \{1, c\}, \quad 0 \leq u_{ij} \leq 1. \tag{2.9}$$

u_{ij} est le degré d'appartenance de l'objet $\boldsymbol{x}_i$ dans la classe ω_j, noté $U_{\omega_j}(\boldsymbol{x}_j)$ dans la théorie des ensembles flous.
$\boldsymbol{U}$ est une **partition floue** si elle respecte les deux contraintes suivantes :

$$\sum_{j=1}^{c} u_{ij} \quad = 1 \quad \forall i \in \{1, n\}, \tag{2.10}$$

$$\sum_{i=1}^{n} u_{ij} \quad > 0 \quad \forall j \in \{1, c\}. \tag{2.11}$$

Pour chaque objet, la somme des degrés d'appartenance est égale à un, et chaque classe possède au moins un objet.
L'**ensemble des partitions floues** est noté :

$$M_{fp} = \left\{ \boldsymbol{U} \in \mathbb{R}^{n \times c} | u_{ij} \in [0, 1]; \sum_{j=1}^{c} u_{ij} = 1 \quad \forall i; \sum_{i=1}^{n} u_{ij} > 0 \quad \forall j \right\}. \tag{2.12}$$

Le modèle FCM applique la théorie des probabilités pour définir la partition floue. En effet, la contrainte 2.2.2 traduit le fait que la fonction $U_{\cdot}(\boldsymbol{x}_i) : \Omega \longrightarrow [0, 1]$ est une fonction de répartition. Cette fonction généralise la relation initialement proposée par Zadeh [33], pour définir les degrés d'appartenance de tout point x de E à un ensemble flou A et son complémentaire $\bar{A}$ tel que $U_A(x) + U_{\bar{A}}(x) = 1$.
Enfin la deuxième contrainte assure que chaque classe est utile. Autrement dit, la fonction d'appartenance de chaque classe ω_j doit avoir un support non vide, $U_j^{>0} \neq \emptyset$.

Exemple 2.2.3: Partitionnement flou

Nous reprenons l'exemple 2.1.1 en appliquant une modélisation floue illustrée par la figure 2.2.1. La frontière est devenue floue et les degrés d'appartenance

des objets aux deux classes sont des fonctions continues différentiables que nous observons sur les figures 2.2.2-2.2.3.

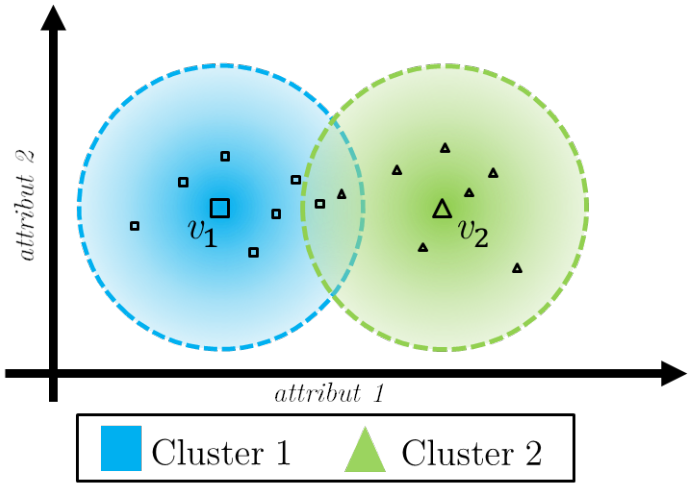


FIGURE 2.2.1 – Exemple de deux classes floues.

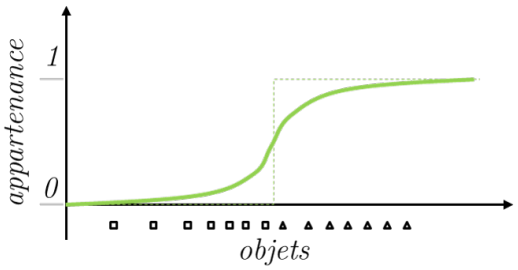


FIGURE 2.2.2 – Fonction d'appartenance de la classe 1.

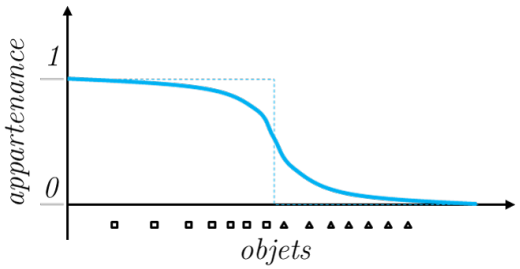


FIGURE 2.2.3 – Fonction d'appartenance de la classe 2.

La méthode FCM cherche $(\boldsymbol{U}, \boldsymbol{\mathcal{V}})$ en minimisant les distances intra-classes :

$$J_{FCM}(\boldsymbol{U}, \boldsymbol{\mathcal{V}}) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} d_{ij}^{2}, \tag{2.13}$$

sous les contraintes (2.9)-(2.11).

où $m \geq 1$, appelé *paramètre de fuzzification*, est un hyperparamètre du modèle qui contrôle la dureté de la partition. Une valeur plus élevée du paramètre permet une plus grande incertitude et ainsi le chevauchement des classes est favorisé. Une valeur plus faible conduit à une séparation plus stricte des classes, les attributions sont plus déterministes jusqu'au moment où $m = 1$ qui réduit FCM à HCM. Généralement, m est fixé à 2 [35].

Par défaut d_{ik}^2 est la distance euclidienne. Nous évoquons un peu plus loin dans la section 2.4.2 d'autres distances testées pour la plupart avec FCM.

Défuzzification

Pourquoi prendre une décision à partir d'un modèle probabiliste ?
La classification non supervisée floue, telle que FCM, fournit des degrés d'appartenance, permettant ainsi de modéliser l'incertitude et l'ambiguïté dans les regroupements. Lorsque nous souhaitons prendre une décision, c'est-à-dire affecter un objet à une classe, il est pertinent de le faire à partir d'un modèle probabiliste. En effet, notre choix sera éclairé par la connaissance des probabilités associées à chaque décision.

Comment prendre une décision à partir d'un modèle probabiliste ?
Pour transformer une partition floue en une partition dure, il convient d'affecter l'objet $\boldsymbol{x}_i$ à la classe ω_j dont la probabilité est maximale :

$$u_{ij} = \begin{cases} 1 & \text{si } j = \underset{\ell \in [1,c]}{argmax}(U_{\omega_\ell}(\boldsymbol{x}_i)), \\ 0 & \text{sinon.} \end{cases}$$

Illustration et limites de FCM

Pour illustrer le modèle probabiliste, reprenons l'exemple 2.1.2 de l'ensemble des symboles.

Exemple 2.2.4: Ensemble symbole en 2 classes - partition floue

Pour rappel, nous souhaitons regrouper $\boldsymbol{X} = \{|, \|, \equiv, \_, [, +, \perp, /, o\}$ en deux classes $\Omega = \{\omega_1, \omega_2\}$, les barres verticales et les barres horizontales.
Pour les objets $\{|, \|, \equiv, \_\}$, la partition floue est identique à la partition dure voir le tableau 2.1.
De surcroît, les objets $\{[, +, \perp\}$ sont problématiques à cause de leur caractérisation associable aux deux classes (incertitude). L'objet [possède une barre verticale plus grande que les barres horizontales, le degré d'appartenance dans la classe ω_1 est ainsi plus important (0.7). Enfin, les objets $\{+, \perp\}$ n'ayant pas une caractéristique prépondérante auront un degré d'appartenance équiprobable ($\frac{1}{c}$). Cependant, cette même valeur sera donné aux objets dont l'ignorance est totale ici / . Les objets atypiques comme o ne sont pas modélisables.
Ainsi la partition floue pour l'ensemble $\boldsymbol{X}$ est décrite dans le tableau 2.3.

Objets	ω_1	ω_2
$\mid$	1	0
$\parallel$	1	0
$\equiv$	0	1
$\_$	0	1
$[$	0.7	0.3
$+$	0.5	0.5
$\perp$	0.5	0.5
$/$	0.5	0.5
o	??	??

TABLEAU 2.3 – Partition floue à deux classes.

Le modèle probabiliste est défini sur un univers fermé qui ne peut pas s'ouvrir : la somme des degrés d'appartenance est égale à un . Il ne permet pas de prendre en compte les objets atypiques. Il induit une ambiguïté, il n'y a pas de distinction entre équiprobabilité et ignorance totale. Il ne peut pas définir l'imprécision.

2.2.3 Autres modèles flous

Possibilitic C-Means (PCM)

La théorie des possibilités a été introduite en 1978 par Zadeh [36], en liaison avec sa théorie des sous-ensembles flous, pour modéliser l'incertitude sans utiliser les probabilités (FCM). Dans *Fuzzy Clustering : A Historical Perspective*, Ruspini et al. [37] mettent en lumière la vision de Zadeh pour définir l'incertitude et ses applications en classification non supervisée.
Krishnapuram et Keller [38] sont les premiers à adapter cette théorie à HCM avec leur modèle PCM. Ils suppriment la contrainte probabiliste 2.2.2 qui forçait les valeurs aberrantes à appartenir à une ou plusieurs classes et dégradait leur qualité de partitionnement, $\sum_{j=1}^{c} u_{ij} = 1, \quad \forall i \in \{1, n\}$.
La partition floue $\boldsymbol{U}$ est remplacée par une partition possibiliste $\boldsymbol{P}$.

Définition 2.2.6: Partition possibiliste

Soit la matrice réelle $\boldsymbol{P}$ de dimension $(n \times c)$ définie :

$$\forall i \in \{1, n\}, \forall j \in \{1, c\}, \quad 0 \leq p_{ij} \leq 1. \tag{2.14}$$

p_{ij} est le degré de possibilité de l'objet $\boldsymbol{x}_i$ dans la classe ω_j noté $P_{\omega_j}(\boldsymbol{x}_j)$.
$\boldsymbol{P}$ est une **partition possibiliste** si elle respecte l'unique contrainte :

$$\sum_{i=1}^{n} p_{ij} \quad > 0 \quad \forall j \in \{1, c\}. \tag{2.15}$$

L'**ensemble des partitions possibilistes** est noté :

$$M_{pp} = \left\{ \boldsymbol{P} \in \mathbb{R}^{n \times c} | p_{ij} \in [0,1]; \sum_{i=1}^{n} p_{ij} > 0 \quad \forall j \right\}. \tag{2.16}$$

La fonction objectif de PCM est

$$J_{PCM}(\boldsymbol{P}, \boldsymbol{\mathcal{V}}) = \sum_{i=1}^{n} \sum_{j=1}^{c} p_{ij}^{m} d_{ij}^{2} + \sum_{j=1}^{c} \eta_j \sum_{i=1}^{n} (1 - p_{ij})^{m}, \tag{2.17}$$

$$\text{sous les contraintes (2.9) et (2.11)}, \tag{2.18}$$

où $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_c)$ sont des constantes définies par l'utilisateur. Si $\boldsymbol{\eta} = (0, \ldots, 0)$ alors les fonctions objectifs de PCM et FCM sont similaires. Les termes $1 - p_{ij}$ sont les complémentaires de p_{ij} évitant la solution triviale $p_{ij} = 0$. Le choix des nouveaux hyperparamètres $\boldsymbol{\eta}$ rend ce modèle difficile à paramétrer. De surcroît, malgré les suggestions de Krishnapuram et Keller [39], le modèle ne contraint pas suffisamment les degrés de possibilité. Lorsque $\boldsymbol{\eta}$ est grand, il est difficile de gérer le compromis entre valeurs aberrantes et valeurs non aberrantes : les objets se comportent presque indépendamment les uns des autres. Pal et al. ont proposé un modèle PFCM [40, 41] pour exploiter les avantages des modélisations possibiliste et probabiliste.

Illustration et limites de PCM

Reprenons l'exemple 2.1.2 et appliquons le au modèle possibiliste.

Exemple 2.2.5: Ensemble symbole en 2 classes - partition floue

Pour rappel, nous souhaitons regrouper $\boldsymbol{X} = \{|, \|, \equiv, \_, [, +, \perp, /, o\}$ en deux classes $\Omega = \{\omega_1, \omega_2\}$, les barres verticales et les barres horizontales.
Pour les objets $\{|, \|, \equiv, \_\}$, la partition possibilistique est identique aux partitions dure et floue référencée, voir les tableaux 2.1 et 2.3.
Désormais pour les objets $\{+, \perp\}$ équiprobables, nous pouvons traduire la possibilité d'appartenir à la classe une et à la classe deux par des degrés de possibilité égaux à 1. Et les objets atypiques, dont l'ignorance est totale $\{/, o\}$ ont des degrés de possibilité nuls.
Ainsi la partition possibiliste pour l'ensemble $\boldsymbol{X}$ est décrite dans le tableau 2.4.

Objets	ω_1	ω_2
$\mid$	1	0
$\parallel$	1	0
$\equiv$	0	1
$\_$	0	1
$[$	1	0.3
$+$	1	1
$\perp$	1	1
$/$	0	0
o	0	0

TABLEAU 2.4 – Partition possibilistique à deux classes.

Rough C-Means (RCM)

La théorie des ensembles approximatifs a été développée par Pawlak [42]. L'idée centrale est de séparer les objets discernables des objets indiscernables et d'attribuer à chaque objet un degré d'appartenance supérieur et inférieur pour chaque classe. Par conséquence chaque classe est également caractérisée par une approximation inférieure et supérieure. Cette théorie a d'abord été appliquée à HCM par Lingras et West [43], Rough C-means (RCM). Puis Maji et Pal l'ont étendue à FCM donnant FRCM [44]. De nombreux travaux ont été réalisés avec cette théorie, ses applications sont présentées par Lingras et al. [45].

2.3 Variante évidentielle

Les modèles de la section précédente ont été développés pour modéliser l'incertitude, afin d'améliorer la prise de décision. Nous avons pu illustrer les limites de ces modèles notamment dans l'analyse de l'imprécision.
Dans cette section, nous étudions un modèle qui décompose le traitement de l'information et la prise de décision. L'objectif est de définir l'imprécision pour qu'un expert apporte des connaissances supplémentaires.

2.3.1 Théorie des fonctions de croyance

La théorie des fonctions de croyance a été introduite par Shafer en 1976 [46] d'après les travaux de Demspter [47]. Le modèle a évolué jusqu'au formalisme proposé par Smets et Kennes [48] : le Modèle des Croyances Transférables. Il est construit sur deux niveaux :

— le niveau évidentiel : niveau où l'on traite l'information, où l'on raisonne avec les fonctions de croyance.
— le niveau pignistique : est le niveau de prise de décision. Le terme pignistique vient du latin *pignus* = une mise (a bet).

Une information est transformée en croyance à son juste niveau, si l'information concerne deux décisions possibles e_1, e_2 sans plus de précision sur l'une ou l'autre alors l'information sera conservée pour le sous-ensemble $e_1 \cup e_2$. Le transfert du niveau évidentiel au niveau pignistique, nécessite une transformation des fonctions de croyance en fonctions de probabilité.

Niveau évidentiel

En vue de décrire le niveau supérieur de la théorie des fonctions de croyance, nous avons besoin de définir l'ensemble puissance et la fonction de masse de croyance.

Définition 2.3.7: Ensemble puissance

Soit E le cadre de discernement, l'ensemble contenant tous les sous-ensembles de E, y compris l'ensemble vide $\emptyset$ et E lui-même, est appelé **ensemble puissance** noté 2^E.

Par exemple : Si $E = \{e_1, e_2\}$ alors l'ensemble puissance est $2^E = \{\emptyset, e_1, e_2, E\}$.

Définition 2.3.8: Fonction de masse de croyance

La fonction de masse de croyance normalisée associée à une information est une fonction de $2^E \longrightarrow [0,1]$ telle que

$$m(\emptyset) = 0,$$
$$\sum_{A \in 2^E} m(A) = 1.$$

La théorie des fonctions croyance attribue une masse de croyance $m(A)$ à chaque élément A de l'ensemble de puissance. Cette affectation placée dans A lui est propre, c'est-à-dire on ne peut pas l'affecter à un de ses sous-ensembles.
La masse $m(\emptyset)$ est appelée masse conflictuelle. Dans la vision probabiliste de Dempster, cette masse est nulle et la fonction de croyance est dite "normalisée".
En revanche, Smets reprend la notion de monde "ouvert" pour son modèle des croyances transférables et s'affranchit de la contrainte de normalisation. Alors la quantité $m(\emptyset)$ est interprétée comme la croyance que la décision n'est pas dans le cadre de discernement E.

Définition 2.3.9: Eléments focaux

Tous les sous-ensemble A portant une information, $m(A) > 0$, sont appelés **éléments focaux** de m.

La fonction m est dite
- Bayésienne : si les éléments focaux sont des singletons. Dans ce cas, avant même toute transformation, la fonction de croyance est une distribution de probabilité bayésienne.
- Catégorique : si toute la croyance est attribuée à un seul sous-ensemble, c'est-à-dire $m(A) = 1, A \in 2^E$.
- Vide : si $m(E) = 1$ (cas catégorique particulier). Cela représente l'absence totale d'information.

Nous parlons de connaissance totale lorsqu'un seul singleton de 2^E possède toute la croyance $m(e) = 1$. Dans ce cas m est bayésienne et catégorique.
A partir de la fonction de masse, nous pouvons définir les fonctions de crédibilité et de plausibilité.

Définition 2.3.10: Fonction de crédibilité

Soit m une fonction de croyance et soit $A, B \in 2^E$, la fonction de crédibilité, $bel : 2^E \longrightarrow [0,1]$,

$$bel(A) = \sum_{B \subset A, B \neq \emptyset} m(B) \tag{2.19}$$

Dans le cadre du monde fermé, elle vérifie les propriétés suivantes,

$$\begin{aligned} &Bel(\emptyset) = 0, \\ &Bel(E) = 1, \\ &Bel(\bigcup_{i=1}^{k} A_i) \geq \sum_{I \in \{1,\ldots,k\}, I \neq \emptyset} (-1)^{|I|+1} Bel(\bigcap_{i \in I} A_i), \quad \forall k \in \{2, 2^E\}. \end{aligned}$$

Définition 2.3.11: Fonction de plausibilité

La fonction de plausibilité, $pl : 2^E \longrightarrow [0,1]$,

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = Bel(\Omega) - Bel(\bar{A}), \tag{2.20}$$

$\bar{A}$ est le complémentaire de A.
Elle est la quantité de croyance qui ne contredit pas A et qui pourrait même lui être allouée sous réserve d'informations supplémentaires.
Dans le cadre du monde fermé, elle vérifie les propriétés suivantes,

$$\begin{aligned} &Pl(\emptyset) = 0, \\ &Pl(E) = 1, \\ &Pl(\bigcap_{i=1}^{k} A_i) \geq \sum_{I \in \{1,\ldots,k\}, I \neq \emptyset} (-1)^{|I|+1} Pl(\bigcup_{i \in I} A_i), \quad \forall k \in \{2, 2^E\}. \end{aligned}$$

Ces deux fonctions définissant les limites supérieure et inférieure d'un intervalle de probabilité dont le minimum est la crédibilité et le maximum la plausibilité : $Bel(A) \leq P(A) \leq Pl(A)$.
L'ignorance totale est représentée par une croyance vide, tel que $m(E) = 1$ puisque $\forall A \neq E, Bel(A) = 0$ et $Bel(E) = 1$. En ce point, la théorie des fonctions de croyance dissocie le cas de l'équiprobabilité et l'ignorance totale contrairement aux modèles probabilistes.

La règle de combinaison de Dempster, appelée aussi la règle de combinaison orthogonale est une méthode de fusion de sources d'information dans le cadre de la théorie

des fonctions de croyance. Elle est utilisée pour combiner les fonctions de masse d'évidence provenant de différentes sources d'information afin de parvenir à une fonction de masse d'évidence globale.

Définition 2.3.12: Règle de combinaison

Soit deux fonctions de croyance distinctes m_1, m_2, la règle de combinaison $\oplus$ est définie

$$m_{\oplus}(A) = \frac{1}{1-K} \sum_{B \cap C = A} m_1(B) m_2(C), \quad \forall A \in 2^E, \tag{2.21}$$

où $K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$ représente la masse conflictuelle. Cette masse correspond au degré de contradiction entre les deux fonctions de croyance, les deux sources d'information fusionnées. Lorsque les sources sont parfaitement en accord alors $K = 0$. De plus, cette règle respecte l'associativité, la commutativité et son élément neutre est la fonction de masse vide, $m_{\oplus}(\emptyset) = 0$.

A partir de la règle de combinaison, on peut définir les combinaisons de deux fonctions de croyance équivalentes aux opérateurs d'union et d'intersection.

Définition 2.3.13: Les combinaisons

La combinaison conjonctive entre m_1 et m_2 notée $m_{\cap}$:

$$m_{\cap}(A) = \sum_{B \cap C = A} m_1(B) m_2(C), \quad \forall A \in 2^E.$$

On remarque que $K = m_{\cap}(\emptyset)$.
La combinaison disjonctive entre m_1 et m_2 notée $m_{\cup}$:

$$m_{\cup}(A) = \sum_{B \cup C = A} m_1(B) m_2(C), \quad \forall A \in 2^E.$$

Niveau pignistique

Une fois le niveau évidentiel défini, nous pouvons prendre des décisions. Nous cherchons à atteindre le niveau pignistique. Nous passons de l'ensemble puissance au cadre de discernement. L'idée est de faire un pari sur la décision en transférant les croyances obtenues dans un modèle probabiliste, par le principe de *raison insuffisante*. La croyance affectée aux sous-ensembles qui ne sont pas des singletons $m(A)$ est du à un manque d'information. Cette quantité est répartie équitablement vers la distribution de probabilité des singletons de A.

Définition 2.3.14: Transformation pignistique

La transformation pignistique est une distribution de probabilité définie

$$BetP(\omega) = \sum_{A,\omega \in A} \frac{m^*(A)}{|A|}, \quad \forall \omega \in E, \tag{2.22}$$

avec $|A|$ la cardinalité du sous-ensemble A et m^* la fonction de croyance normalisée.

Dans le cas où $m(\emptyset) \neq 0$, il est nécessaire de transférer cette masse. Dempster propose de ne pas en tenir compte en utilisant la normalisation suivante :

$$m^*(A) = \begin{cases} \frac{m(A)}{1-m(\emptyset)} & \text{si } A \neq \emptyset, \\ 0 & \text{sinon.} \end{cases} \tag{2.23}$$

Yager propose de l'ajouter à $m(E)$ donc indirectement de répartir la masse sur tous les singletons [49] :

$$m^*(A) = \begin{cases} 0 & \text{si } A = \emptyset, \\ m(E) + m(\emptyset) & \text{si } A = E, \\ m(A) & \text{sinon.} \end{cases} \tag{2.24}$$

2.3.2 Evidential C-Means (ECM)

Masson et Denœux ont étendu FCM en appliquant le modélisme évidentiel, Evidential C-Means [11]. Le cadre de discernement est l'ensemble des classes Ω. Le regroupement proposé par le modèle est une partition crédale $\boldsymbol{M}$, m_{ij} représente la croyance allouée à l'affectation de l'objet $\boldsymbol{x}_i$ au sous-ensemble $\mathcal{A}_j \subseteq \Omega$, $j^{\text{ème}}$ élément de l'ensemble puissance 2^Ω.

Définition 2.3.15: Partition crédale

Soit la matrice réelle $\boldsymbol{M}$ de dimension $(n \times 2^c)$ définie :

$$\forall i \in \{1, n\}, \forall j \in \{1, 2^c\}, \quad 0 \leq m_{ij} \leq 1. \tag{2.25}$$

$\boldsymbol{M}$ est une **partition crédale** si elle respecte les contraintes :

$$\sum_{i=1}^{n} m_{i\ell} \quad > 0 \quad \forall \omega_\ell \in \Omega, \tag{2.26}$$

$$\sum_{j=1}^{2^c} m_{ij} \quad = 1 \quad \forall i \in \{1, n\}. \tag{2.27}$$

L'**ensemble des partitions crédales** est noté :

$$M_{cp} = \left\{ \boldsymbol{M} \in \mathbb{R}^{n \times 2^c} | m_{ij} \in [0,1]; \sum_{j=1}^{2^c} m_{ij} = 1 \quad \forall i; \sum_{i=1}^{n} m_{ij} > 0 \quad \forall j \right\}. \tag{2.28}$$

La partition crédale généralise les partitions [11] dure $\boldsymbol{H}$, floue $\boldsymbol{U}$, possibiliste $\boldsymbol{P}$.

Exemple 2.3.6: Partitionnement crédal

La figure 2.3.1 représente le partitionnement crédal obtenu sur l'exemple visuel testé préalablement dans l'exemple 2.1.1 pour HCM et dans l'exemple 2.2.3 pour FCM. Outre l'apparition d'un nouveau centroïde caractérisant le sous-ensemble, union des deux classes, nous retrouvons des frontières floues. Les fonctions de croyance associées aux trois sous-ensembles non vides de l'ensemble puissance sont représentées par les figures 2.3.2-2.3.3 et 2.3.4.

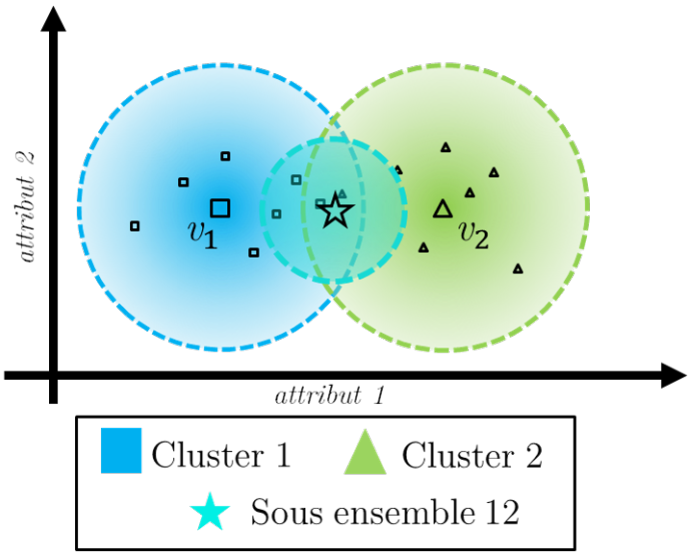


FIGURE 2.3.1 – Exemple de deux classes avec un sous-ensemble.

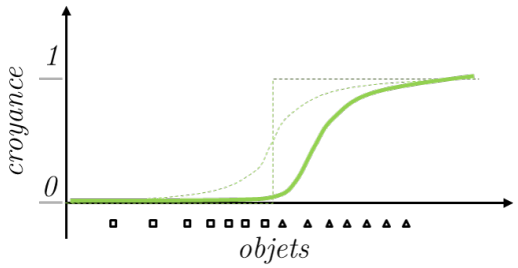


FIGURE 2.3.2 – Fonction de croyance de la classe 1.

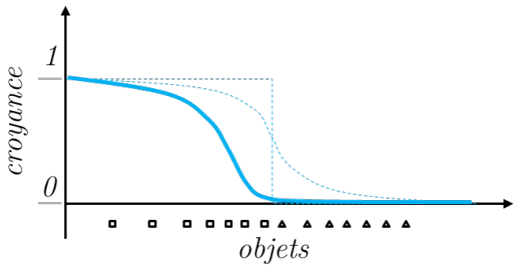


FIGURE 2.3.3 – Fonction de croyance de la classe 2.

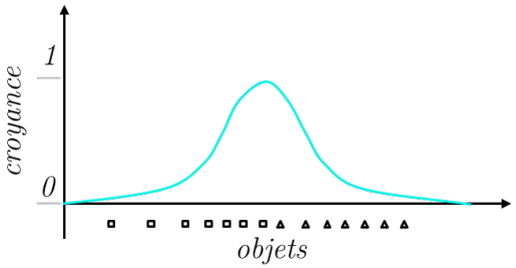


FIGURE 2.3.4 – Fonction de croyance du sous-ensemble 12.

D'après le modèle original [11], les centroïdes des sous-ensembles sont définis de manière automatique comme le barycentre des centroïdes des classes le constituant,

$$\boldsymbol{v}_j = \overline{\boldsymbol{v}}_j \triangleq \frac{1}{|\mathcal{A}_j|} \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{v}_\ell = \frac{1}{|\mathcal{A}_j|} \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{v}_\ell, \quad \forall j \in [1, 2^c] \tag{2.29}$$

où $s_{\ell j} = 1$ si $\omega_\ell \in \mathcal{A}_j$ sinon $s_{\ell j} = 0$.
La méthode ECM cherche $(\boldsymbol{M}, \boldsymbol{\mathcal{V}})$ minimisant les distances intra-classes :

$$J_{ECM}(\boldsymbol{M}, \boldsymbol{\mathcal{V}}) = \sum_{i=1}^{n} \sum_{\mathcal{A}_j \subseteq \Omega, \mathcal{A}_j \neq \emptyset} |\mathcal{A}_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^\beta, \tag{2.30}$$

sous les contraintes 2.26-2.27.

où $|\mathcal{A}_j|^\alpha$ est la cardinalité du sous-ensemble $\mathcal{A}_j$ à la puissance α, ce terme régule l'importance des sous-ensembles.
Plus $\alpha > 1$ est grand et plus les ensembles à forte cardinalité seront pénalisés. Dans ce sens, nous pouvons nous rapprocher de FCM. L'exposant $\beta > 1$ correspond au *paramètre de fuzzification* de FCM, il contrôle la dureté de la partition, usuellement fixé à 2. d_{ij} est la distance entre l'objet $\boldsymbol{x}_i$ et le sous-ensemble $\mathcal{A}_j$. A l'origine, la distance est euclidienne dans ECM [11], mais nous pouvons employer la distance de Mahalanoibis comme dans CECM [50].

ECM, utilise le formalisme proposé par Dave [51] pour les valeurs aberrantes. L'ensemble vide $\emptyset$ est utilisé pour contenir ces objets atypiques, renommé parfois classe de bruit. Le paramètre δ représente la distance constante entre chaque objet et l'ensemble vide.

Illustration et limites de ECM

Pour illustrer le modèle évidentiel, reprenons l'exemple 2.1.2 de l'ensemble des symboles.

Exemple 2.3.7: Ensemble symbole en 2 classes - partition crédale

Pour rappel, nous souhaitons regrouper $\boldsymbol{X} = \{|, \|, \equiv, \_, [, +, \perp, /, o\}$ en deux classes $\Omega = \{\omega_1, \omega_2\}$, les barres verticales et les barres horizontales.
Dans un raisonnement évidentiel, on ne s'intéresse pas seulement au cadre de discernement Ω, mais à l'ensemble puissance $2^{\Omega} = \{\emptyset, \omega_1, \omega_2, \Omega\}$. Pour les objets $\{|, \|, \equiv, \_\}$, la partition crédale est identique aux autres partitions puisque les objets appartiennent à une unique classe, voir le tableau 2.1.
Les objets $\{+, \perp\}$ équiprobables, sont des barres. Sans plus d'informations, nous ne pouvons pas dissocier à quelle classe ils appartiennent. Cette information se traduit par une croyance de 1 allouée à Ω. En revanche pour [, nous possédons l'information que la longueur de la barre verticale est plus longue que les barres horizontales, nous pouvons grâce à cette information accréditer une petite croyance pour son appartenance à ω_1.
Les objets atypiques $\{/, o\}$, dont l'ignorance est totale, sont affectés à l'ensemble vide. L'avantage du modèle évidentiel est le traitement de l'information à plusieurs niveaux de distinction. Par exemple pour / bien que ce soit ni une barre verticale, ni une barre horizontale, le fait qu'elle soit une barre peut être traduit par une petite croyance allouée à l'ensemble Ω.
Ainsi la partition crédale pour l'ensemble $\boldsymbol{X}$ est décrite dans le tableau 2.5.

Objets	$\emptyset$	ω_1	ω_2	Ω
$\|$	0	1	0	0
$\|\|$	0	1	0	0
$\equiv$	0	0	1	0
_	0	0	1	0
[	0	0.3	0	0.7
$+$	0	0	0	1
$\perp$	0	0	0	1
/	0.8	0	0	0.2
o	1	0	0	0

TABLEAU 2.5 – Partition crédale à deux classes.

2.4 Extensions de HCM et ses variantes

2.4.1 Enjeux et paramètres

HCM et ses variantes nécessitent un paramétrage spécifié par l'utilisateur pour le nombre de classes c, l'initialisation des classes et le choix de la distance. En classification non supervisée, nous ne disposons pas souvent de connaissance au préalable pour définir le paramétrage. Or un bon paramétrage est primordial pour avoir un bon partitionnement [20]. De surcroît, l'exécution de ces algorithmes est coûteuse en temps et en mémoire. Il est important de mettre en place des techniques sophistiquées plutôt que d'utiliser des processus en force brute pour déterminer le bon paramétrage.

Si certaines applications présupposent le nombre de classes, il y en a d'autres où le nombre de classes attendu n'est pas déterminé. Une question fondamentale se pose sur la valeur de l'hyperparamètre de ces modèles.

Quel est le nombre de classes c ?

Le processus standard, en force brute, pour déterminer ce nombre est d'exécuter HCM pour différentes valeurs c, puis une analyse d'après un critère choisi est réalisée. Le nombre de classes sélectionné est celui qui minimise le critère, ou celui obtenu par la méthode du coude. Cette dernière méthode est très répandue mais n'est pas très fiable d'après le récent travail de Schubert [52].
X-Means [53] vise à automatiser la détermination du nombre optimal de classes. L'algorithme explore différentes valeurs de c en subdivisant les classes et en conservant les meilleures divisions résultantes, jusqu'à ce qu'un critère tel que le critère d'information d'Akaike ou le critère d'information bayésien soit satisfait.
Cette méthode est plus rapide et efficace que le processus en force brute.

Comment initialiser l'algorithme ?

HCM est une heuristique qui converge vers un minimum local, ce qui signifie que différentes initialisations peuvent conduire à différentes partitions. Par conséquent, il est judicieux de procéder à plusieurs initialisations aléatoires et de conserver la partition qui minimise le mieux la fonction objectif. Des méthodes visant à améliorer le processus initial ont été suggérées, notamment les travaux de Fränti et al. [54].

2.4.2 Distances

La distance dans le modèle basé sur les centroïdes définit le concept de similarité. Son choix est donc essentiel, elle permet de donner plus ou moins d'importance aux attributs. La distance est dite "adaptative" si elle est spécifique à chaque classe.

Définition 2.4.16: Distance

Soit un ensemble $E \subseteq \mathbb{R}^n$, la distance d est une fonction de $E \times E \longrightarrow \mathbb{R}^+$ (réel positif ou nul) vérifiant,
- la symétrie : $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in E.$
- la séparation : $d(\boldsymbol{x}, \boldsymbol{y}) = 0 \Longleftrightarrow \boldsymbol{x} = \boldsymbol{y}, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in E.$
- l'inégalité triangulaire : $d(\boldsymbol{x}, \boldsymbol{z}) \geq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z}), \quad \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in E.$

(E, d) est un espace métrique.

2.4.2.1 Distances de l'espace vectoriel $\mathbb{R}^n$

La distance utilisée intialement dans HCM est la distance euclidienne, induite par la norme 2 de $\mathbb{R}^{n_d}$. Dans ce cas, chaque attribut a la même importance et la forme des classes détectée par le modèle est sphérique.

$$d_{ij}^2 = \| \boldsymbol{x}_i - \boldsymbol{v}_j \|_2^2 = \sum_{l=1}^{n_d} (\boldsymbol{x}_i)_l - (\boldsymbol{v}_j)_l = (\boldsymbol{x}_i - \boldsymbol{v}_j)^\top (\boldsymbol{x}_i - \boldsymbol{v}_j).$$

Les distances induites par les autres normes $(1, p, \infty)$ ont été testées : distance de Manhattan [55], Minkowski [56, 57], Chebyshev [58].

2.4.2.2 Distances adaptives

En supposant que chaque classe a des propriétés différentes, nous souhaitons utiliser des métriques différentes adaptées à chaque classe. Nous obtiendrons aussi des classes plus compactes.

La distance de Mahalanobis [59,60] évalue la similarité entre un objet et un centroïde en tenant compte de la corrélation des objets appartenant à cette classe selon leurs attributs. Elle accorde plus d'importance aux attributs prédominants et moins d'importance aux attributs les plus dispersés. Soit l'objet $\boldsymbol{x}_i$ et la classe j représentée par son centroïde $\boldsymbol{v}_j$ et sa matrice de variance-covariance $\boldsymbol{\Sigma}_j$, la distance de Mahalanobis est définie :

$$d_{ij}^2 = (\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{v}_j).$$

Dans leur travail [61], Gustasfon et Kessel retrouvent cette distance en proposant d'appliquer la métrique suivante :

$$d_{ij}^2 = (\boldsymbol{x}_i - \boldsymbol{v}_j)^T \boldsymbol{S}_j (\boldsymbol{x}_i - \boldsymbol{v}_j),$$

avec $\boldsymbol{S}_j$ une matrice strictement définie positive. Le modèle FCM-GK cherche $\boldsymbol{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}$

minimisant les distances intra-classes :

$$J_{GK}(\boldsymbol{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}) = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m} d_{ij}^{2}, \tag{2.31}$$

sous les contraintes (2.9)-(2.11) et

$$\det(\boldsymbol{S}_j) = \rho_j, \quad \forall j \in \{1, c\}. \tag{2.32}$$

D'après l'optimisation détaillée dans la section 3.2.2.1, ils obtiennent que la matrice $\boldsymbol{S}_j = \boldsymbol{\Sigma}_k^{-1} \succ 0$ correspond à l'inverse de la matrice de variance-covariance floue associée à la classe j :

$$\boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{n} u_{ij}^{2}(\boldsymbol{x}_i - \boldsymbol{v}_j)(\boldsymbol{x}_i - \boldsymbol{v}_j)^{\top}}{\sum_{i=1}^{n} u_{ij}^{2}}. \tag{2.33}$$

Ainsi, nous retrouvons la distance de Mahalanobis. Elle permet la détection de classe à forme ellipsoïdale et non plus seulement sphérique comme le démontre la figure 2.4.1.

Dans l'intention d'éviter, la solution triviale $\boldsymbol{S}_j = 0$, il est nécessaire de rajouter la contrainte (2.32) sur cette matrice. Gustasfon et Kessel fixent le volume des matrices par défaut à 1, $\rho_j = 1$. Dans d'autres travaux, Liu et al. [62] ajoutent à la fonction objectif $-log(det(\boldsymbol{\Sigma}_j^{-1}))$ pour enlever cette contrainte. Enfin Rammal et al. [63] pondèrent la matrice de covariance.

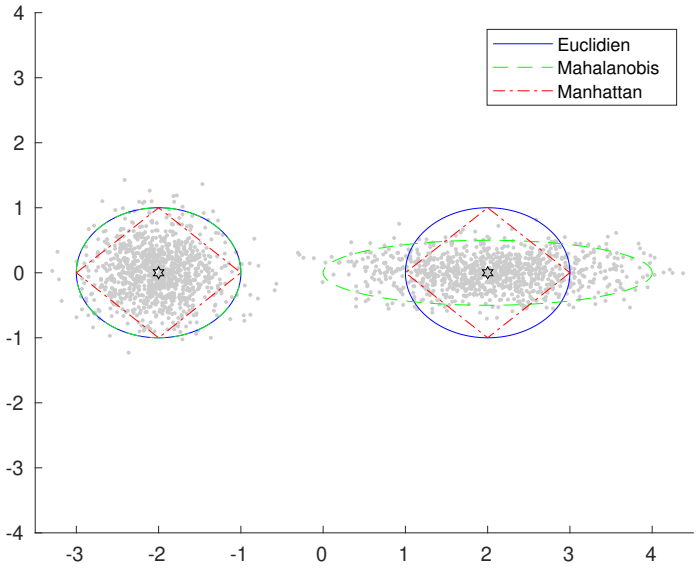


FIGURE 2.4.1 – Une classe à forme sphérique et une classe à forme ellipsoïdale et la forme de distance unitaire pour les 3 distances (euclidienne, Mahalanobis et Manhattan).

Une autre distance adaptative introduite par Gath et Geva [64] est la distance exponentielle basée sur l'estimation du maximum de vraisemblance :

$$d_{ij}^2 = \frac{n * det(\mathbf{\Sigma}_j)^{\frac{1}{2}}}{\sum_{i=1}^{n} u_{ij}} exp((\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \mathbf{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{v}_j)/2).$$

Cette distance suppose une distribution gaussienne ce qui n'est pas forcément vrai. Aussi les performances de cette distance sont mitigées [63].

Il existe d'autres travaux sur de nouvelles métriques [65,66] qui montrent l'importance du concept de similarité. De plus, Kumar et al. [65] ont consacré une étude sur le développement de distance tolérante au bruit et aux valeurs aberrantes.

Dans le tableau 2.6, nous récapitulons les principales distances déjà testées dans la littérature. Les citations renvoient soit aux méthodes d'origine, soit à des études comparatives. HCM est généralement appliqué avec la distance euclidienne [20]. En l'état de nos connaissances pour FCM, la distance euclidienne et la distance de Mahalanobis sont les plus utilisées. De plus, il est envisageable que la détection des classes à forme d'ellipse avec chevauchement offre une plus grande possibilité de partitionnement. Ainsi, dans certains domaines comme en analyse d'imagerie médicale, la distance de Mahalanobis semble plus précise et moins sensible au bruit [67, 68].

distance	HCM	FCM	ECM
Euclidienne	[30]	[5]	[11]
Mahalanobis	[69]	[57, 61]	[50]
Manhattan	[55]		
Minkowski	[56]	[57]	
Chebyshev		[58]	
Exponentielle		[64]	

Tableau 2.6 – Différentes distances employées par HCM, FCM et ECM.

2.4.3 Kernel K-Means

Toutes les extensions précédentes permettent d'obtenir différentes formes de classes mais toutes sont des applications linaires. Grâce à l'astuce du noyau, *kernel trick*, il est possible à un classifieur linéaire de détecter des classes à forme non linéaire. Cette astuce a été appliquée à différentes méthodes d'apprentissage [70]. Scholkopf et al. ont proposé une extension de HCM, Kernel K-Means [71]. La méthode projette les données dans une autre dimension grâce à une fonction noyau *kernel* pour rendre les données séparables linéairement. Le choix de la nature du noyau, gaussien, polynomial, ou autre doit être défini par l'utilisateur. Cependant, le noyau le plus approprié pour un jeu de données quelconque n'est pas connu à l'avance et la performance de la méthode dépend largement de ce choix. Pour atténuer cette dépendance, l'apprentissage à noyaux multiples combinant plusieurs noyaux a été proposé pour HCM [72].

2.5 Mesures d'évaluation

Une mesure d'évaluation quantifie la performance d'un modèle prédictif.

Pour valider un modèle, il est essentiel d'examiner les résultats obtenus. En premier lieu, il est crucial de vérifier que la fonction a été correctement minimisée. Cependant, nous ne pouvons pas nous en contenter car cette analyse se limite à un seul critère. Dans le cas de HCM et ses variantes, le critère utilisé est lié à la notion de compacité, puisque la fonction est la somme des distances intra-classes.
La classification non supervisée est un problème mal posé. Il n'existe pas un unique regroupement possible, la définition mathématique de la compacité et la séparabilité n'est pas unique, ainsi il n'existe pas de solution unique.

Ainsi, il est nécessaire d'utiliser plusieurs critères, des mesures de validation pour évaluer la qualité des résultats. En classification non supervisée, nous cherchons à évaluer la qualité de la partition soit en la comparant à une autre partition (évaluation externe), soit en analysant ses variables (évaluation interne), en vérifiant l'homogénéité des classes et/ou leur séparation adéquate les unes des autres.

2.5.1 Mesures d'évaluation externe

L'objectif de l'évaluation externe est de déterminer l'adéquation des résultats obtenus d'une méthode par rapport aux données réelles. Cette évaluation est surtout utilisée en classification supervisée. La classification obtenue par le modèle est comparée avec les données étiquetées. Les informations externes sont indépendant du modèle de classification choisi, c'est une évaluation extérieure soumise à aucun biais. Les mesures externes s'appuient sur la matrice de confusion, le tableau 2.7 présente la matrice pour une classification binaire, deux classes. Chaque ligne de la matrice représente les instances d'une classe réelle tandis que chaque colonne représente les instances d'une classe estimée (prédite), ou vice versa.

		Classe estimée	
		Positif	Négatif
Classe réelle	Positif	Vrai positif (VP)	Faux négatif (FN)
	Négatif	Faux positif (FP)	Vrai négatif (VN)

Tableau 2.7 – Matrice de confusion.

2.5.1.1 Évaluation externe pour une partition dure

En apprentissage non supervisé, les classes sont non nominatives. Les étiquettes associées aux classes sont des nombres arbitraires. Pour deux partitions différentes, la même classe peut être associée à deux nombres différents. La comparaison de deux partitions demande donc de mettre en correspondance ces nombres. Ce processus pouvant être complexe, les mesures d'évaluation interne s'intéressent aux paires d'objets. Les

mesures d'évaluation interne s'intéressent aux paires d'objets. Soient deux partitions, la partition de référence π_r et la partition du modèle π_m. La matrice de confusion s'appelle désormais la matrice appariement (*matching matrix*) :

— Vrai positif : la paire d'objets similaires (c'est-à-dire ayant la même classe dans la partition de référence : π_r) est dans la même classe (de la partition prédite : π_m).
— Vrai négatif : la paire d'objets dissimilaires (dans π_r) n'est pas la même classe (dans π_m).
— Faux positif : la paire d'objets dissimilaires (dans π_r) est dans la même classe (dans π_m).
— Faux négatif : la paire d'objets similaires (dans π_r) est dans des classes différentes (dans π_m).

On note a et b le nombre de vraies décisions positives et négatives, c et d le nombre de fausses décisions positives et négatives. Un jeu de n objets aura alors 2 parmi n : $\binom{n}{2}$ paires d'objets différents.

Il existe de nombreuses mesures statistiques combinant a, b, c et d. Parmi elles, deux mesures populaires en classification peuvent être utilisées en classification non supervisée :

— *La précision* (P) mesure la précision des résultats, le ratio entre le nombre de décisions vraies correctement estimées et le nombre de toutes les décisions estimées vraies :

$$P(\pi_r, \pi_m) = \frac{a}{a+c}. \tag{2.34}$$

— *Le rappel* (R) est le ratio entre le nombre de décisions vraies correctement estimées et le nombre de toutes les décisions vraies attendues :

$$R(\pi_r, \pi_m) = \frac{a}{a+d}. \tag{2.35}$$

Nous souhaitons que la partition apparie tous les objets similaires ($R = 1$), sans faire aucune erreur d'association ($P = 1$).

Chaque mesure permet d'évaluer un aspect différent. En classification non supervisée, l'objectif est de vérifier l'exactitude de la méthode, c'est-à-dire la proximité des résultats par rapport aux valeurs réelles, on parle aussi de taux d'accord. Ainsi la mesure principalement utilisée en classification non supervisée est l'indice introduit par Rand [73], aussi appelé *Simple matching coefficient*, qui correspond à la mesure de l'exactitude (*accurency*) en classification supervisée.

$$RI(\pi_r, \pi_m) = \frac{a+b}{a+b+c+d} = \frac{a+c}{\binom{n}{2}}. \tag{2.36}$$

L'indice de Rand RI comme P et R est compris entre 0 et 1, la valeur 1 (respectivement 0) correspond au fait que les partitions sont complètement identiques (respectivement totalement différentes). Cependant une certaine concordance entre deux partitions peut se produire par hasard, Arabie et Hubert [74] ont ajusté l'indice pour tenir compte

de cette éventualité, *ARI Ajusted Rand Index*. Cette mesure, recentrée entre -1 et 1, ou 0, correspond au cas de concordance aléatoire. Le centrage est réalisé par la formule suivante

$$\text{Indice ajusté} = \frac{\text{Indice} - \text{Indice supposé}}{\text{Indice maximal} - \text{Indice supposé},}$$

où Index Supposé correspond à la valeur de l'indice obtenu dans le cas de concordance aléatoire. La valeur proposée par Arabie et Hubert [74] a été reformulée par Hoffman et al. [75] :

$$\mathbb{E}(RI) = \frac{(a+d)(a+c)+(c+b)(b+d)}{(a+b+c+d)^2}.$$

D'où,

$$ARI(\pi_r, \pi_m) = \frac{RI - \mathbb{E}(RI)}{1 - \mathbb{E}(RI)} = \frac{2(ab-cd)}{(a+d)(d+b)+(a+c)(c+b)}. \tag{2.37}$$

2.5.1.2 Évaluation externe pour une partition floue ou crédale

Campello a généralisé l'indice de Rand aux partitions floues, Fuzzy Rand Index (FRI) [76]. Plus récemment, Denœux et al. [77], ont étendu RI aux partitions crédales : Credal Rand Index (CRI). L'indice mesure la ressemblance entre deux partitions à l'aide de la distance de Jousselme calculée pour les $\frac{n(n-1)}{2}$ paires d'objets. Considérons la paire d'objets i, l, ainsi que leurs fonctions de croyance $m_{i.}$ et $m_{l.}$ issues du partitionnement crédal. Notre objectif est de déterminer si cette paire d'objets appartient à la même classe, ce qui est représenté par l'événement s, ou s'ils appartiennent à deux classes différentes, représenté par l'événement $\neg s$. Pour cela, nous définissons l'ensemble de discernement $\Theta = \{s, \neg s\}$. Par la règle de combinaison de Dempster sur $m_{i.}$ et $m_{l.}$, on peut définir la fonction de croyance de $\Theta, \tilde{M}_{il}$:

$$\begin{aligned}
\tilde{m}_{il}(\emptyset) &= m_{i\emptyset} + m_{l\emptyset} - m_{i\emptyset} m_{l\emptyset}, \\
\tilde{m}_{il}(\{s\}) &= \sum_{k=1}^{c} m_{i\omega_k} m_{l\omega_k}, \\
\tilde{m}_{il}(\{\neg s\}) &= \sum_{A \cap B = \emptyset} m_{iA} m_{lB} - \tilde{m}_{il}(\emptyset), \\
\tilde{m}_{il}(\Theta) &= \sum_{A \cap B \neq \emptyset} m_{iA} m_{lB} - \tilde{m}_{il}(\{s\}).
\end{aligned}$$

Soit la représentation relationnelle associée est le vecteur

$$\mathbf{m}_{il} = (m_{il}(\emptyset), m_{il}(\{s\}), m_{il}(\{\neg s\}), m_{il}(\Theta)).$$

La distance de Jousselme δ_{il} de la paire d'objet $\{i, l\}$ entre la partition π_m ($\mathbf{m}_{il}^m$) et la partition π_r ($\mathbf{m}_{il}^r$) est,

$$\delta_{il} \quad = \quad \sqrt{\frac{1}{2}(\mathbf{m}_{il}^r - \mathbf{m}_{il}^m)^\top \mathbf{J} (\mathbf{m}_{il}^r - \mathbf{m}_{il}^m)}, \tag{2.38}$$

avec la matrice

$$\mathbf{J} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}.$$

Ainsi l'indice entre les deux partitions crédales est

$$CRI(\pi_r, \pi_m) = 1 - 2\frac{\sum_{1 \leq i < l \leq n} \delta_{il}^2}{n(n-1)}. \tag{2.39}$$

Nous pouvons utiliser cet indice avec des partitions dures et floues. Si la partition est dure, la fonction de croyance $\tilde{M}_{il}$ est définie

$$\begin{aligned} &\tilde{m}_{il}(\{s\}) = \begin{cases} 1 & \text{si ils sont dans la même classe,} \\ 0 & \text{sinon,} \end{cases} \\ &\tilde{m}_{il}(\{\neg s\}) = 1 - \tilde{m}_{il}(\{s\}), \\ &\tilde{m}_{il}(\emptyset) = \tilde{m}_{il}(\Theta) = 0. \end{aligned}$$

Si la partition $\boldsymbol{U}$ est floue , la fonction de croyance $\tilde{M}_{il}$ est définie

$$\begin{aligned} &\tilde{m}_{il}(\{s\}) = \sum_{j=1}^{c} u_{ij} u_{lj}, \\ &\tilde{m}_{il}(\{\neg s\}) = 1 - \tilde{m}_{il}(\{s\}), \\ &\tilde{m}_{il}(\emptyset) = \tilde{m}_{il}(\Theta) = 0. \end{aligned}$$

Avec un formalisme un peu plus simple, Zhou et al. ont proposé l'extension de la précision (P), rappel (R) et le Rand Index (RI) au partitionnement évidentiel : EP, ER et ERI [78]. L'approche consiste à examiner les correspondances entre chaque élément de l'ensemble de puissance, plutôt que de se limiter à chaque classe individuellement. Pour la comparaison de deux partitions probabilistes, Quéré donne plus de détails dans sa thèse notamment pages 26-27 [79].

2.5.2 Mesures d'évaluation interne

L'évaluation interne utilise les variables du modèle définissant le regroupement. Elle consiste à calculer un indice mesurant l'adéquation du regroupement aux données et non à des étiquettes (mesure externe). Nous souhaitons avoir un bon partitionnement, c'est-à-dire des classes homogènes et bien séparées les unes des autres. Cela fait référence à deux notions : la compacité et la séparabilité.

La compacité évalue la proximité des objets au sein d'une même classe avec des distances intra-classes.

La séparabilité détermine à quel point une classe est bien séparée des autres avec des distances inter-classes comme les distances entre les centres des classes, les distances minimales par paire entre les objets de différentes classes... Certaines mesures évaluent seulement la compacité ou la séparabilité, d'autres évaluent les deux en même temps.

2.5.2.1 Évaluation interne pour une partition dure

Nous énumérons les mesures les plus utilisées pour l'évaluation la partition dure générée par HCM.

— L'indice de Dunn D [80] est le rapport séparabilité sur compacité, la compacité étant définie comme la distance maximum qui sépare deux objets d'une même classe, et la séparabilité comme la distance minimum qui sépare deux classes en prenant leur centre de gravité.

$$D = \frac{\min\limits_{1 \leqslant j < j' \leqslant c} dist(\boldsymbol{v}_j, \boldsymbol{v}'_j)}{\max\limits_{1 \leqslant j \leqslant c} \Delta_j}, \tag{2.40}$$

où $\Delta_j = \max_{i,i' \in \omega_j} dist(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ est le diamètre de la classe j. La distance $dist$ n'est pas spécifique. Pour HCM, la distance euclidienne est appliquée. L'objectif est de maximiser cet indice compris entre $[0, +\infty]$.

— L'indice de Davies-Bouldin DB [81] est le rapport compacité sur séparabilité, la compacité étant définie comme la distance moyenne entre le centre de la classe et ses objets, et la séparabilité, comme pour D, est la distance minimum qui sépare deux classes compris.

$$DB = \frac{1}{c} \sum_{j=1}^{c} \underset{j \neq j'}{max} \left(\frac{\delta_j + \delta'_j}{dist(\boldsymbol{v}_j, \boldsymbol{v}'_j)} \right), \tag{2.41}$$

avec δ_j étant la distance moyenne entre les points et le centre de gravité de la classe j,

$$\delta_j = \frac{1}{|\omega_j|} \sum_{i \in \omega_j} dist(\boldsymbol{x}_i, \boldsymbol{v}_j), \tag{2.42}$$

où $|\omega_j|$ est la cardinalité de la classe ω_j, le nombre d'objets affectés à cette dernière. L'objectif est de minimiser cet indice compris entre $[0, +\infty]$.

Pour le partitionnement dur, Qiao et Edwards [82] ainsi que Bezdek et Nal [83] donnent plus de détails et des outils de compréhension.

2.5.2.2 Évaluation interne pour une partition floue ou crédale

Dans le cas des algorithmes à partitionnement flou, les deux premiers indices ont été proposés par Bezdek [84] qui a étudié la séparabilité de la partition floue U,

— L'indice Partition Coefficient PC [84] mesure le degré de chevauchement entre les classes floues donc évalue la séparabilité.

$$PC = \frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^2. \tag{2.43}$$

L'objectif est de maximiser PC compris entre $[\frac{1}{c}, 1]$. L'indice atteint la borne inférieure $\frac{1}{c}$ lorsque l'incertitude est totale $\forall i, j \quad u_{ij} = \frac{1}{c}$. A contrario, la borne supérieure 1 est obtenue lorsque la partition est dure. Dave a proposé une extension du coefficient permettant de le normaliser entre 0 et 1 [85] :

$$MPC = 1 - \frac{c}{c-1}(1 - PC). \tag{2.44}$$

— L'indice Partition Entropy PE [84] a été développé pour être moins sensible au nombre de classes c

$$PE = -\frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}\log_2(u_{ij}). \tag{2.45}$$

La mesure PE comprise entre $[0, \ln(c)]$ doit être minimisée, d'où $PC = \frac{1}{c} \iff PE = \log_2(c)$ et $PC = 1 \iff PE = 0$.

Ces indices mesures n'utilisent pas directement les informations de la modélisation de HCM et ses variantes, c'est-à-dire les propriétés géométriques contenues dans les distances et les centroïdes.

Il existe de nombreuses autres mesures qui combinent de différentes façons la compacité et la séparabilité comme le *Partition Coefficient And Exponential Separation PCAES* [86] ou plus récemment le *SMI* [87]. Les principaux indices sont référencés dans [87,88]. Parmi eux, se trouvent :

— Le Fuzzy Silhouette index FS [89], une extension du Silhouette index au partitionnement flou $\boldsymbol{U}$ qui permet la détection des régions à forte densité lorsque des chevauchements entre classes existent. Cet indice permet de valider la compacité d'une partition. Pour chaque objet i, nous gardons les deux classes avec le plus fort degré d'appartenance respectivement u_{ij_1}, u_{ij_2} pour $\omega_{j_1}, \omega_{j_2}$.

$$FS = \frac{\sum_{i=1}^{n}(u_{ij_1} - u_{ij_2})^{\alpha} sil(i)}{\sum_{i=1}^{n}(u_{ij_1} - u_{ij_2})^{\alpha}}, \tag{2.46}$$

avec $\alpha \geq 0$ un coefficient similaire au paramètre de fuzzification m de FCM qui permet de régler l'importance de la dureté ou non, généralement $\alpha = 1$. Et avec,

$$sil(i) = \frac{b_{j_2}(i) - a_{j_1}(i)}{\max(a_{j_1}(i), b_{j_2}(i))}. \tag{2.47}$$

En considérant, a_j la distance moyenne de l'objet i à la classe j_1 et b_{j_2} la distance moyenne de l'objet i à la classe j_2 :

$$a_{j_1}(i) = \frac{1}{|\omega_{j_1}| - 1}\sum_{i' \in \omega_{j_1}, i' \neq i} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}), \quad b_{j_2}(i) = \frac{1}{|\omega_{j_2}|}\sum_{i' \in \omega_{j_2}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}),$$

où la distance $d(.)$ est la distance euclidienne. C'est un indice à maximiser dans son intervalle de définition [-1,1]. Une partition ayant son indice FS égale à -1 est de très mauvaise qualité. En revanche, si l'indice est maximal, la partition est d'excellente qualité, les points aux seins des classes sont proches et les classes bien séparées.

— L'indice de *Xie-Beni* XB [6] est le rapport compacité sur séparabilité, la compacité est définie comme la fonction objectif de Fuzzy C-means (2.13) et la séparabilité, comme pour les indices de Dunn D et Davies-Bouldin DB, est la distance minimum qui sépare deux classes.

$$XB = \frac{\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{m} \parallel v_j - x_i \parallel^2}{n \times \underset{j \neq j'}{min}(\parallel v_j - v'_j \parallel^2)}. \tag{2.48}$$

Une meilleure partition minimise ce critère compris dans l'intervalle $[0, +\infty]$.
Contrairement au partitionnement flou, le partitionnement évidentiel est assez récent, il y a donc peu de méthodes et peu de critères d'évaluation adaptés pour ce formalisme. Masson et Denœux ont été les premiers à se confronter à ce problème quand ils ont crée le modèle ECM [11]. Ils ont développé la mesure de non-spécificité *nonspecificity* qui s'inspire de l'entropie floue (PE) :

$$N^* = \frac{1}{n \log_2 c} \times \sum_{i=1}^{n} \sum_{\mathcal{A}_j \neq \emptyset} m_{ij} \log_2 |\mathcal{A}_j| + m_{i\emptyset} \log_2 c. \tag{2.49}$$

N^* compris entre $[0, 1]$ doit être minimisé.
Chaque indice évalue différemment la partition obtenue selon un point de vue spécifique. Il est important lors d'une comparaison entre deux partitions d'étudier plusieurs indices [83]. Par ailleurs, les indices ayant des intervalles à bornes finis permettent de meilleures comparaison et description des résultats. Nous venons d'étudier ces indices comme outil de comparaison entre deux partitions puisque nous en aurons besoin pour l'évaluation de nos expérimentations. Cependant, ces indices permettent aussi de définir les valeurs des hyperparamètres : le nombre de classes c [86, 87], les paramètres de fuzzification m [35] et autres α, β, δ [11]...

2.6 Conclusion

Dans ce chapitre, les notions clés de la classification non supervisée ont été présentées pour introduire le modèle étudié Hard c-means (HCM), les c classes y sont représentées par leur centre de gravité (*means*), chaque objet appartient à une unique classe (Hard). Dans certaines situations, il existe des doutes pour le choix de la classe. Sous l'angle de la gestion de cette incertitude, nous avons pu dissocier les différentes variantes de HCM. Fuzzy C-means prend en compte l'incertitude en évaluant le degré d'appartenance de

chaque objet à chaque classe. Enfin, Evidential C-means utilise la théorie des fonctions de croyance pour spécifier l'imprécision. Le tableau 2.8 récapitule les éléments clés de ces trois modèles. Nous avons également étudié différentes extensions, telles que l'utilisation d'autres métriques et la gestion des hyperparamètres des modèles.

	Théorie	(Fonction)	Partition	Modélise	Citation
HCM	Logique de décision	(Booléenne)	Dure	Certitude	[30]
RCM	Ensembles approximatifs	(Approximation)	Dure	Imprécision	[43]
FCM	Logique floue	(Probabilité)	Floue	Incertitude	[5]
FRCM	Ensembles approximatifs	(Approximation)	Floue	Incertitude et imprécision	[44]
PCM	Ensembles flous	(Possibilité)	Floue	Incertitude et imprécision	[38]
ECM	Modèle de croyance transférable	(Masse)	Evidentielle	Incertitude et imprécision	[11]

Tableau 2.8 – Les différents modèles de classification non supervisée.

Avec pour objectif d'évaluer la qualité et la performance d'un algorithme de classification non supervisée, de nombreux indices ont été proposés. Nous avons analysé les principaux. Il existe deux grandes catégories, les indices internes qui utilisent seulement les variables de l'algorithme et les indices externes qui comparent deux partitions. Les tableaux 2.9-2.10 présentent les principaux indices s'ils sont à maximiser (↑) ou à minimiser (↓) dans un certain intervalle. La référence de leur formule présentée dans ce chapitre est donnée. Nous avons répertorié dans la dernière colonne les mesures internes utilisées pour ajuster les hyperparamètres des modèles [11, 35, 86, 87].

	(Direction)	Intervalle	Eq.	Citation
Partition dure				
P (Précision)	(↑)	$[0, 1]$	2.34	
R (Rappel)	(↑)	$[0, 1]$	2.35	
RI (Rand index)	(↑)	$[0, 1]$	2.36	[73]
ARI (Ajusted RI)	(↑)	$[-1, 1]$	2.37	[74]
Partition floue				
FRI (Fuzzy RI)	(↑)	$[0, 1]$		[76]
Partition crédale				
CRI	(↑)	$[0, 1]$	2.39	[77]
PE, RE (Précision)	(↑)	$[0, 1]$		[78]

Tableau 2.9 – Les différentes mesures d'évaluation externe.

	(Direction)	Intervalle	Eq.	Citation	Paramètres
Partition dure					
D	(↑)	$[0, +\infty]$	2.40	[80]	c
DB	(↓)	$[0, +\infty]$	2.41	[81]	c
Partition floue					
PC (Partition Coefficiant)	(↑)	$[\frac{1}{c}, 1]$	2.43	[84]	c, m
MPC (PC normalisé)	(↑)	$[0, 1]$	2.44	[85]	c
PE (Partition Entropy)	(↓)	$[0, \ln(c)]$	2.45	[84]	c, m
FS (Fuzzy Silhouette)	(↑)	$[-1, 1]$	2.46	[89]	c, m
XB	(↓)	$[0, +\infty]$	2.48	[6]	c, m
Partition crédale					
N^* (Nonspecificity)	(↓)	$[0, 1]$	2.49	[11]	c, α

TABLEAU 2.10 – Les différentes mesures d'évaluation interne.

Chapitre 3

Optimisation mathématique

Contents

L'objectif de l'optimisation mathématique est de trouver une solution qui maximise ou minimise une fonction objectif tout en satisfaisant un ensemble de contraintes. Les problèmes d'optimisation se retrouvent dans divers domaines d'applications, de l'économie à la physique en passant par les sciences de l'ingénieur, la logistique, la chimie... [90] La fonction objectif est souvent définie en terme de coûts, de revenus, de temps, de distances ou d'autres mesures quantifiables. Les contraintes peuvent inclure des exigences de capacité, des limites de temps, des exigences de qualité et d'autres contraintes spécifiques à chaque problème.

La nature d'un problème d'optimisation est déterminée par la caractéristique de la fonction (continue, convexe, différentiable, par bloc, etc.) et des contraintes (convexe, linéaires, quadratiques, égalités, inégalités, etc.). Nous distinguons respectivement l'optimisation continue et l'optimisation combinatoire selon la nature continue ou discrète de l'espace de recherche.

Les méthodes d'optimisation mathématique peuvent être classées en deux catégories principales : les méthodes déterministes et les méthodes stochastiques. Les méthodes déterministes sont utilisées pour résoudre des problèmes où toutes les données sont connues avec certitude. En revanche, les méthodes stochastiques sont employées pour résoudre des problèmes où les données sont incertaines ou sujettes à une certaine variabilité.

Parmi les problèmes d'optimisation courants, on trouve la programmation linéaire [91, 92], la programmation non linéaire [91, 93], la programmation quadratique [94], la programmation en nombres entiers [95], et la programmation génétique [96]. Les logiciels d'optimisation peuvent aider à résoudre ces problèmes en utilisant des algorithmes sophistiqués pour trouver la solution la plus efficace possible. Il existe des solveurs spécifiques tels que CPLEX pour la programmation linéaire et KNITRO pour la programmation non linéaire, ainsi que des bibliothèques disponibles dans les principaux langages de programmation comme *SciPy.optimize* pour Python et *Optimization Toolbox* pour MATLAB.

Dans le chapitre précédent, nous avons exploré plusieurs modèles de classification non supervisée et défini leur problème d'optimisation. L'objectif est de trouver le meilleur partitionnement possible en réalisant l'optimisation à l'aide de la méthode la plus appropriée. Notre étude se concentre sur trois méthodes classiques d'optimisation, dont l'optimisation alternée, qui est la méthode la plus couramment utilisée pour le HCM (Hard C-Means) et ses variantes [97]. De nombreuses méta-heuristiques ont été testées et une analyse détaillée a été réalisée par Nanda et Panda [16].

Dans ce chapitre, après avoir rappelé des notions importantes de l'optimisation (section 3.1), nous détaillons la méthode d'optimisation alternée actuellement utilisée en classification non supervisée (section 3.2). Ensuite, nous mettons en avant deux alternatives, deux méthodes d'optimisation (sections 3.3 et 3.4). Enfin, à l'aide d'un exemple, nous comparons ces trois méthodes (section 3.5).

3.1 Éléments théoriques

3.1.1 Définition du problème

Nous cherchons à minimiser le problème à n variables de décision, m contraintes d'égalité, et p contraintes d'inégalité formulé par :

$$\min f(\boldsymbol{x}), \tag{3.1}$$

sous les contraintes,

$$\boldsymbol{h}(\boldsymbol{x}) = 0, \tag{3.2}$$
$$\boldsymbol{g}(\boldsymbol{x}) \leq 0, \tag{3.3}$$
$$\boldsymbol{x} \in \mathbb{R}^n. \tag{3.4}$$

où la fonction objectif $f : \mathbb{R}^n \to \mathbb{R}$ est supposée continue et différentiable, et les contraintes $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^m$ et $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$. L'existence d'une solution est vraie, lorsqu'au moins f est continue et l'ensemble admissible est compact ou f coercive et l'ensemble admissible est fermé. Il faut remarquer qu'il suffit de changer le signe dans (3.1) pour avoir le problème de maximisation.

Définition 3.1.1: Solution réalisable

Un point $\boldsymbol{x}^* \in \mathcal{X}$ est une solution réalisable du problème, s'il respecte les contraintes du problème (3.2)-(3.3) :

$$\boldsymbol{h}(\boldsymbol{x}^*) = 0,$$
$$\boldsymbol{g}(\boldsymbol{x}^*) \leq 0.$$

L'ensemble des points réalisables, est appelé ensemble admissible noté ici, $\mathcal{X} = \{\boldsymbol{x} \in \mathcal{X} | \boldsymbol{h}(\boldsymbol{x}) = 0 \text{ et } \boldsymbol{g}(\boldsymbol{x}) \leq 0\}$.

Définition 3.1.2: Fonction coercive

La fonction $f : \mathbb{R}^n \to \mathbb{R}$ est coercive si

$$\lim_{\|\boldsymbol{x}\| \longrightarrow +\infty} f(\boldsymbol{x}) \longrightarrow +\infty.$$

Définition 3.1.3: Fonction bornée inférieurement

La fonction $f : \mathbb{R}^n \to \mathbb{R}$ est bornée inférieurement s'il existe un réel M tel que

$$f(\boldsymbol{x}) \geq M, \quad \forall \boldsymbol{x} \in \mathbb{R}^n.$$

La solution minimise la fonction soit localement, c'est un minimum local, soit sur tout l'ensemble de définition, c'est un minimum global.

Définition 3.1.4: minimum local-global

Le vecteur $\boldsymbol{x}^*$ est appelé **minimum local** si

$$\exists \mathcal{V} \text{ (un voisinage) }, f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \quad \forall x \in \mathcal{X} \cap \mathcal{V}.$$

Le vecteur $\boldsymbol{x}^*$ est appelé **minimum global** si

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \quad \forall x \in \mathcal{X}.$$

3.1.2 Notions de convexité

La nature de la fonction détermine l'existence et le nombre de minima locaux et globaux. En particulier lorsque la fonction est convexe, tout minimum local est un minimum global. De plus, si la convexité est stricte alors le minimum est unique.

Définition 3.1.5: Degrés de convexité

1. La fonction f est dite **convexe** sur $\mathcal{X}$ (ensemble convexe) si et seulement si

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}, \forall t \in [0,1], f(t\boldsymbol{x} + (1-t)\boldsymbol{y}) \leq tf(\boldsymbol{x}) + (1-t)f(\boldsymbol{y}). \tag{3.5}$$

   f est dite strictement convexe si l'inégalité stricte est vérifiée lorsque $\boldsymbol{x} \neq \boldsymbol{y}$ et $t \in]0,1[$.

2. Lorsque f est différentiable, elle est dite **pseudo-convexe** sur $\mathcal{X}$ si et seulement si

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}, \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \geq 0 \Longrightarrow f(\boldsymbol{y}) \geq f(\boldsymbol{x}). \tag{3.6}$$

   Une fonction différentiable est croissante dans toute direction où elle a une dérivée directionnelle positive. f est dite strictement quasi-convexe si l'inégalité stricte dans (3.6) est vérifiée lorsque $\boldsymbol{x} \neq \boldsymbol{y}$ et $t \in]0,1[$.

3. La fonction f est dite **quasi-convexe** sur $\mathcal{X}$ si et seulement si

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}, \forall t \in [0,1], f(t\boldsymbol{x} + (1-t)\boldsymbol{y}) \leq \max(f(\boldsymbol{x}), f(\boldsymbol{y})). \tag{3.7}$$

   f est dite strictement quasi-convexe si l'inégalité stricte est vérifiée lorsque $\boldsymbol{x} \neq \boldsymbol{y}$ et $t \in]0,1[$.

Les conditions nécessaires d'optimalité permettent de caractériser un minimum local : il est impossible de diminuer la fonction en partant d'un minimum. En présence des contraintes, un minimum local vérifie qu'il n'y a aucune direction admissible (direction respectant les contraintes) qui soit une direction descente.

3.1.3 Théorie de la dualité de Lagrange

La dualité en optimisation évoque l'étude du problème en utilisant le point de vue de ses contraintes. Il existe plusieurs formalismes, tels que celui de Wolfe [98] ou de Fenchel [99]. Étant donné que les problèmes que nous étudions impliquent des fonctions continues et dérivables, la dualité de Lagrange est la plus appropriée. Dans cette section, nous présentons uniquement les résultats nécessaires à notre étude, plus de détails sur l'analyse convexe et la théorie de la dualité sont données en [99–102].

La première étape de cette approche, la relaxation des contraintes, consiste à les incorporer à la fonction objectif. La nouvelle fonction ainsi obtenue est appelée la fonction de Lagrange, ou plus simplement, le Lagrangien.

Définition 3.1.6: Fonction de Lagrange

Soit le problème d'optimisation (3.1)-(3.3), soient $\boldsymbol{\lambda} \in \mathbb{R}^m$, $\boldsymbol{\mu} \in \mathbb{R}^p, \boldsymbol{\mu} \geq 0$ et la fonction $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\mu}^\top \boldsymbol{g}(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{\lambda}_i^\top \boldsymbol{h}_i(\boldsymbol{x}) + \sum_{j=1}^{p} \boldsymbol{\mu}_j^\top \boldsymbol{g}_j(\boldsymbol{x}). \quad (3.8)$$

est appelée Lagrangien ou fonction lagrangienne du problème d'optimisation (3.1)-(3.3). $\boldsymbol{\lambda}, \boldsymbol{\mu}$ sont appelés les multiplicateurs de Lagrange associés aux contraintres.

Définition 3.1.7: Problème primal et dual

Le problème d'optimisation (3.1)-(3.3) est le **problème primal** (P) noté

$$\inf_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}),$$

avec $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{h}(\boldsymbol{x}) = 0, \boldsymbol{g}(\boldsymbol{x}) \leq 0\}$.
Le **problème dual** de Lagrange associé (P') est,

$$\sup_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq 0} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}).$$

Il permet de se placer sous le point de vue des contraintes et offre une borne inférieur au problème primal. La direction de l'objectif est inversée, ainsi le minimum dans le primal devient le maximum dans le dual.

Les conditions nécessaires d'optimalité (CNO), également connues sous le nom de conditions de Karush-Kuhn-Tucker (KKT), sont un ensemble de conditions mathématiques qui décrivent les propriétés des solutions optimales pour les problèmes d'optimisation contraints.

Théorème 3.1.1: CNO Lagrange : contraintes d'égalité

1er ordre
Soit $\boldsymbol{x}^*$ un minimum local du problème (3.1)-(3.2) avec f, h continûment différentiables, alors

$$\exists \boldsymbol{\lambda}^* \in \mathbb{R}^m, \nabla\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0 \iff \begin{cases} \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\boldsymbol{x}^*) = 0, \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \boldsymbol{h}(\boldsymbol{x}^*) = 0. \end{cases} \tag{3.9}$$

2nd ordre
Si f, h sont deux fois différentiables

$$y^\top \nabla^2_{\boldsymbol{x}\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)y \geq 0, \forall y \in \mathcal{D}(\boldsymbol{x}^*), y \neq 0, \tag{3.10}$$

où $\mathcal{D}(\boldsymbol{x}^*)$ est le cône tangent en $\boldsymbol{x}^*$, $\mathcal{D}(\boldsymbol{x}^*) = \{\boldsymbol{d} | \boldsymbol{d}^\top \nabla h_i(\boldsymbol{x}^*) = 0, \forall i = 1, \ldots, m\}$.

Théorème 3.1.2: CNO KKT : contraintes d'égalité et d'inégalité

1er ordre
Soit $\boldsymbol{x}^*$ un minimum local du problème (3.1)-(3.3) avec f, h, g continûment différentiables, alors

$$\exists(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in (\mathbb{R}^m, \mathbb{R}^p), \boldsymbol{\lambda}^* \geq 0, \begin{cases} \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\boldsymbol{x}^*) + \sum_{j=1}^p \mu_j^* \nabla g_j(\boldsymbol{x}^*) = 0, \\ \boldsymbol{h}(\boldsymbol{x}^*) = 0, \\ \mu_j^* g_j(\boldsymbol{x}^*) = 0 \quad \forall j \in [1, p]. \end{cases} \tag{3.11}$$

La dernière condition est appelée condition de complémentarité. Deux cas sont possibles, soit $\mu_j^* = 0$ alors $g_j(\boldsymbol{x}^*) < 0$, soit $\mu_j^* > 0$ alors $g_j(\boldsymbol{x}^*) = 0$.
2nd ordre
Si f, h, g sont deux fois différentiables

$$y^\top \nabla^2_{\boldsymbol{xx}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) y \geq 0, \forall y \in \mathcal{D}(\boldsymbol{x}^*), y \neq 0. \tag{3.12}$$

où $\mathcal{D}(\boldsymbol{x}^*)$ est le cône de direction en $\boldsymbol{x}^*$.
$\mathcal{D}(\boldsymbol{x}^*) = \{\boldsymbol{d} | \boldsymbol{d}^\top \nabla h_i(\boldsymbol{x}^*) = 0, \forall i = 1, \ldots, m; \boldsymbol{d}^\top \nabla g_i(\boldsymbol{x}^*) \leq 0, \forall i = 1, \ldots, p\}$.

Grâce à toutes ces notions, nous pouvons maintenant caractériser les points en parlant de point stationnaire, point selle, solution d'un problème et ainsi développer une compréhension approfondie de la manière dont les solutions optimales sont identifiées.

Définition 3.1.8: Points et solutions

1. **Point stationnaire (ou point critique)**
   Tout (autre) point $\bar{\boldsymbol{x}}$ vérifiant les conditions du premier ordre est appelé point stationnaire. Ce point n'est pas nécessairement un minimum local.
2. **Solution**
   Les vecteurs $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ sont appelés solution duale du problème.
   Les vecteurs $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ sont appelés solution primal-duale du problème.
3. **Point selle**
   Si $\sup_{\boldsymbol{\lambda}^*, \boldsymbol{\mu}^* \geq 0} (\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)) = \inf_{\boldsymbol{x}^* \in \mathcal{X}} f(\boldsymbol{x}^*)$ alors $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ est appelé point selle du Lagrangien.

Théorème 3.1.3: Existence d'un point-selle

Avec f est convexe et C^1, les g_j convexes et C^1, les h_i affines, $\exists \boldsymbol{x}_o, \boldsymbol{h}(\boldsymbol{x}_o) = 0, \boldsymbol{g}(\boldsymbol{x}_o) < 0$, si le problème (3.1)-(3.3) admet une solution alors le Lagrangien $\mathcal{L}$ possède un point selle $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$.

Pour résoudre le problème, on peut dissocier les méthodes en fonction des variables qu'elles utilisent :

— Méthode primale : utilisation des variables primales comme variables principales, par exemple la méthode de descente de gradient, projection...
— Méthode duale : utilisation des variables duales comme variables principales, tandis que les variables primales deviennent des variables auxiliaires. Ce type de méthode tire son nom des travaux d'Uzawa [103].
— Méthode primale-duale : les variables primales et duales sont traitées avec la même importance.

3.2 Méthode d'optimisation alternée

L'optimisation alternée (*alternating optimization* : AO) est une approche couramment utilisée pour résoudre des problèmes de classification non supervisée notamment pour HCM Lloyd [30]. Bezdek et Hathaway [97] détaillent l'utilisation de cette méthode pour les problèmes d'optimisation non linaire avec des contraintes (3.1)-(3.3) telles que FCM. Afin de simplifier la résolution du problème FCM, le problème est décomposé en sous-problème plus faciles à résoudre. C'est un cas particulier de la méthode de Gauss-Seidel par blocs. Elle remplace la résolution simultanée sur l'ensemble des n variables par une suite de résolution sur des sous-ensembles de variable. L'ensemble de définition $\mathcal{X}$ est décomposé alors en s sous-ensembles $\mathcal{X} : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_s$ où $\mathcal{X}_j \subset \mathbb{R}^{s_j}$ avec $\sum_{j=1}^{s} s_j = n$. Ainsi $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{X}$ est décomposé en s vecteurs, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s)$, chaque variable est affectée une seule fois dans un vecteur. Le vecteur $\boldsymbol{x}_j \in \mathcal{X}_j \subset \mathbb{R}^{s_j}$ est composé de s_j variables.

Exemple 3.2.1: Décomposition de l'ensemble des variables

Supposons un problème avec cinq variables à l'origine $n = 5, \boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$, nous souhaitons le décomposer en deux sous-ensembles $s = 2, \boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$.
Nous pouvons, par exemple, choisir le vecteur $\boldsymbol{x}_1 = \big(x_2, x_3\big)$ ayant deux variables, $s_1 = 2$, et le vecteur $\boldsymbol{x}_2 = \big(x_1, x_4, x_5\big)$ ayant trois variables, $s_2 = 3$.
Nous vérifions que le nombre de variables est conservé : $s_1 + s_2 = 5 = n$ et qu'il n'y a pas de variable dans deux sous-ensembles différents.

Une fois la décomposition réalisée, les vecteurs de variables sont mis à jour successivement en utilisant les conditions d'optimalité. Le choix de cette décomposition est un moment important de l'application de l'optimisation alternée, il doit être réalisé afin de faciliter la résolution des sous-problèmes.

A chaque itération $k+1$, la méthode résout successivement les s sous-problèmes :

$$\boldsymbol{x}_j^{k+1} = \underset{\boldsymbol{x}_j}{\operatorname{argmin}}\ f(\boldsymbol{x}_1^{k+1}, \ldots, \boldsymbol{x}_{j-1}^{k+1}, \boldsymbol{x}_j, \boldsymbol{x}_{j+1}^{k}, \ldots, \boldsymbol{x}_s^{k}), \quad \forall j \in [1, s], \quad \forall k \geq 1.$$

Pour la $j$ème résolution, seules les variables du vecteur $\boldsymbol{x}_j$ ne sont pas fixées. Les sous-problèmes sont résolus en utilisant les conditions d'optimalité vues précédemment. Nous pouvons construire l'algorithme 2 de l'optimisation alternée.

Algorithme 2 AO *optimisation alternée.*

Itération $k = 0$: $\boldsymbol{X}^0$ donnée
Itération $k \geq 1$:
pour j de 1 à s **faire**
1: $\boldsymbol{x}_j^k = \underset{\boldsymbol{x}_j}{\operatorname{argmin}}\ f(\boldsymbol{x}_{<j}^k, \boldsymbol{x}_j, \boldsymbol{x}_{>j}^{k-1})$

Une solution obtenue par l'optimisation alternée $\boldsymbol{X}^* = (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_s^*)$ n'est pas nécessairement une solution du problème d'optimisation $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ [97].

3.2.1 Convergence de la méthode de Gauss-Seidel par bloc

Il est important de noter que la convergence d'une méthode peut être soit locale, soit globale. Si l'algorithme ne converge que lorsque la solution de départ est proche de la solution optimale, on parle de convergence "locale", c'est le cas pour la méthode de Newton. En revanche, on parle de convergence "globale" lorsque l'algorithme génère une suite convergente quel que soit le point de départ initial.
Nous avons besoin de définir deux autres notions supplémentaires pour caractériser la convergence de la méthode de Gauss-Seidel.

Définition 3.2.9: Ensemble niveau

L'ensemble niveau $\mathcal{L}_{\mathcal{X}}^0$ de la fonction f définie sur $\mathcal{X}$ donné pour un point $\boldsymbol{x}^0$, est défini comme l'ensemble des points de $\mathcal{X}$ dont l'image par la fonction f est inférieure ou égale à l'image du point de référence $\boldsymbol{x}^0$:

$$\mathcal{L}_{\mathcal{X}}^0 = \{\boldsymbol{x} \in \mathcal{X} | f(\boldsymbol{x}) \leq f(\boldsymbol{x}^0)\}.$$

Définition 3.2.10: Stricte quasiconvexité (selon une composante)

Soient $k \in \{1, \ldots, s\}$, f est dite strictement quasi-convexe selon la composante k si et seulement si

$$\forall \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y}_k \in \mathcal{X}_k, \boldsymbol{y}_k \neq \boldsymbol{x}_k, \forall t \in [0,1],$$
$$f(\boldsymbol{x}_1, \ldots, t\boldsymbol{x}_k + (1-t)\boldsymbol{y}_k, \ldots, \boldsymbol{x}_s) < \max(f(\boldsymbol{x}), f(\boldsymbol{x}_1, \ldots, \boldsymbol{y}_k, \ldots, \boldsymbol{x}_s)).$$

Grippo et Sciandrone [104] ont montré la converge globale vers un point stationnaire (non unique) de la méthode, sans aucune hypothèse de convexité nécessaire si la décomposition est en deux blocs, $s = 2$.

En revanche, lorsqu'il existe plus de deux blocs, $s > 2$, il est alors nécessaire que la fonction objectif f soit strictement quasiconvexe par rapport à $s - 2$ composantes ou que f soit pseudoconvexe et son ensemble niveau soit compact.

3.2.2 Applications

Lorsqu'il existe des divisions de variables naturelles, comme en classification non supervisée, il est très simple d'appliquer la méthode d'optimisation alternée. L'écriture du problème d'optimisation est également simple. Hu et Hathaway [105] ont montré l'efficacité de cette méthode comparée à des méthodes de type Newton (Powell's Method Discarding the Direction of Largest Decrease, Conjugate Gradient Method, Quasi-Newton Method). Même en utilisant des méthodes hybrides pour utiliser la convergence asymptotique rapide des méthodes de type Newton et compenser la sensibilité à l'initialisation, la méthode d'AO reste la plus rapide en terme de temps d'exécution.
Nous appliquons la méthode d'optimisation alternée aux modèles de classification non supervisée, FCM-GK [61] dans la section 3.2.2.1 et ECM [11] dans la section 3.2.2.2.

3.2.2.1 FCM

La décomposition
La décomposition des variables est évidente, le premier sous-ensemble est la matrice des degrés d'appartenance $\boldsymbol{U} = (u_{ij})$ de taille $(n \times c)$, le deuxième est l'ensemble des centroïdes de chaque classe $\boldsymbol{\mathcal{V}} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_c\}$, $\boldsymbol{v}_j \in \mathbb{R}^p$ et enfin l'ensemble des matrices définies induisant la norme de chaque classe $\boldsymbol{\mathcal{S}} = \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_c\}$. Rappelons la fonction à minimiser

$$\min_{(\boldsymbol{U} \in \mathcal{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}} \in \mathcal{S}_1)} J_{FCM-GK}(\boldsymbol{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m (\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j (\boldsymbol{x}_i - \boldsymbol{v}_j), \tag{3.13}$$

Avec les ensembles des contraintes,

$$\begin{aligned}\mathcal{U} &= \left\{\forall i,j \in [1,n]\times[1,c], u_{ij}\geq 0, \quad \sum_{j=1}^{c} u_{ij} = 1, \quad \sum_{i=1}^{n} u_{ij} > 0\right\},\\ \mathcal{S}_1 &= \left\{\forall j \in [1,c]\boldsymbol{S}_j \in \mathbb{R}^{n_d\times n_d}, \text{matrice symétrique définie positive}, \det(\boldsymbol{S}) = 1\right\}.\end{aligned}$$

L'optimisation
En commençant par $(\boldsymbol{U}^0, \boldsymbol{\mathcal{V}}^0, \boldsymbol{\mathcal{S}}^0)$, la méthode met à jour successivement les différentes variables $\boldsymbol{U}$, $\boldsymbol{\mathcal{V}}$ et $\boldsymbol{\mathcal{S}}$ afin de minimiser la fonction objectif :

$$\begin{aligned}\boldsymbol{U}^k &= \arg\min_{\boldsymbol{U}\in\mathcal{U}} J(\boldsymbol{U}, \boldsymbol{\mathcal{V}}^{k-1}, \boldsymbol{\mathcal{S}}^{k-1}),\\ \boldsymbol{\mathcal{V}}^k &= \arg\min_{\boldsymbol{\mathcal{V}}} J(\boldsymbol{U}^k, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}^{k-1}),\\ \boldsymbol{\mathcal{S}}^k &= \arg\min_{\boldsymbol{\mathcal{S}}\in\mathcal{S}_1} J(\boldsymbol{U}^k\boldsymbol{\mathcal{V}}^k, \boldsymbol{\mathcal{S}}).\end{aligned}$$

La minimisation successive des différentes variables $\boldsymbol{U}$, $\boldsymbol{\mathcal{V}}$ et $\boldsymbol{\mathcal{S}}$ est réalisée à l'aide des conditions d'optimalité du premier ordre.

Pour la mise à jour de $\boldsymbol{U}$, les variables $\boldsymbol{\mathcal{V}}$ et $\boldsymbol{\mathcal{S}}$ sont fixés. Le Lagrangien pour $\boldsymbol{U}$ s'écrit,

$$\mathcal{L}(\boldsymbol{U}, \boldsymbol{\Lambda}, \boldsymbol{M}) = J_{FCM-GK}(\boldsymbol{U}) + \sum_{i=1}^{n}\lambda_i(\sum_{j=1}^{c} u_{ij} - 1) + \sum_{j=1}^{n}\mu_j\sum_{i=1}^{n} u_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{c}\mu_{ij}u_{ij}.$$

La nouvelle partition $\boldsymbol{U}^{k+1}$ est la partition $\boldsymbol{U}^*$ qui annule le gradient du Lagrangien,

$$\frac{\partial\mathcal{L}(\boldsymbol{U}, \boldsymbol{\Lambda}, \boldsymbol{M})}{\partial u_{ij}} = m u_{ij}^{m-1}(\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j(\boldsymbol{x}_i - \boldsymbol{v}_j) - \lambda_i + \mu_j + \mu_{ij} = 0 \quad \forall i,j.$$

En supposant que les contraintes $\sum_{i=1}^{n} u_{ij} > 0$ et $u_{ij} \geq 0$ sont respectées, nous savons d'après les conditions de complémentarité que $\mu_j = 0$ et $\mu_{ij} = 0$ donc

$$\begin{aligned}& m u_{ij}^{m-1}(\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j(\boldsymbol{x}_i - \boldsymbol{v}_j) - \lambda_i = 0 \quad \forall i,j\\ \Leftrightarrow\ & u_{ij} = \left(\frac{\lambda_i}{m(\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j(\boldsymbol{x}_i - \boldsymbol{v}_j)}\right)^{\frac{1}{m-1}}.\end{aligned}$$

Or,

$$\begin{aligned}&\sum_{l=1}^{c} u_{il} = 1 \Leftrightarrow \sum_{l=1}^{c}\left(\frac{\lambda_i}{m(\boldsymbol{x}_i - \boldsymbol{v}_l)^\top \boldsymbol{S}_l(\boldsymbol{x}_i - \boldsymbol{v}_l)}\right)^{\frac{1}{m-1}} = 1\\ \Leftrightarrow\ &\left(\frac{\lambda_i}{m}\right)^{\frac{1}{m-1}} = \left(\sum_{l=1}^{c}\left(\frac{1}{(\boldsymbol{x}_i - \boldsymbol{v}_l)^\top \boldsymbol{S}_l(\boldsymbol{x}_i - \boldsymbol{v}_l)}\right)^{\frac{1}{m-1}}\right)^{-1}.\end{aligned}$$

D'où,

$$u_{ij} = \left(\sum_{l=1}^{c} \left[\frac{(\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j (\boldsymbol{x}_i - \boldsymbol{v}_j)}{(\boldsymbol{x}_i - \boldsymbol{v}_l)^\top \boldsymbol{S}_l (\boldsymbol{x}_i - \boldsymbol{v}_l)} \right]^{\frac{1}{m-1}} \right)^{-1} \quad \forall i, j. \tag{3.14}$$

Nous vérifions bien que les contraintes sont respectés, tous les degrés d'appartenance sont positifs ou nuls, $u_{ij} \geq 0$.
Ensuite $\boldsymbol{U}$ et $\boldsymbol{\mathcal{S}}$ sont fixés pour la mise à jour $\boldsymbol{\mathcal{V}}$, il n'y a pas de contraintes donc

$$\frac{\partial J_{FCM-GK}(\boldsymbol{\mathcal{V}})}{\partial \boldsymbol{v}_j} = 0 \Leftrightarrow 2 \sum_{i=1}^{n} u_{ij}^m \boldsymbol{S}_j (\boldsymbol{x}_i - \boldsymbol{v}_j) = 0,$$

$$\Rightarrow \boldsymbol{v}_j = \frac{\sum_{i=1}^{n} u_{ij}^m \boldsymbol{x}_i}{\sum_{i=1}^{n} u_{ij}^m} \quad \forall j. \tag{3.15}$$

Enfin pour la mise à jour de $\boldsymbol{\mathcal{S}}$, la partition $\boldsymbol{U}$ et les centroïdes $\boldsymbol{\mathcal{V}}$ sont fixés. Soit le Lagrangian,

$$\mathcal{L}(\boldsymbol{\mathcal{S}}, \boldsymbol{\Lambda}) = J_{FCM-GK}(\boldsymbol{\mathcal{S}}) + \sum_{j=1}^{c} \lambda_j (det(S_j) - 1),$$

avec les multiplicateurs λ_j spécifiques au Lagrangien de $\boldsymbol{\mathcal{S}}$.

La méthode calcule $S_j^{k+1} = S_j^*$ qui annule la dérivée du Lagrangien :

$$\frac{\partial \mathcal{L}(\boldsymbol{\mathcal{S}}, \boldsymbol{\Lambda})}{\partial \boldsymbol{S}_j} = \sum_{i=1}^{n} u_{ij}^m \frac{\partial (\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j (\boldsymbol{x}_i - \boldsymbol{v}_j)}{\partial \boldsymbol{S}_j} + \frac{\partial \lambda_j (det(\boldsymbol{S}_j) - 1)}{\partial \boldsymbol{S}_j} = 0.$$

Or, $\frac{\partial x^\top A x}{\partial A} = x x^\top$ et $\frac{\partial det(A)}{\partial A} = det(A) A^{-1}$, d'où

$$0 = \sum_{i=1}^{n} u_{ij}^m (\boldsymbol{x}_i - \boldsymbol{v}_j)(\boldsymbol{x}_i - \boldsymbol{v}_j)^\top - \lambda_j det(S_j) S_j^{-1}, \text{ en supposant } det(\boldsymbol{S}_j) = 1,$$

$$\Leftrightarrow \boldsymbol{S}_j = \frac{1}{\lambda_j} \Sigma_j^{-1}, \text{ avec } \Sigma_j = \sum_{i=1}^{n} u_{ij}^m (\boldsymbol{x}_i - \boldsymbol{v}_j)(\boldsymbol{x}_i - \boldsymbol{v}_j)^\top. \tag{3.16}$$

Pour retrouver $det(\boldsymbol{S}_j) = 1$, il faut prendre $\lambda_j = \frac{1}{det(\Sigma_j)^{\frac{1}{p}}}$. Par définition, Σ_j est la matrice de variance-covariance floue qui est semi-définie positive. Cependant, pour l'inverser, il est nécessaire de la transformer en une matrice définie positive. Pour ce faire, nous pouvons ajouter un terme de régularisation $\varepsilon \boldsymbol{I}$ où ε est une petite valeur, typiquement $\varepsilon = 10^{-10}$. Concrètement, la matrice sera réellement semi définie lorsque les données de la classe seront alignées sur une droite. Babuka et al. proposent deux autres méthodes alternatives pour traiter cette situation [106].

L'algorithme
Finalement, nous pouvons écrire l'algorithme 3 de l'optimisation alternée du modèle FCM-GK. L'algorithme est initialisé par une partition aléatoire $\boldsymbol{U}^0$. Le critère d'arrêt

est la stabilité de la partition, c'est-à-dire quand l'erreur absolue entre deux matrices $\boldsymbol{U}$ successives est plus petite qu'un seuil fixé à 10^{-3} [34]. L'initialisation et le critère d'arrêt peuvent être définis également par rapport aux cendroïdes $\boldsymbol{\mathcal{V}}$. Pour t itérations, la complexité temporelle de la méthode est $O(tnc^2n_d)$ [107] dans le cas euclidien. Pour $\boldsymbol{\mathcal{S}}^k$, il est nécessaire de former c matrices variance-covariance floue et de les inverser, donnant une complexité $O(c(nn_d^2 + n_d^3))$. Ainsi la complexité temporelle de FCM-GK est $O(t(nc^2n_d + ncn_d^2 + cn_d^3))$. La taille des variables est nn_d pour $\boldsymbol{X}$, nc pour $\boldsymbol{U}^k$, cn_d pour $\boldsymbol{\mathcal{V}}^k$ et cn_d^2 pour $\boldsymbol{\mathcal{S}}^k$. Donc la complexité spatiale de FCM-GK est $O(nn_d+nc+cn_d^2)$.

Algorithme 3 FCM-GK par AO.

Entrée : $\boldsymbol{X}$ les données, c le nombre de classes, m.
Sortie : $\boldsymbol{U}^k$,$\boldsymbol{\mathcal{V}}^k$,$\boldsymbol{\mathcal{S}}^k$

1: $err = 0, k = 0$,
2: $\boldsymbol{U}^0$ initialisation aléatoire.
3: **tant que** $err > 10^{-3}$ **faire**
4: $k = k + 1$
5: calcul $\boldsymbol{\mathcal{V}}^k$ (3.15) :

$$\boldsymbol{v}_j^k = \frac{\sum_{i=1}^n \left(u_{ij}^{k-1}\right)^m \boldsymbol{x}_i}{\sum_{i=1}^n \left(u_{ij}^{k-1}\right)^m}, \boldsymbol{q}_{ij}^k = \boldsymbol{x}_i - \boldsymbol{v}_j^k.$$

6: calcul $\boldsymbol{\mathcal{S}}^k$ (3.16) :

$$\boldsymbol{\Sigma}_j^k = \sum_{i=1}^n \left(u_{ij}^{k-1}\right)^m \boldsymbol{q}_{ij}^k (\boldsymbol{q}_{ij}^k)^\top, \boldsymbol{S}_j^k = \det(\boldsymbol{\Sigma}_j^k)^{\frac{1}{p}} (\boldsymbol{\Sigma}_j^k)^{-1}.$$

7: calcul $\boldsymbol{U}^k$ (3.14) :

$$u_{ij}^k = \left[\sum_{\ell=1}^c \frac{(\boldsymbol{q}_{ij}^k)^\top \boldsymbol{S}_j^k \boldsymbol{q}_{ij}^k}{(\boldsymbol{q}_{i\ell}^k)^\top \boldsymbol{S}_\ell^k \boldsymbol{q}_{i\ell}^k}\right]^{-1}.$$

8: $err = \| \boldsymbol{U}^k - \boldsymbol{U}^{k-1} \|$
9: **fin tant que**

La convergence

Höppner [108] a repris les travaux de Bezdek [109] sur la convergence de l'optimisation alternée pour FCM et FCM-GK vers un point fixe. Il est difficile d'établir des zones locales de convergence, par défaut le point fixe sera supposé être un point selle. Höppner définit une condition assurant le point fixe d'être un minimum local et non un point selle. Dans son étude, il ne regarde pas l'AO comme une méthode Gauss-Seidel par bloc. En effet, son analyse est rendue difficile par la résolution des sous-problèmes, selon chaque variable $\boldsymbol{U}, \boldsymbol{\mathcal{V}}$, c'est pourquoi il reformule FCM pour n'avoir plus qu'une

seule variable ($\boldsymbol{\mathcal{V}}$) :

$$J_{FCM'}(\boldsymbol{\mathcal{V}}) = \sum_{i=1}^{n} \left[\sum_{k=1}^{c} \parallel \boldsymbol{x}_i - \boldsymbol{v}_j \parallel_2^{\frac{2}{1-m}} \right]^{1-m}. \tag{3.17}$$

Gröll et Jäkel [110] vont reprendre cette reformulation et l'identifier comme une méthode de descente à plus forte pente et à pas variable. En utilisant les propriétés de cette méthode, ils prouvent la convergence globale vers un minimum local ou un point selle.

3.2.2.2 ECM

La décomposition
Pour la version d'ECM avec la distance de Mahalanobis [50], les matrices $\boldsymbol{\mathcal{S}}$ des sous-ensembles sont définies par la même formule barycentrique que les centroïdes. (2.29). De manière analogue, la décomposition des variables est évidente, la partition crédale $\boldsymbol{M}$, l'ensemble des centroïdes $\boldsymbol{\mathcal{V}}$ et l'ensemble des matrices $\boldsymbol{\mathcal{S}}$ induisant la norme de chaque sous-ensemble. Le problème est le suivant :

$$\min_{(\boldsymbol{M}\in\mathcal{M},\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{S}}\in\mathcal{S}_1)} J_{ECM}(\boldsymbol{M},\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{S}}) = \sum_{i=1}^{n}\sum_{\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^{\alpha} m_{ij}^{\beta} d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{3.18}$$

avec,

$$\begin{aligned} d_{ij}^2 &= (\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j)^{\top} \overline{\boldsymbol{S}}_j (\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j), \\ \overline{\boldsymbol{v}}_j &= \frac{1}{|\mathcal{A}_j|} \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{v}_\ell, \\ \overline{\boldsymbol{S}}_j &= \frac{1}{|\mathcal{A}_j|} \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{S}_\ell, \end{aligned}$$

où $s_{\ell j} = 1$ si $\omega_\ell \in \mathcal{A}_j$ sinon $s_{\ell j} = 0$.

Et les ensembles des contraintes,

$$\begin{aligned} \mathcal{M} &= \left\{ \forall i,j \in [1,n]\times[1,2^c], m_{ij} \geq 0, \quad \sum_{\mathcal{A}_j} m_{ij} = 1, \sum_{i=1}^{n} m_{ij} > 0 \right\}, \\ \mathcal{S}_1 &= \left\{ \forall \ell \in [1,c], \boldsymbol{S}_\ell \in \mathbb{R}^{n_d \times n_d}, \text{matrice symétrique définie positive}, \det(\boldsymbol{S}_\ell) = 1 \right\}. \end{aligned}$$

L'optimisation
L'application de la méthode d'optimisation alternée consiste à mettre à jour successivement les différentes variables soit

$$\begin{aligned}
\boldsymbol{M}^k &= \arg\min_{\boldsymbol{M}\in\mathcal{M}} J(\boldsymbol{M},\boldsymbol{\mathcal{V}}^{k-1},\boldsymbol{\mathcal{S}}^{k-1}),\\
\boldsymbol{\mathcal{V}}^k &= \arg\min_{\boldsymbol{\mathcal{V}}} J(\boldsymbol{M}^k,\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{S}}^{k-1}),\\
\boldsymbol{\mathcal{S}}^k &= \arg\min_{\boldsymbol{\mathcal{S}}\in\mathcal{S}_1} J(\boldsymbol{M}^k,\boldsymbol{\mathcal{V}}^k,\boldsymbol{\mathcal{S}}).
\end{aligned}$$

Comme vu précédemment pour FCM, il faut résoudre les conditions d'optimalité de chaque variable en fixant les autres.

Pour $\boldsymbol{M}$, le Lagrangien associé est

$$\begin{aligned}
\mathcal{L}(\boldsymbol{M},\boldsymbol{\Lambda},\boldsymbol{\Gamma}) = \quad & \sum_{i=1}^{n}\sum_{j,\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^{\alpha} m_{ij}^{\beta} d_{ij}^2 + \delta^2 m_{i\emptyset}^{\beta} + \sum_{i=1}^{n}\lambda_i\left(\sum_{\mathcal{A}_j} 1 - m_{ij}\right)\\
& + \sum_{j,\mathcal{A}_j\neq\emptyset}\gamma_j\sum_{i=1}^{n} m_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{2^c}\gamma_{ij}m_{ij}.
\end{aligned}$$

En supposant que $m_{ij}\geq 0, \sum_{i=1}^{n} m_{ij} > 0$, d'après les conditions de complémentarités $\gamma_{ij}=0, \gamma_j = 0$ donc la condition d'optimalité donne

$$\begin{cases}
\frac{\partial\mathcal{L}(\boldsymbol{M},\boldsymbol{\Lambda})}{\partial m_{ij}} = 0 \Longleftrightarrow m_{ij} = \left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}}\left(\frac{1}{|\mathcal{A}_j|^{\alpha} d_{ij}^2}\right)^{\frac{1}{\beta-1}} \quad \forall i, \mathcal{A}_j\neq\emptyset,\\
\frac{\partial\mathcal{L}(\boldsymbol{M},\boldsymbol{\Lambda})}{\partial m_{i\emptyset}} = 0 \Longleftrightarrow m_{i\emptyset} = \left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}}\left(\frac{1}{\delta^2}\right)^{\frac{1}{\beta-1}} \quad \forall i,\\
\frac{\partial\mathcal{L}(\boldsymbol{M},\boldsymbol{\Lambda})}{\partial \lambda_i} = 0 \Longleftrightarrow \sum_{\mathcal{A}_\ell} m_{i\ell} = 1.
\end{cases}$$

En utilisant le même raisonnement que dans FCM pour isoler le multiplicateur de Lagrange, on obtient $\left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}} = \left(\sum_{\mathcal{A}_\ell\neq\emptyset}|\mathcal{A}_\ell|^{\frac{-\alpha}{\beta-1}} d_{i\ell}^{\frac{-2}{\beta-1}} + \delta^{-\frac{2}{\beta-1}}\right)^{-1}$. D'où la formulation de la mise à jour :

$$\forall i, \mathcal{A}_j\neq\emptyset, m_{ij} = \frac{|\mathcal{A}_j|^{\frac{-\alpha}{\beta-1}} d_{ij}^{\frac{-2}{\beta-1}}}{\sum_{\mathcal{A}_\ell\neq\emptyset}|\mathcal{A}_\ell|^{\frac{-\alpha}{\beta-1}} d_{i\ell}^{\frac{-2}{\beta-1}} + \delta^{-\frac{2}{\beta-1}}}, \quad m_{i\emptyset} = 1 - \sum_j m_{ij}. \tag{3.19}$$

Nous remarquons que l'hypothèse sur les contraintes sont vérifiées :

$$\forall i, j[1,n]\times[1,2^c], m_{ij}\geq 0, \sum_{i=1}^{n} m_{ij} > 0$$

.

Pour la minimisation de $\boldsymbol{\mathcal{V}}$, la distance utilisée à l'itération est formulée

$$d_{ij}^2 = (\boldsymbol{x}_i - \frac{1}{|\mathcal{A}_j|}\sum_{\ell=1}^{c} s_{\ell j}\boldsymbol{v}_\ell)^\top \overline{\boldsymbol{S}}_j(\boldsymbol{x}_i - \frac{1}{|\mathcal{A}_j|}\sum_{\ell=1}^{c} s_{\ell j}\boldsymbol{v}_\ell),$$

$\boldsymbol{M}, \boldsymbol{\mathcal{S}}$ sont fixés. On ne s'intéresse qu'aux centroïdes des classes et non aux sous-ensembles. Le Lagrangian associé à $\boldsymbol{\mathcal{V}}$ est la fonction J_{ECM} en $\boldsymbol{\mathcal{V}}$

$$\mathcal{L}(\boldsymbol{\mathcal{V}}) = \sum_{i=1}^{n}\sum_{\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^{n}\delta^2 m_{i\emptyset}^\beta.$$

La condition d'optimalité du premier ordre revient à annuler le gradient de $\mathcal{L}(\boldsymbol{\mathcal{V}})$:

$$\frac{\partial\mathcal{L}(\boldsymbol{\mathcal{V}})}{\partial\boldsymbol{v}_\ell} = \sum_{i=1}^{n}\sum_{\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^\alpha m_{ij}^\beta \frac{-2s_{\ell j}}{|\mathcal{A}_j|}\overline{\boldsymbol{S}}_j(\boldsymbol{x}_i - \frac{1}{|\mathcal{A}_j|}\sum_{l=1}^{c} s_{lj}\boldsymbol{v}_l) = 0, \quad \forall\ell\in[1,c].$$

$$\iff \sum_{l=1}^{c}\sum_{i=1}^{n}\sum_{\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^{\alpha-2} m_{ij}^\beta s_{\ell j}s_{lj}\overline{\boldsymbol{S}}_j\boldsymbol{v}_l = \sum_{i=1}^{n}\sum_{\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^{\alpha-1} m_{ij}^\beta s_{\ell j}\overline{\boldsymbol{S}}_j\boldsymbol{x}_i.$$

A partir de ces équations, nous pouvons écrire un système linéaire. Pour cela, il est nécessaire de vectoriser la matrice des objects $\boldsymbol{X}$ en un vecteur $\boldsymbol{x}$ de taille $(nn_d \times 1)$ et de vectoriser $\boldsymbol{\mathcal{V}}$ en un vecteur $\boldsymbol{v}$ de taille $(cn_d \times 1)$. Le système linéaire à résoudre est

$$\boldsymbol{G}\boldsymbol{v} = \boldsymbol{F}\boldsymbol{x}, \tag{3.20}$$

avec les deux matrices $\boldsymbol{F}, \boldsymbol{G}$ de dimensions $(cn_d \times nn_d)$, $(cn_d \times cn_d)$:

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{F}^{1,1} & \dots & \boldsymbol{F}^{1,n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{F}^{c,1} & \dots & \boldsymbol{F}^{c,n} \end{pmatrix}, \quad \boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}^{1,1} & \dots & \boldsymbol{G}^{1,c} \\ \vdots & \ddots & \vdots \\ \boldsymbol{G}^{c,1} & \dots & \boldsymbol{G}^{c,c} \end{pmatrix},$$

formées par,

$$\boldsymbol{F}^{\ell,i} = \sum_{\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^{\alpha-1} m_{ij}^\beta s_{\ell j}\overline{\boldsymbol{S}}_j, \quad \boldsymbol{G}^{\ell,l} = \sum_{i=1}^{n}\sum_{\mathcal{A}_j\neq\emptyset} |\mathcal{A}_j|^{\alpha-2} m_{ij}^\beta s_{\ell j}s_{lj}\overline{\boldsymbol{S}}_j.$$

Enfin, les centroïdes des sous-ensembles peuvent-être mis à jour avec la formule barycentrique,

$$\overline{\boldsymbol{v}}_j = \frac{1}{|\mathcal{A}_j|}\sum_{\ell=1}^{c} s_{\ell j}\boldsymbol{v}_\ell. \tag{3.21}$$

Nous supposons désormais $\boldsymbol{M}, \boldsymbol{\mathcal{V}}$ fixés, et utilisons la distance

$$d_{ij}^2 = (\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j)^\top \frac{1}{|\mathcal{A}_j|} \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{S}_\ell (\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j),$$

pour la mise à jour des $\boldsymbol{\mathcal{S}}$ de chaque classe. Le Lagrangien associé est

$$\mathcal{L}(\boldsymbol{\mathcal{S}}) = \sum_{i=1}^{n} \sum_{\mathcal{A}_j \neq \emptyset} |\mathcal{A}_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^\beta + \sum_{\ell=1}^{c} \lambda_\ell (1 - det(\boldsymbol{S}_\ell)).$$

L'optimisation est similaire à FCM avec $\forall \ell \in [1, c]$

$$\boldsymbol{S}_\ell = det(\boldsymbol{\Sigma}_\ell)^{\frac{1}{p}} \boldsymbol{\Sigma}_\ell^{-1}, \tag{3.22}$$

avec la matrice de variance-covariance

$$\boldsymbol{\Sigma}_\ell = \sum_{i=1}^{n} \sum_{\mathcal{A}_j \neq \emptyset} s_{\ell j} |\mathcal{A}_j|^{\alpha-1} m_{ij}^\beta (\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j)(\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j)^\top.$$

En utilisant la formule barycentrique, nous pouvons calculer la matrice de Mahalanobis de chaque sous-ensemble non vide $\mathcal{A}_j$,

$$\overline{\boldsymbol{S}}_j = \frac{1}{|\mathcal{A}_j|} \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{S}_\ell. \tag{3.23}$$

Algorithme

L'algorithme 4 présente l'optimisation alternée appliquée à ECM. La condition d'arrêt est similaire à celle de l'algorithme 3 FCM. On vérifie la stabilisation de la partition, avec l'erreur absolue entre deux matrices $\boldsymbol{M}$ successives. L'initialisation peut être aléatoire avec la génération aléatoire des centroïdes $\boldsymbol{\mathcal{V}}^0$ ou de la partition crédale $\boldsymbol{M}^0$. Elle peut aussi être intelligente, par exemple en appelant la méthode FCM, algorithme 3. En ce qui concerne la complexité, la mise à jour des matrices $\boldsymbol{\mathcal{V}}_\ell$ nécessite la formation d'un système linéaire suivi de sa résolution, ce qui a une complexité de l'ordre de $O(cn2^{c-1} + (cn_d)^3)$. La formulation des matrices de variance-covariance floues et leur inversion coûtent $O(c2^{n-1}nn_d^2 + cn_d^3)$. Enfin, la mise à jour de la partition $\boldsymbol{M}$ nécessite $O(n2^{2c-2}n_d^2)$ opérations. Ainsi, après t itérations, la complexité temporelle totale de la méthode est $O(t(n2^{2c-2}n_d^2 + (cn_d)^3))$. En ce qui concerne la complexité spatiale, les variables ont des tailles respectives de nn_d pour $\boldsymbol{X}$, $n2^{c-1}$ pour $\boldsymbol{U}$, $2^{c-1}n_d$ pour $\boldsymbol{\mathcal{V}}$ et $2^{c-1}n_d^2$ pour $\boldsymbol{\mathcal{S}}$. De plus, les matrices $\boldsymbol{F}$ et $\boldsymbol{G}$ ont des dimensions de cn_dnn_d et $(cn_d)^2$. Par conséquent, la complexité spatiale totale d'ECM est en $O(nn_d + n2^{c-1} + 2^{c-1}n_d^2 + ncn_d^2)$.

Algorithme 4 ECM par AO.

Entrée : $\boldsymbol{X}$ les données, c le nombre de classes, α,β,δ.
Sortie : $\boldsymbol{M}^k$,$\boldsymbol{\mathcal{V}}^k$,$\boldsymbol{\mathcal{S}}^k$

1: $err = 1, k = 0$,
2: $\boldsymbol{M}^0$ initialisation aléatoire ou FCM.
3: **tant que** $err > 10^{-3}$ **faire**
4: $k = k + 1$
5: Calcul de $\boldsymbol{\mathcal{V}}_\ell^k$ (3.20) *(résolution du système linéaire)* :

$$\boldsymbol{G}^{k-1}\boldsymbol{v}^k = \boldsymbol{F}^{k-1}\boldsymbol{x}.$$

6: Calcul de $\boldsymbol{\mathcal{V}}_j^k$ (3.21) *(formule barycentrique)* :

$$\overline{\boldsymbol{v}}_j^k = \frac{1}{|\mathcal{A}_j|}\sum_{\ell=1}^{c} s_{\ell j}\boldsymbol{v}_\ell^k.$$

7: Calcul de $\boldsymbol{\mathcal{S}}_\ell^k$ (3.22) :

$$\boldsymbol{\Sigma}_\ell^k = \sum_{i=1}^{n}\sum_{\mathcal{A}_j\neq\emptyset} s_{\ell j}|\mathcal{A}_j|^{\alpha-1}\left(m_{ij}^{k-1}\right)^{\beta}(\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j^k)(\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j^k)^\top,$$

$$\boldsymbol{S}_\ell^k = det(\boldsymbol{\Sigma}_\ell^k)^{\frac{1}{p}}\left(\boldsymbol{\Sigma}_\ell^k\right)^{-1}.$$

8: Calcul de $\boldsymbol{\mathcal{S}}_j^k$ (3.23) *(formule barycentrique)* :

$$\overline{\boldsymbol{S}}_j^k = \frac{1}{|\mathcal{A}_j|}\sum_{\ell=1}^{c} s_{\ell j}\boldsymbol{S}_\ell^k.$$

9: Calcul de $\boldsymbol{M}^k$ (3.19) :

$$\forall i, \mathcal{A}_j \neq \emptyset, m_{ij}^k = \frac{|\mathcal{A}_j|^{\frac{-\alpha}{\beta-1}}((\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j^k)^\top\overline{\boldsymbol{S}}_j^k(\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j^k))^{\frac{-2}{\beta-1}}}{\sum_{\mathcal{A}_l\neq\emptyset}|\mathcal{A}_l|^{\frac{-\alpha}{\beta-1}}((\boldsymbol{x}_i - \overline{\boldsymbol{v}}_l^k)^\top\overline{\boldsymbol{S}}_l^k(\boldsymbol{x}_i - \overline{\boldsymbol{v}}_l^k))^{\frac{-2}{\beta-1}} + \delta^{-\frac{2}{\beta-1}}}, \quad m_{i\emptyset}^k = 1 - \sum_j m_{ij}^k.$$

10: $err = \| \boldsymbol{M}^k - \boldsymbol{M}^{k-1} \|$
11: **fin tant que**

3.3 Méthode du gradient proximal accéléré

3.3.1 Méthode

Dans le cas d'une fonction non différentiable, les techniques conventionnelles d'optimisation de descentes de gradient pour résoudre le problème ne sont plus applicables. Soit f une fonction convexe décomposable de $\mathbb{R}^n$ dans $\mathbb{R}$,

$$min_{\boldsymbol{x}} f(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}), \tag{3.24}$$

avec f_1 différentiable mais pas f_2. Il est impossible de dériver la fonction f_2. Pour résoudre ce problème, les méthodes adaptées, comme la méthode du gradient proximal accéléré, utilisent l'*opérateur proximal* pour exploiter la convexité de la fonction à minimiser. En effet, cet opérateur permet de prendre en compte la structure convexe de la fonction tout en gérant les termes non différentiables de manière efficace.

Définition 3.3.11: Opérateur proximal

Soit f une fonction convexe de $\mathbb{R}^n$ dans $\mathbb{R}$, l'opérateur proximal $prox_f$ de f, $\mathbb{R}^n \longrightarrow \mathbb{R}^n$,

$$prox_f(\boldsymbol{x}) = \underset{\boldsymbol{y}}{argmin} \left(f(\boldsymbol{y}) + \frac{1}{2} \parallel \boldsymbol{x} - \boldsymbol{y} \parallel_2^2 \right), \tag{3.25}$$

L'opérateur $prox_{f,r}$ de f avec la pénalité r, $\mathbb{R}^n \longrightarrow \mathbb{R}^n$

$$prox_{f,r}(\boldsymbol{x}) = \underset{\boldsymbol{y}}{argmin} \left(f(\boldsymbol{y}) + \frac{r}{2} \parallel \boldsymbol{x} - \boldsymbol{y} \parallel_2^2 \right), \tag{3.26}$$

où $\parallel . \parallel_2^2$ est la norme euclidienne.

Les opérateurs proximaux sont une généralisation des projections dans le cas particulier où $f = \mathbb{I}_C$ est une fonction indicatrice d'un ensemble convexe C, $prox_f = \prod_C$. Ce qui leur confère de nombreuses propriétés [111]. L'opérateur proximal de f peut également être interprété comme une sorte de pas de gradient pour la fonction f.
Ces méthodes sont itératives, partant d'un $\boldsymbol{x}^0$, l'itération $k+1$

$$\boldsymbol{x}^{k+1} = prox_{f_2,t_k} \left(\boldsymbol{x}^k - t_k \nabla f_1(\boldsymbol{x}^k) \right). \tag{3.27}$$

Ces méthodes sont appelées méthodes de gradient proximal puisqu'on retrouve la formulation d'une méthode de descente de gradient :

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - t_k G_{tk}(\boldsymbol{x}^k),$$

avec le gradient généralisé de f,

$$G_t(\boldsymbol{x}) = \frac{x - prox_{f_2,t}\left(\boldsymbol{x} - t\nabla f_1(\boldsymbol{x})\right)}{t} = \frac{1}{t}\left(\boldsymbol{x} - t\nabla f_1(\boldsymbol{x})\right).$$

De nombreuses recherches ont été menées pour améliorer l'efficacité des méthodes de gradient. Une approche notable consiste à introduire un concept d'inertie, d'historique des états précédents dans la mise à jour des variables. Cette idée s'inspire de la physique, où un objet en mouvement conserve son élan et son accélération dans sa direction. Cette technique vise à accroître la probabilité de convergence vers un minimum global plutôt que de rester bloqué dans un minimum local ou un point stationnaire. Les méthodes qui intègrent cette notion d'historique dans leur processus de mise à jour sont couramment appelées méthodes de type *momentum*. La quantité d'informations ajoutés est défini par un hyperparamètre dont sa valeur est entre 0 (descente de gradient) et 1. Cette démarche a été introduite par Nesterov [9,112] dans le cas des fonctions convexes continues à gradient lipschitzien, qui respectent l'inégalité (3.28).
Comme la méthode du gradient proximal ressemble à la méthode du gradient, Beck et Teboulle [10] ont étendu la méthode de Nesterov au gradient proximal dont voici l'algorithme 5.

Algorithme 5 APG *gradient proximal accéléré.*

Itération $k = 0 : \boldsymbol{x}^0$ aléatoire , $\boldsymbol{y}^0 = \boldsymbol{x}^0, t_0 = 1$ et $\delta > 0$
Itération $k \geq 1$:

1: $\boldsymbol{x}^k = prox_{f_2,\delta}(\boldsymbol{y}^{k-1} - \delta \nabla f_1(\boldsymbol{y}^{k-1}))$
2: $t_k = \frac{1}{2}\left(1 + \sqrt{1 + 4t_{k-1}^2}\right)$
3: $\boldsymbol{y}^k = \boldsymbol{x}^k + (t_{k-1} - 1)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})/t_k$

Le paramètre δ est défini comme $\delta = 1/L$, où L est la constante de Lipschitz de ∇f_1, i.e,

$$\forall \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n, \quad \| \nabla f_1(\boldsymbol{x}) - \nabla f_1(\boldsymbol{z}) \| \leq L \| \boldsymbol{x} - \boldsymbol{z} \| . \tag{3.28}$$

3.3.2 Convergence

Les méthodes du premier ordre, utilisant la dérivée première, ont besoin de l'hypothèse de la stricte convexité de la fonction objectif. Pour la méthode de Nesterov, le taux de convergence de l'algorithme est de $O(1/k^2)$, plus rapide qu'une descente de gradient standard $O(1/k)$ [9] (voir le théorème 3.3.4). D'autres travaux, comme celui de Necaora et al. tentent d'assouplir l'hypothèse de stricte convexité [113].

Théorème 3.3.4: Taux de convergence du Gradient Proximal Accéléré

Soit f une fonction continue, convexe et ∇f_1 est L-Lipschitzienne, pour n'importe quel minimiseur x^*, la séquence $\boldsymbol{x}^k$ générée par l'algorithme 5, avec $\delta = 1/L$ est telle que

$$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^*) \leq \frac{2L}{k^2} \parallel \boldsymbol{x}^0 - \boldsymbol{x}^* \parallel^2,$$

pour tout $k \geq 0$.

Bien que plus rapide que la descente de gradient, cette méthode perd la propriété de monotonies de la valeur de la fonction objectif car elle peut augmenter. Dans ce cas, on parle de "rebond". Pour éviter cela, il est nécessaire de faire des redémarrages, repartir du point précédent l'algorithme ($t_k = 1 \Longrightarrow \boldsymbol{y}^k = \boldsymbol{x}^{k-1}$). O'Donoghue et Candès [114] proposent deux schémas de redémarrage adaptatif qui suggère une amélioration de la convergence par rapport à des redémarrages à pas fixe (tous les k itérations).

Schéma de la fonction

$$f(\boldsymbol{x}^{k+1}) > f(\boldsymbol{x}^k).$$

A chaque itération $k+1$ où la fonction n'est pas minimisée (début du rebond), l'algorithme redémarre avec le point précédent $\boldsymbol{x}^k$.

Schéma du gradient

$$\nabla f_1(\boldsymbol{y}^k)^\top (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) > 0.$$

A chaque itération $k+1$ où l'inertie semble entraîner la minimisation dans une mauvaise direction (gradient négatif), l'algorithme redémarre avec le point précédent $\boldsymbol{x}^k$. Le deuxième schéma est plus intéressant pour la complexité spatiale et temporelle de l'algorithme puisque chaque quantité de la formule est disponible.

3.3.3 Applications

La descente de gradient accéléré est utilisée dans divers domaines comme en traitement des signaux et en apprentissage automatique [115], pour la résolution d'équations aux dérivées partielles [116]. Plus récemment pour la restauration d'images, son efficacité et sa précision ont été démontrées [117]. Elle est aussi utilisée en apprentissage profond pour optimiser les poids des réseaux neuronaux [118].

3.4 Méthode des directions alternées et multiplicateurs

3.4.1 Méthode d'Uzawa

Nous souhaitons résoudre le problème avec les contraintes d'égalité (3.1)-(3.2). Pour rappel :

$$\underset{\boldsymbol{x}}{min} f(\boldsymbol{x}),$$

sous les contraintes,

$$\begin{aligned} &\boldsymbol{h}(\boldsymbol{x}) = 0, \\ &\boldsymbol{x} \in \mathbb{R}^n, \end{aligned}$$

où la fonction objectif $f : \mathbb{R}^n \to \mathbb{R}$ est supposée continue et différentiable, et les contraintes d'égalité modélisées par la fonction $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^m$.
Nous souhaitons résoudre ce problème par son problème dual de Lagrange associé.
Soit J la fonction duale,

$$J(\boldsymbol{\lambda}) = \inf_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{h}(\boldsymbol{x}).$$

Le problème dual est alors,

$$\max_{\boldsymbol{\lambda} \geq 0} J(\boldsymbol{\lambda}).$$

Pour se faire, nous pouvons utiliser la méthode d'Uzawa. Elle est l'équivalent dual de la méthode de descente de gradient et est ainsi appelée méthode de montée duale, *dual ascent method*, puisqu'elle maximise le problème dual. C'est un algorithme itératif de type Uzawa de la forme suivante

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho_k \boldsymbol{\nu}^k,$$

où $\rho_k > 0$ est le pas et $\boldsymbol{\nu}_k$ est la direction de recherche telle que,

$$\langle \nabla J(\boldsymbol{\lambda}^k), \boldsymbol{\nu}_k \rangle > 0.$$

Une simple direction de montée est $\boldsymbol{\nu}_k = \nabla J(\boldsymbol{\lambda}^k)$. Pour une direction donnée, le pas recherché maximise la fonction $\phi(\rho) = J(\boldsymbol{\lambda}^k + \rho \boldsymbol{\nu}^k)$.

3.4.2 Méthode du Lagrangien augmenté

La méthode du Lagrangien augmenté, développée par Hestenes [119] et Powell [120] transforme un problème d'optimisation sous contraintes en un problème d'optimisation sans contraintes [93, 121]. Cette méthode s'inspire des méthodes de pénalité qui supprime les contraintes du problème par l'ajout d'une fonction pénalité dans la fonction objectif. La nouvelle fonction est appelée le Lagrangien augmenté.

Définition 3.4.12: Lagrangien augmenté

Soit le problème d'optimisation (3.1)-(3.2), soit le vecteur multiplicateur de Lagrange $\boldsymbol{\lambda} \in \mathbb{R}^m$, $\boldsymbol{\lambda} \geq 0$, le Lagrangien augmenté $\mathcal{L}_r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ est la fonction

$$\mathcal{L}_r(\boldsymbol{x}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) + \frac{r}{2} \parallel \boldsymbol{h}(\boldsymbol{x}) \parallel_2^2 = f(\boldsymbol{x}) + \boldsymbol{\lambda}_i^\top \boldsymbol{h}_i(\boldsymbol{x}) + \frac{r}{2} \parallel \boldsymbol{h}(\boldsymbol{x}) \parallel_2^2, \qquad (3.29)$$

où $\parallel . \parallel_2$ est la norme associée à $\mathbb{R}^n$et où $r > 0$ est appelée paramètre de pénalité.

La méthode du Lagrangien augmenté est une méthode itérative qui utilise le formalisme de la méthode d'Uzawa où le pas choisi est le terme de pénalité r et la direction de descente est la valeur de la contrainte d'égalité $\boldsymbol{h}(\boldsymbol{x})$, voir l'algorithme 6 d'Uzawa pour le Lagrangien augmenté. Elle est initialisée avec des valeurs choisies pour les multiplicateurs et le terme de pénalité qui peut être adapté selon le résidu primal ou le résidu dual [122].

Algorithme 6 Lagrangien augmenté.

Itération $k = 0$: $\boldsymbol{\lambda}^0$ donnée et $r^0 > 0$
Itération $k \geq 1$:
1: $\boldsymbol{x}^k = \underset{\boldsymbol{x}}{\text{argmin}} \mathcal{L}_r(\boldsymbol{x}, \boldsymbol{\lambda}^{k-1})$
2: $\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} + r^{k-1} \boldsymbol{h}(\boldsymbol{x}^k)$
3: Mise à jour de r^k

Le terme de pénalité r est l'hyperparamètre de cette méthode, c'est aussi son avantage principal car il n'est pas nécessaire de l'augmenter jusqu'à l'infini pour obtenir une convergence contrairement à la méthode de pénalisation intérieure. La précision de l'estimation du multiplicateur de Lagrange s'améliore à chaque étape. Les bonnes performances de cette méthode ont été illustrées par les travaux de Miele et al. [123, 124].

3.4.3 La méthode des directions alternées

La méthode des directions alternées, ou ADMM pour *Alternating Direction Method of Multipliers*, a été développée dans les années soixante-dix par Glowinski et Marrocco [7] et par Gabay et Mercier [8]. L'idée de la méthode ADMM est d'utiliser un processus de décomposition-coordination en vue de résoudre plus facilement les problèmes complexes. La décomposition permet de résoudre des sous-problèmes plus simples plutôt qu'un seul gros problème (processus similaire à Gauss-Seidel). Il existe plusieurs méthodes utilisant des schémas de fractionnement "*splitting scheme*" comme Douglas-Rachfor splitting [125]. La coordination est réalisée par les multiplicateurs de Lagrange. Cette méthode est une version du Lagrangien augmenté par bloc, aussi

appelée Uzawa bloc relaxation [126].
Le problème qui nous intéresse dans cette partie est défini comme,

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{z}\in\mathcal{X}\subseteq\mathbb{R}^n}{min} f(\boldsymbol{x},\boldsymbol{z}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{z}), \\ &\text{sous la contrainte } \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{z} = 0. \end{aligned}$$

où $f_1, f_2 : \mathcal{X} \longrightarrow \mathbb{R}$ sont deux fonctions convexes et f_1 différentiable et f_2 différentiable ou non.
Le Lagrangien augmenté associé à ce problème est

$$\mathcal{L}_r(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\lambda}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{z}) + \boldsymbol{\lambda}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{z}) + \frac{r}{2} \parallel \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{z} \parallel_2^2 . \tag{3.30}$$

L'algorithme 7 de la méthode des directions alternées reprend exactement les même étapes que celui du Lagrangien augmenté en décomposant la résolution du problème variable par variable : "directions alternées" (procédure type Gauss-Seidel). Chaque sous-problème considère la minimisation d'une seule variable, une seule direction, les autres variables sont fixées. Quand toutes les directions ont été explorées, la mise à jour des multiplicateurs de Lagrange permettent de coordonner les variables.

Algorithme 7 ADMM *directions alternées.*

Itération $k = 0 : \boldsymbol{\lambda}^0, \boldsymbol{z}^0$ donnée et $r > 0$
Itération $k \geq 0$:
1: $\boldsymbol{x}^k = \underset{\boldsymbol{x}}{\text{argmin}}\ \mathcal{L}_r(\boldsymbol{x}, \boldsymbol{z}^{k-1}, \boldsymbol{\lambda}^{k-1})$
2: $\boldsymbol{z}^k = \underset{\boldsymbol{z}}{\text{argmin}}\ \mathcal{L}_r(\boldsymbol{x}^k, \boldsymbol{z}, \boldsymbol{\lambda}^{k-1})$
3: $\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} + r(\boldsymbol{x}^k - \boldsymbol{z}^k)$
4: Mise à jour de r

ADMM exploite son plein potentiel pour la minimisation de fonction décomposable. Dans ce cas, l'ajout de variables auxiliaires pour séparer les variables originales permet la reformulation du problème et sa résolution en sous-problème plus simple. Par exemple, prenons un problème d'optimisation convexe séparable sans contrainte,

$$\underset{\boldsymbol{x}\in\mathcal{X}\subseteq\mathbb{R}^n}{min} f(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}).$$

Il est possible de reformuler le problème en ajoutant une variable auxiliaire ($\boldsymbol{z} \in \mathbb{R}^n$) pour séparer la fonction objectif en deux parties indépendantes,

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{z}\in\mathcal{X}\subseteq\mathbb{R}^n}{min} f(\boldsymbol{x},\boldsymbol{z}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{z}), \\ &\text{sous la contrainte } \boldsymbol{x} - \boldsymbol{z} = 0. \end{aligned}$$

En pratique, le choix des variables auxiliaires revêt une grande importance pour la performance de cette méthode. Pour tirer pleinement partie de la décomposition, il est judicieux de sélectionner des variables qui simplifient au maximum la résolution des sous-problèmes. En revanche, si les variables auxiliaires sont mal introduites, le problème peut devenir non convexe.
Enfin, si la décomposition conduit à des sous-problèmes indépendants et de complexité temporelle équivalente, il devient intéressant d'envisager la parallélisation de leur résolution.

3.4.4 Convergence

Dans le cadre des fonctions convexes, Glowinski [127] énonce les hypothèses nécessaires à la convergence de l'algorithme (voir [128, théorème 4.1]). Pour les fonctions multi-convexes, une formulation spécifique de l'ADMM ainsi que certaines hypothèses garantissent la convergence vers un point de Nash avec un taux de convergence en $o(1/k)$ équivalent au cas convexe [129]. Le point de Nash est un point d'équilibre introduit par John Forbes Nash en théorie des jeux.

Définition 3.4.13: Point de Nash

Pour la fonction f, $\tilde{\boldsymbol{x}} = \{\tilde{x}_1, \dots, \tilde{x}_n\}$ est un point de Nash s'il vérifie :

$$\forall i \in [1, n], \quad \forall x_i \in \mathcal{X}_i(\tilde{x}_1, \dots, \tilde{x}_{i-1}, \tilde{x}_{i+1}, \tilde{x}_n),$$
$$f(\tilde{\boldsymbol{x}}) \leq f(\tilde{x}_1, \dots, \tilde{x}_{i-1}, x_i, \tilde{x}_{i+1}, \tilde{x}_n).$$

En optimisation, le point de Nash peut être interprété comme une condition d'optimalité locale. Lorsque l'on optimise une variable tout en fixant les autres, le point de Nash assure l'optimalité de cette variable.

Pour le cas non convexe, la convergence de l'algorithme a été démontrée pour des cas particuliers [130–132], notamment en respectant l'inégalité de Kurdyka–Lojasiewicz (KL). Souvent considérée comme un cas particulier de la méthode du Lagrangien augmenté, elle est également étudiée comme équivalente à la méthode de Douglas-Rachford Splitting [133]. Plus simplement, Koko [134] propose une boucle interne sur la mise à jour des variables primales (5 par défaut) dans le but d'assurer la diminution du Lagrangien augmenté : soit à l'itération k, $\quad \mathcal{L}_r(\boldsymbol{x}^k, \boldsymbol{z}^k, \boldsymbol{\lambda}^{k-1}) < \mathcal{L}_r(\boldsymbol{x}^{k-1}, \boldsymbol{z}^{k-1}, \boldsymbol{\lambda}^{k-1})$.

Comme pour tout hyperparamètre, la valeur du terme de pénalité r est généralement cruciale à l'exception du cas convexe avec deux blocs, où l'algorithme converge pour n'importe quelle pénalité strictement positive, $r > 0$ [135]. Même dans un cas convexe avec trois blocs, le choix de sa valeur est primordiale [136]. Lorsque la pénalité est trop faible, l'algorithme risque de diverger mais il n'est pas nécessaire de l'augmenter jusqu'à l'infini pour obtenir convergence. C'est l'avantage du Lagrangien

augmenté. En outre, il est possible d'adapter à chaque itération la pénalité selon le résidu primal et dual [122]. Wolheber propose une étude plus approfondie des pénalités adaptatives [137].

3.4.5 Applications

Ces méthodes sont approfondies dans [128] et [122]. Ces ouvrages présentent et citent de nombreuses applications pratiques dans divers domaines.

En mécanique [128, 135, 138–140], ADMM est appliqué avec succès. Dans le contexte de la mécanique, ADMM offre des avantages significatifs en terme de précision et d'efficacité pour résoudre ces problèmes complexes.

Dans le domaine du machine learning, ADMM a également été appliqué [126, 134]. Il est largement utilisé pour des tâches telles que la restauration d'images [122, 141], où il permet de restaurer des images dégradées en réduisant le bruit et en améliorant la qualité visuelle. La méthode est également employée pour la régression avec des normes non différentiables [122, 142]

3.5 Comparaison des méthodes

3.5.1 Théorie

Cette section présente un comparatif des méthodes d'optimisation présentées dans la section précédente.

La modularité de résolution, offerte par l'optimisation alternée (AO), permet une simplification de la résolution et un potentiel parallélisme des calculs. Son application en classification non supervisée est particulièrement pertinente. En effet, son problème associé présente une décomposition naturelle des variables.

Cependant, le manque de coordination entre les variables dans cette méthode peut poser des difficultés. La convergence globale n'est pas garantie pour les fonctions non convexes, ce qui constitue une limitation importante de l'utilisation de l'optimisation alternée. C'est là que la méthode ADMM se distingue. En effet, ADMM simplifie le problème en le décomposant en sous-problèmes à l'aide de variables auxiliaires et coordonne ces variables à l'aide de multiplicateurs de Lagrange, ce qui garantit de bonnes propriétés de convergence globale. De plus, cette méthode s'applique très bien aux problèmes séparables. Elle exploite non seulement la décomposition naturelle des variables, mais aussi la décomposition de la fonction objectif, vue comme la somme de fonctions indépendantes. Toutefois, il est important de noter que l'ADMM est sensible aux hyperparamètres.

Le plein potentiel d'ADMM et d'AO dépend de la nature intrinsèque du problème. Ces méthodes sont sensibles à la manière dont les variables sont décomposées, ce qui peut avoir un impact significatif sur la qualité de la convergence et la complexité computationnelle des sous-problèmes.

Le gradient proximal accéléré (APG) n'est pas une méthode de résolution de type Gauss-Seidel. Son application se limite à la résolution de problèmes univariés, ce qui signifie qu'elle ne peut pas décomposer un ensemble de variables. Cependant, à l'instar de l'ADMM, cette méthode tire pleinement parti des fonctions objectif séparables, en particulier lorsque certaines parties de la fonction ne sont pas différentiables. Ses principaux avantages résident dans sa rapidité de convergence et sa stabilité pour les fonctions convexes.

Dans la section 5.3, le tableau 5.14 reprend les avantages et inconvénients théoriques mais aussi pratiques de ces méthodes.

3.5.2 Exemple démonstratif : paraboloïde hyperbolique

Dans l'intention de comparer, les méthodes d'optimisation alternée (AO), des directions alternées (ADMM) et du gradient proximal accéléré (APG), nous étudions le problème suivant : soit la fonction définie f de l'ensemble compact $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 = [-10, 10] \times [-10, 10]$ dans $\mathbb{R}^2$ voir la figure 3.5.1,

$$\min_{\boldsymbol{x} \in \mathcal{K}} f(\boldsymbol{x}) = x_1^2 + 0.2x_1x_2 - x_2^2 + 6x_2. \tag{3.31}$$

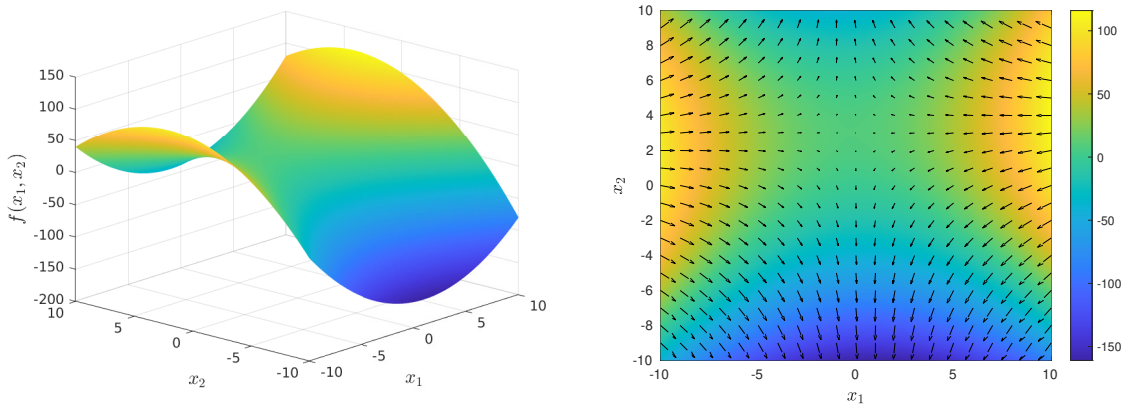
Méthode d'optimisation alternée (AO)

Notons que $\boldsymbol{x} \in \mathcal{K}$ peut s'écrire sous les contraintes d'inégalités

$$\begin{aligned} x_1 - 10 &\leq 0, \\ -x_1 - 10 &\leq 0, \\ x_2 - 10 &\leq 0, \\ -x_2 - 10 &\leq 0. \end{aligned}$$

Afin d'appliquer cette méthode, il est préalablement nécessaire de décomposer l'ensemble $\boldsymbol{x}$ des variables. La seule décomposition possible est $\boldsymbol{x}_1 = (x_1)$ et $\boldsymbol{x}_2 = (x_2)$. Les sous-problèmes sont résolus en calculant les conditions d'optimalité du premier ordre. Soit le Lagrangien associé à f,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \lambda_{1+}(x_1 - 10) + \lambda_{1-}(-x_1 - 10) + \lambda_{2+}(x_2 - 10) + \lambda_{2-}(-x_2 - 10).$$



(a) en 3D. (b) en 2D.

FIGURE 3.5.1 – Représentation de f (paraboloïde hyperbolique).

Nous supposons qu'à chaque itération les contraintes sont respectées, $\forall k, \boldsymbol{x}^k \in \mathcal{K}$ donc $\boldsymbol{\lambda} = 0$. Les conditions d'optimalité sont réduites à la fonction objectif. Il suffit de l'annuler pour la mise à jour des variables, $\frac{\partial f}{\partial x_1} = 2x_1 + 0.2x_2$ et $\frac{\partial f}{\partial x_2} = 0.2x_1 - 2x_2 + 3$. Ainsi, partant d'un $\boldsymbol{x}^0 \in \mathcal{K}$, les mises à jour des variables sont :

1. $x_1^{k+1} = -0.1x_2^k$
2. $x_2^{k+1} = 0.1x_1^{k+1} + 3$

Nous vérifions que les contraintes sont bien toujours respectées.

Méthode du gradient proximal accéléré (APG)

La condition $\boldsymbol{x} \in \mathcal{K}$ est remplacée par une fonction indicatrice correspondante $\mathbb{I}_{\mathcal{K}}$. Son opérateur proximal associé $prox_{\mathbb{I}_{\mathcal{K}}}$ est la projection sur l'ensemble $\mathbb{I}_{\mathcal{K}}$: $\prod_{\mathcal{K}} = \begin{pmatrix} min(max(x_1, -10), 10) \\ min(max(x_2, -10), 10) \end{pmatrix}$.

D'autre part f est Lipzitchienne de constance $L = \sqrt{4,04}$.

En effet, nous avons,

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} 2x_1 + 0.2x_2 \\ 0.2x_1 - 2x_2 + 3 \end{pmatrix}.$$

donc

$$\begin{aligned} \| \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) \| &= \| \begin{pmatrix} 2(x_1 - y_1) + 0.2(x_2 - y_2) \\ 0.2(x_1 - y_1) - 2(x_2 - y_2) \end{pmatrix} \| \\ &= \sqrt{(2(x_1 - y_1) + 0,2(x_2 - y_2))^2 + (0,2(x_1 - y_1) - 2(x_2 - y_2))^2} \\ &= \sqrt{4,04\left((x_1 - y_1)^2 + (x_2 - y_2)\right)} \\ &\leq \sqrt{4,04} \| \boldsymbol{x} - \boldsymbol{y} \| . \end{aligned}$$

Partant de $x^0 = y^0$ aléatoire $t_1 = 1$ et $\delta = \frac{1}{L}$, à chaque itération k

1. $\boldsymbol{x}^{k+1} = \prod_{\mathcal{K}}(\boldsymbol{y}^k - \delta\nabla f(\boldsymbol{y}^k))$
2. $t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1+4t_k^2}\right)$
3. $\boldsymbol{y}^{k+1} = \boldsymbol{x}^{k+1} + (t_k - 1)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)/t^{k+1}$.

Méthode des directions alternées (ADMM)

L'idée est de séparer les variables x_1 et x_2 dans l'intention de décomposer le problème en sous-problèmes plus simples à résoudre. Nous introduisons une variable auxiliaire, z_1, stockée dans le vecteur des variables auxiliaires $\boldsymbol{z} = (z_1)$. Le problème est réécrit

$$\min_{\boldsymbol{x},\boldsymbol{z}\in\mathcal{K},\mathcal{K}_2} f(x_1, z_1) = x_1^2 + 0.2x_1z_1 - z_1^2 + 6z_1, \tag{3.32}$$

avec la contrainte $x_2 - z_1 = 0$.
Le Lagrangien augmenté associé est

$$\mathcal{L}_r(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) \;=\; \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) + \gamma(x_2 - z_1) + \frac{r}{2} \parallel x_2 - z_1 \parallel_2^2 .$$

Appliquer la méthode ADMM, revient à décomposer la minimisation selon les variables du problème $\boldsymbol{x}$ et auxiliaires $\boldsymbol{z}$ puis à les coordonner grâce au multiplicateur $\boldsymbol{\gamma}$. Par l'ajout de la variable auxiliaire z_1 nous avons isolé les variables x_1 et x_2. Cela permet de décomposer également la minimisation de $\boldsymbol{x}$.

1. $\boldsymbol{x}^k = \underset{\boldsymbol{x}}{\text{argmin}}\mathcal{L}_r(\boldsymbol{x}, \boldsymbol{z}^{k-1}, \boldsymbol{\lambda}^{k-1}, \boldsymbol{\gamma}^{k-1}) \Longleftrightarrow \begin{cases} x_1^k = \underset{x_1}{\text{argmin}}\mathcal{L}_r(x_1, \boldsymbol{z}^{k-1}, \boldsymbol{\lambda}^{k-1}) \\ x_2^k = \underset{x_2}{\text{argmin}}\mathcal{L}_r(x_2, \boldsymbol{\lambda}^{k-1}) \end{cases}$
2. $\boldsymbol{z}^k = \underset{\boldsymbol{z}}{\text{argmin}}\mathcal{L}_r(\boldsymbol{x}^k, \boldsymbol{z}, \boldsymbol{\gamma}^{k-1}) \Longleftrightarrow \frac{\partial\mathcal{L}_r(\boldsymbol{x}^k, z_1, \gamma^{k-1})}{\partial z_1} = 0$
   $\Longleftrightarrow z_1^k = \frac{1}{2+r}((0.2 + r)x_1^{k+1} + 6 - \gamma^{k-1}))$
3. $\gamma^k = \gamma^{k-1} + r^{k-1}(x_2^k - z_1^k)$.

Pour x_1 les conditions d'optimalité KKT s'écrivent :

$$\begin{cases} 2x_1 + 0.2z_1^{k-1} + \lambda_{1+} - \lambda_{1-} = 0, \\ \lambda_{1+}(x_1 - 10) = 0, \\ \lambda_{1-}(-x_1 - 10) = 0. \end{cases} \tag{3.33}$$

Les deux dernières équations sont les conditions de complémentarité.
En résolvant

$$x_1^k = \frac{1}{2}(\lambda_{1-}^{k-1} - 0.2z_1^{k-1} - \lambda_{1+}^{k-1}),$$

trois cas sont alors possibles,
— Soit $x_1^k \in]-10, 10[\Longrightarrow \lambda_{1+}^k = \lambda_{1-}^k = 0$,
— Soit $x_1^k \leq -10$ alors on pose $x_1^k = -10 \Longrightarrow \lambda_{1+}^k = 0, \lambda_{1-}^k = -20 + 0.2z_1^{k-1}$,
— Soit $x_1^k \geq 10$ alors on pose $x_1^k = 10 \Longrightarrow \lambda_{1+}^k = -20 - 0.2z_1^{k-1}, \lambda_{1-}^k = 0$.

Pour x_2 les conditions d'optimalité KKT s'écrivent :

$$\begin{cases} \lambda_{2+} - \lambda_{2-} + \gamma^{k-1} + r(x_2 - z_1^{k-1}) = 0, \\ \lambda_{2+}(x_2 - 10) = 0, \\ \lambda_{2-}(-x_2 - 10) = 0. \end{cases} \tag{3.34}$$

Les deux dernières équations sont les conditions de complémentarité.
En résolvant

$$x_2^k = \frac{1}{r}(\lambda_{2-}^{k-1} - \lambda_{2+}^{k-1} - \gamma^{k-1}) + z_1^{k-1},$$

trois cas sont alors possibles,
— Soit $x_2^k \in]-10, 10[\Longrightarrow \lambda_{2+}^k = \lambda_{2-}^k = 0$,
— Soit $x_2^k \leq -10$ alors on pose $x_2^k = -10 \Longrightarrow \lambda_{2+}^k = 0, \lambda_{2-}^k = \gamma^{k-1} - r(z_1^{k-1} + 10)$,
— Soit $x_2^k \geq 10$ alors on pose $x_2^k = 10 \Longrightarrow \lambda_{2+}^k = r(z_1^{k-1} - 10) - \gamma^{k-1}, \lambda_{2-}^k = 0$.
Il faut prendre r suffisamment grand ici $r = 10$.

Résultats
Dans ce test, la condition d'arrêt est $\frac{\|\boldsymbol{x}^k - \boldsymbol{x}^{k+1}\|}{\|\boldsymbol{x}^k\|} < 10^{-6}$. Les trois algorithmes ont été testés avec deux initialisations différentes ; au sommet de la fonction $\boldsymbol{x}^0 = [10, 3]$ et $[0, 0]$.
Les tableaux 3.1 et 3.2 présentent la convergence des méthodes : le nombre d'itérations et la solution du problème $\boldsymbol{x}^*$. Afin de mieux analyser le comportement des méthodes, les figures 3.5.2 permettent de suivre le parcours de minimisation : les différentes solutions itératives sont représentées par des losanges rouges et le sens est donné par les flèches blanches.

— *AO :* En partant du point culminant de la fonction, l'algorithme d'optimisation alternée est descendu en quatre itérations vers le col de la fonction $\boldsymbol{x}_{col} = [0, 3]$. La méthode converge vers le point stationnaire $\bar{\boldsymbol{x}}$ de la fonction $(\nabla f(\bar{\boldsymbol{x}}) = 0), \bar{\boldsymbol{x}} = \boldsymbol{x}_{col}$. Lorsque le point de départ $\boldsymbol{x}^0 = [0, 0]$ est plus bas que le point $\bar{\boldsymbol{x}}$ $(f(\bar{\boldsymbol{x}}) > f(\boldsymbol{x}^0))$, la méthode remonte jusqu'au point stationnaire.
— *ADMM :* Pour le premier test, la méthode passe par le point stationnaire $\bar{\boldsymbol{x}}$ puis continue de descendre vers un minimum local $\boldsymbol{x}_{min} = [-1, 10]$ en un total de sept itérations. La méthode converge vers un minimum local, qui peut être global selon l'initialisation. Dans le deuxième test, $\boldsymbol{x}^0 = [0, 0]$, la méthode converge vers le minimum global $\boldsymbol{x}_{max} = [-1, -10]$ en six itérations.
— *APG :* De même qu'*ADMM*, la méthode converge vers un minimum local, $\boldsymbol{x}_{min}$ puis $\boldsymbol{x}_{max}$. Dans le premier test, l'APG ne passe pas par le point stationnaire $\bar{\boldsymbol{x}}$ mais s'oriente déjà dans la direction de $\boldsymbol{x}_{min}$ sous l'effet de l'inertie. Il est important de souligner que la pente est symétrique par rapport à la droite passant par les sommets $[10, 3]$ et $[-10, 3]$, l'orientation d'un côté ou de l'autre est arbitraire. En prenant une initialisation légèrement plus proche que le minimum global par exemple $\boldsymbol{x}^0 = [10, 2.99]$, la méthode convergerait vers le minimum global (même

	AO	APG	ADMM
Itérations	4	9	7
Convergence $\boldsymbol{x}^*$	[0,3]	[-1,10]	[-1,10]

TABLEAU 3.1 – Résultats de l'optimisation selon les différentes méthodes $\boldsymbol{x}^0 = [10, 3]$.

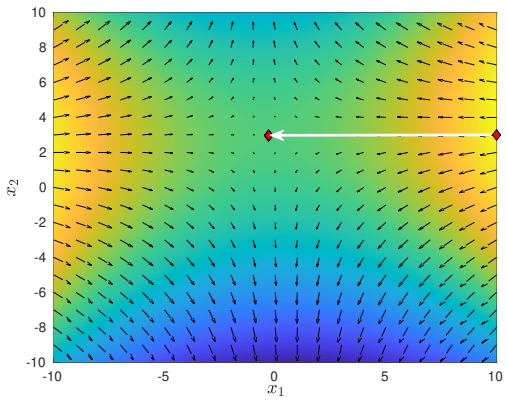
	AO	APG	ADMM
Itérations	5	9	6
Convergence $\boldsymbol{x}^*$	[0,3]	[1,-10]	[1,-10]

TABLEAU 3.2 – Résultats de l'optimisation selon les différentes méthodes $\boldsymbol{x}^0 = [0, 0]$.

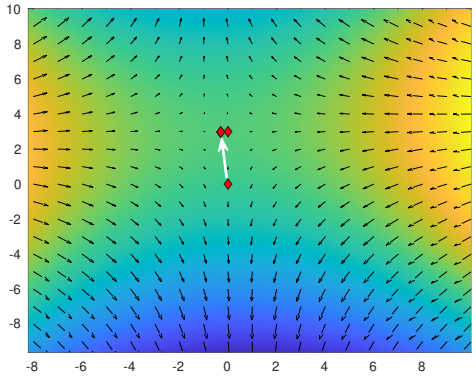
observation pour *ADMM*). L'inertie accumulée pendant la descente peut expliquer le nombre d'itérations plus important qu'*ADMM* pour atteindre la stabilité. L'inertie initiale dépendant de la pente à l'origine, on observe clairement qu'avec la deuxième initialisation, le premier déplacement est plus petit.

3.6 Conclusion

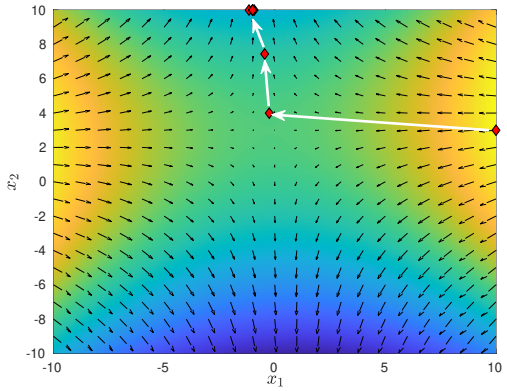
Dans ce chapitre, nous avons introduit l'optimisation mathématique et ses notions fondamentales. Nous avons présenté trois méthodes applicables aux modèles de classification non supervisée : l'optimisation alternée, la méthode des directions alternées et la méthode du gradient accéléré. À l'aide d'un exemple simple, nous avons illustré les avantages et les inconvénients de ces méthodes. L'optimisation alternée est simple à mettre en œuvre en classification non supervisée, mais sa convergence vers un point stationnaire ne garantit pas nécessairement une convergence vers un point-selle. Dans ce contexte, l'utilisation de méthodes plus sophistiquées comme ADMM et APG se révèle pertinente.



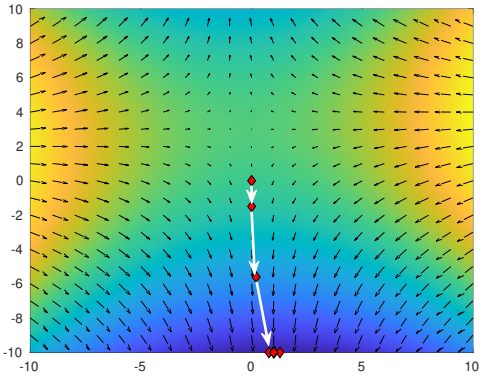
(a) AO | $\boldsymbol{x}^0 = [10, 3] \rightarrow \boldsymbol{x}^4 = [0, 3]$.



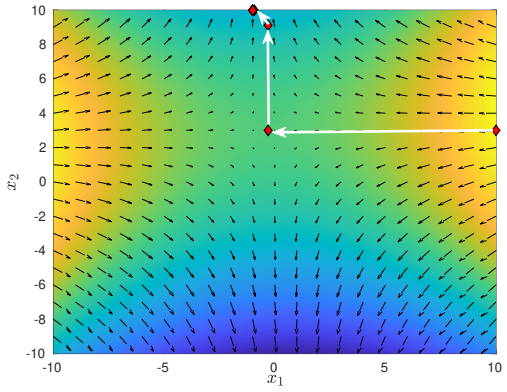
(b) AO | $\boldsymbol{x}^0 = [0, 0] \rightarrow \boldsymbol{x}^5 = [0, 3]$.



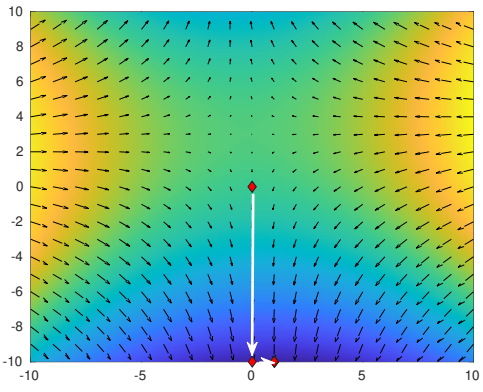
(c) APG | $\boldsymbol{x}^0 = [10, 3] \rightarrow \boldsymbol{x}^9 = [-1, 10]$.



(d) APG | $\boldsymbol{x}^0 = [10, 3] \rightarrow \boldsymbol{x}^9 = [1, -10]$.



(e) ADMM | $\boldsymbol{x}^0 = [10, 3] \rightarrow \boldsymbol{x}^7 = [-1, 10]$.



(f) ADMM | $\boldsymbol{x}^0 = [0, 0] \rightarrow \boldsymbol{x}^6 = [1, -10]$.

FIGURE 3.5.2 – Affichage du parcours des méthodes d'optimisation.

Deuxième partie

Contributions

Chapitre 4

Mesure d'évaluation pour FCM avec la distance de Mahalanobis

Contents

Les mesures d'évaluation permettent d'évaluer la qualité du partitionnement des données, et nous en avons présenté quelques-unes dans la section 2.5. Cependant, il est essentiel de choisir des mesures qui tiennent compte des spécificités des algorithmes de classification non supervisée. La distance de Mahalanobis est une caractéristique particulière des modèles étudiés dans notre recherche. Malheureusement, aucun des indices existants ne peut évaluer correctement les partitions générées par ces algorithmes.
Dans ce chapitre, nous illustrons la problématique rencontrée (section 4.1), nous détaillons notre approche et la mesure que nous proposons (section 4.2). Enfin, les résultats démontrent la pertinence de notre nouvel indice et ses limites (section 4.3).

4.1 Problématique

Pour HCM et ses variantes, les mesures évaluent la similarité et/ou la compacité. Comme la plupart des extensions de HCM emploient la distance euclidienne, les mesures sont généralement conçues pour cette distance. Dans le cas de l'utilisation de la distance exponentielle, Gath et Geva ont proposé deux mesures [64] évaluant le volume de l'ellipse $det(\boldsymbol{S})$: Fuzzy Hypervolume et Average Partition Density. Ils ont développé ces indices dans le cadre de leur modèle basé sur une distance exponentielle sans contrôle du déterminant de la matrice $\boldsymbol{S}$. En revanche, dans le modèle de Gustafson et Kessel [61] le volume de l'ellipse (le déterminant) est constant, par défaut fixé à $det(\boldsymbol{S}) = 1$.

A ce jour, il n'existe aucune mesure évaluant géométriquement la séparabilité/compacité de la partition à l'aide d'une distance adaptative qui permettraient de tenir compte des formes ellipsoïdales.

Nous avons donc décidé d'améliorer l'indice de Xie et Beni (2.48) [6] puisqu'il est très répandu dans la littérature [143–146] et qu'il évalue à la fois la compacité et la séparabilité d'une partition floue et sa formulation est aisément adaptable. L'amélioration doit tenir compte de la distance de Mahalanobis, spécificité de notre modèle étudié.

4.2 Amélioration de XB : $XBMW$

Pour rappel, l'indice de Xie-Beni est calculé en employant le rapport compacité / séparabilité :

$$XB = \frac{\text{compacité}}{\text{séparabilité}} = \frac{\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{m} \parallel \boldsymbol{v}_j - \boldsymbol{x}_i \parallel^2}{n \times \min\limits_{j \neq j'}(\parallel \boldsymbol{v}_j - \boldsymbol{v}'_j \parallel^2)}. \tag{4.1}$$

4.2.1 Étude de la compacité

La compacité est définie par extension de l'indice PC [84] (éq. 2.43), où les degrés d'appartenance sont multipliés par la distance au centre de gravité :

$$\text{compacité}(XB) = \frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{2}(\boldsymbol{x}_i - \boldsymbol{v}_j)^{\top}(\boldsymbol{x}_i - \boldsymbol{v}_j). \tag{4.2}$$

Exemple 4.2.1: Analyse de la compacité

Considérons un jeu de données en deux dimensions comportant deux classes bien distinctes, illustré dans la 4.2.1. Les classes sont définies par leur centre de gravité $\boldsymbol{v}$ et matrice de covariance $\boldsymbol{\Sigma}$:

1. $\omega_1 : \boldsymbol{v}_1 = [-2, 0], \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$,
2. $\omega_2 : \boldsymbol{v}_2 = [2, 0], \boldsymbol{\Sigma}_2 = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$.

Les objets sont distribués aléatoirement grâce à la fonction *mvrnd* de MATLAB.

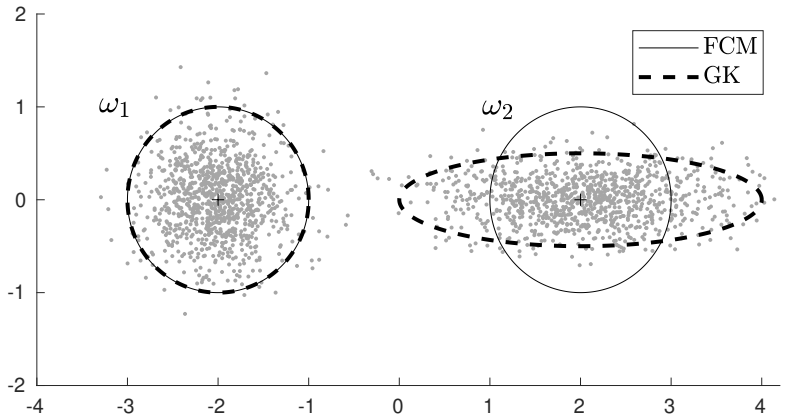


FIGURE 4.2.1 – Ensemble de données à deux classes.

La première classe ω_1, à gauche, a une structure sphérique, tandis que la seconde est de forme ellipsoïdale. Les algorithmes FCM et GK ont été appliqués à ce jeu de données et les matrices de co-variances obtenues sont présentées sur la figure 4.2.1.
Pour la première classe ω_1, les deux méthodes détectent la même structure. La compacité est donc la même. Pour la seconde classe ω_2, la méthode GK détecte mieux la forme réelle de la classe et devrait avoir une meilleure compacité que l'algorithme FCM. Néanmoins, comme la compacité mesurée par l'indice de Xie-Beni utilise la distance euclidienne, les valeurs sont similaires.

Nous remarquons que la compacité correspond à la fonction objectif de Fuzzy C-means divisée par le nombre d'objet n. Pour rappel, HCM minimise les distances intra-classes, donc la compacité. Nous proposons comme nouvelle formule de compacité d'utiliser la fonction objectif de Gustafson et Kessel (éq. 2.31) divisée par n :

$$\text{compacité}(M) = \frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^2(\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j(\boldsymbol{x}_i - \boldsymbol{v}_j). \tag{4.3}$$

La distance intra-classe de l'indice est désormais la distance de Mahalanobis qui généralise la distance euclidienne. Ainsi, lorsque toutes les classes ont une forme sphérique, $\boldsymbol{S}_j = I, \forall j \in [1, c]$, la nouvelle mesure de compacité est identique à la mesure de compacité de l'indice de Xie Beni.
Nous vérifions la justesse de cette nouvelle formulation avec l'exemple de la figure 4.2.1. La compacité de la première classe, ω_1, est toujours identique pour les deux modèles, mais la compacité de la seconde classe, ω_2, est réduite pour GK. Ainsi, la compacité de la partition générée par GK est plus faible. D'après le tableau 4.1, nous retrouvons numériquement ce résultat avec la nouvelle formulation.

	FCM	GK
compacité(XB)	0.71	0.71
compacité(M)	0.71	0.42

TABLEAU 4.1 – Comparaison de la compacité.

4.2.2 Étude de la séparabilité

La séparabilité correspond à la distance euclidienne minimale entre deux centroïdes. Cette formulation est identique à celles existantes pour les indices de Dunn D (éq. 2.40) et Davies-Bouldin DB (éq. 2.41) :

$$\text{séparabilité}(XB) = \min_{j,k\in[1,c],j\neq k} \| \boldsymbol{v}_j - \boldsymbol{v}_k \|_2^2 . \tag{4.4}$$

Pour une classe sphérique, chaque attribut a la même importance. Cependant, avec les formes ellipsoïdales, nous disposons d'informations sur l'importance des attributs pour chaque classe. En évaluant la séparabilité avec la distance euclidienne, l'indice annule tout l'avantage du modèle GK.

Exemple 4.2.2: Analyse de la séparabilité

Prenons l'exemple de trois classes dont la deuxième et la troisième partagent le même centroïde mais ont des ellipses différentes (cf. figure 4.2.2) :

1. $\omega_1 : \boldsymbol{v}_1 = [-2, 0], \boldsymbol{\Sigma}_1 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$.
2. $\omega_2 : \boldsymbol{v}_2 = [2, 0], \boldsymbol{\Sigma}_2 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$.
3. $\omega_3 : \boldsymbol{v}_2 = \boldsymbol{v}_3 = [2, 0], \boldsymbol{\Sigma}_3 = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$.

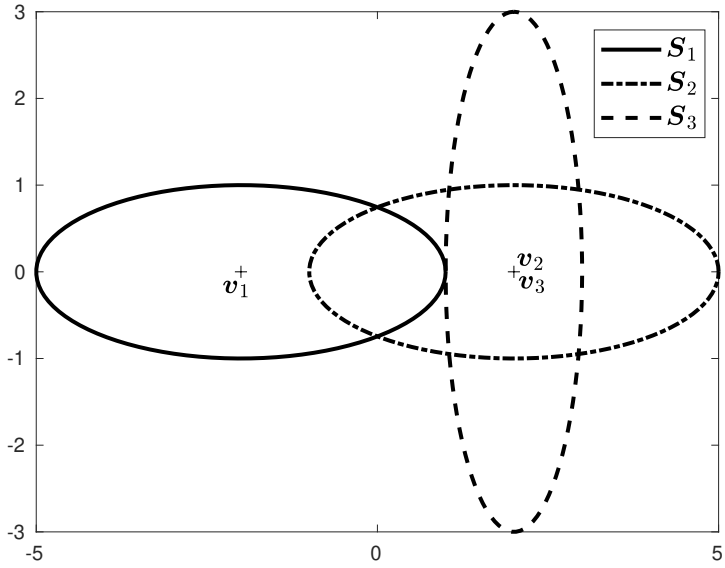


FIGURE 4.2.2 – Deux classes partageant le même centroïde mais ayant deux formes différentes.

Comme $\boldsymbol{v}_2 = \boldsymbol{v}_3$, la distance euclidienne $d(\omega_1, \omega_2)$ entre les classes 1 et 2 est la même qu'entre les classes 1 et 3, $d(\omega_1, \omega_2) = d(\omega_1, \omega_3)$. Or nous remarquons dans cet exemple que la classe 3 accorde beaucoup plus d'importance à l'attribut porté par l'axe des ordonnées, contrairement aux deux autres classes. Les classes 1 et 2 sont plus proches, la partition formée par les classes 1 et 2 est moins séparable que celle formée par les classes 1 et 3.

Pour mesurer la distance entre deux classes ellipsoïdales, nous devons prendre en compte à la fois leur centre de gravité $\boldsymbol{v}$ et leur matrice $\boldsymbol{S}$. En vue de comparer deux ellipses, nous avons considéré les classes $\omega(\boldsymbol{v}, \boldsymbol{S})$ comme des distributions gaussiennes multivariées $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: la moyenne étant le centroïde $\boldsymbol{\mu} = \boldsymbol{v}$ et la matrice de variance-covariance étant l'inverse de la matrice de distance $\boldsymbol{\Sigma} = \boldsymbol{S}^{-1}$. Cette association nous ouvre les portes d'un domaine bien connu, les mesures de dissimilarité entre deux distributions. Il en existe de nombreuses : Kullback–Leibler divergence [147, 148], Hellinger distance [149, 150], Bhattacharyya distance [151, 152] ...

Nous avons choisi la distance de Wasserstein [153, 154] pour plusieurs raisons. C'est une vraie métrique contrairement à de nombreuses *"pseudo"*-distances statistiques qui ne vérifient pas toutes les propriétés définissant une distance. Il est très important dans notre cas de respecter l'inégalité triangulaire puisque l'on compare les classes deux à deux. Elle généralise la distance euclidienne et son interprétation est très simple. Issue

des travaux sur le problème du transport optimal, cette distance modélise la difficulté de transposer un tas de terre vers un autre, d'où son autre nom de distance de déplacement de la terre (EMD). Elle somme deux quantités, qui mesure respectivement la translation (distance euclidienne) et la rotation nécessaire pour passer d'une distribution à une autre.
La distance de 2-Wasserstein entre les deux gaussiennes $g_1 = \mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ et $g_2 = \mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ est :

$$W_2(g_1, g_2)^2 = \| \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \|_2^2 + tr\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2\sqrt{\boldsymbol{\Sigma}_2^{1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2^{1/2}}\right), \tag{4.5}$$

où $\| \, . \, \|_2$ est la norme euclidenne et $tr(.)$ la fonction trace. Ainsi, nous donnons la distance inter-classe comme la distance 2-Wasserstein entre les deux classes, selon la formule suivante :

$$W_2(\omega_j, \omega_k)^2 = \| \boldsymbol{v}_j - \boldsymbol{v}_k \|_2^2 + tr\left(\boldsymbol{S}_j^{-1} + \boldsymbol{S}_k^{-1} - 2\sqrt{\boldsymbol{S}_k^{-1/2}\boldsymbol{S}_j^{-1}\boldsymbol{S}_k^{-1/2}}\right). \tag{4.6}$$

La distance de Wasserstein dans ce cas est aussi connue sous le nom de distance d'inception de Frechet (FID) [155].

Propriétés de la distance de Wasserstein

Proposition 4.2.1: Généralisation de la distance euclidienne

Soit deux gaussiennes $g_1 = \mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ et $g_2 = \mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, ayant la même matrice de covariance $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$ alors,

$$W_2(g_1, g_2)^2 = d_2(g_1', g_2')^2, \tag{4.7}$$

où $d_2(.,.)^2$ est la distance euclidienne définit dans la section 2.4.2.1. Ainsi dans le cas où les classes ont la même forme, la distance de Wasserstein se réduit à la distance euclidienne.

Démonstration. Puisque $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, les termes dans la trace s'annulent.

$$\begin{aligned} W_2(g_1, g_2)^2 &= ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2 + Tr\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2\sqrt{\sqrt{\boldsymbol{\Sigma}_1}\boldsymbol{\Sigma}_2\sqrt{\boldsymbol{\Sigma}_1}}\right) \\ &= ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2. \end{aligned}$$

□

Remarquons que lorsque les classes ont toutes une forme sphérique comme dans FCM, alors toutes les matrices de covariance sont égales entre elles, $\boldsymbol{\Sigma} = \boldsymbol{I}$. Dans ce cas, la distance de Wasserstein est réduite à la distance euclidienne.

Proposition 4.2.2: Invariance par rotation

Soit deux gaussiennes $g_1 = \mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ et $g_2 = \mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, et soit une matrice de rotation $\boldsymbol{R}$ définissant deux autres gaussiennes $g'_1 = \mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^\top)$ et $g'_2 = \mathcal{N}_1(\boldsymbol{\mu}_2, \boldsymbol{R}\boldsymbol{\Sigma}_2\boldsymbol{R}^\top)$, alors

$$W_2(g_1, g_2)^2 = W_2(g'_1, g'_2)^2. \tag{4.8}$$

La distance de Wasserstein est invariante par rotation des ellipses.

Démonstration. La matrice de rotation $\boldsymbol{R}$ est une matrice orthogonale donc $\boldsymbol{R}^\top = \boldsymbol{R}^{-1}$. Nous partons de la distance de Wasserstein entre les deux gaussiennes g'_1, g'_2. Nous utilisons les propriétés de la trace notamment la linéarité et l'invariant de similitude.

$$\begin{aligned}
W_2(g'_1, g'_2)^2 &= ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2 + Tr\left(\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T + \boldsymbol{R}\boldsymbol{\Sigma}_2\boldsymbol{R}^T - 2\sqrt{\sqrt{\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T}\boldsymbol{R}\boldsymbol{\Sigma}_2\boldsymbol{R}^T\sqrt{\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T}}\right) \\
&= ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2 + Tr\left(\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T\right) + Tr\left(\boldsymbol{R}\boldsymbol{\Sigma}_2\boldsymbol{R}^T\right) \\
&\quad - 2Tr\left(\sqrt{\sqrt{\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T}\boldsymbol{R}\boldsymbol{\Sigma}_2\boldsymbol{R}^T\sqrt{\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T}}\right) \\
&= ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2 + Tr\left(\boldsymbol{\Sigma}_1\right) + Tr\left(\boldsymbol{\Sigma}_2\right) - 2Tr\left(\sqrt{\sqrt{\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T}\boldsymbol{R}\boldsymbol{\Sigma}_2\boldsymbol{R}^T\sqrt{\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T}}\right) \\
&= ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2 + Tr\left(\boldsymbol{\Sigma}_1\right) + Tr\left(\boldsymbol{\Sigma}_2\right) - 2Tr\left(\sqrt{\sqrt{\boldsymbol{\Sigma}_1 T}\boldsymbol{\Sigma}_2\sqrt{\boldsymbol{\Sigma}_1}}\right) \\
&= W_2(g_1, g_2)^2.
\end{aligned}$$

□

Après ces deux propriétés intéressantes de Wasserstein, nous proposons la nouvelle formule de la séparabilité qui se base sur cette distance :

$$\text{séparabilité}(W) = \min_{j,k\in[1,c],j\neq k} W_2(\omega_j, \omega_k)^2. \tag{4.9}$$

Dans l'exemple de la figure 4.2.2, nous savons qu'il existe une plus grande séparabilité entre ω_1, ω_3 qu'entre ω_1, ω_2. Contrairement à la formulation de XB, nous retrouvons ce résultat d'après le tableau 4.2 puisque la séparabilité passe de 16 à 24.

	(ω_1, ω_2)	(ω_1, ω_3)
séparabilité(XB)	16	16
séparabilité(W)	16	24

TABLEAU 4.2 – Comparaison de la séparabilité.

4.2.3 Nouvelle formule : $XBMW$

Finalement, nous proposons l'indice $XBMW$ qui étend l'indice XB en évaluant la distance intra-classes avec la distance de Mahalanobis et la distance inter-classes avec la distance de Wassertein. Pour une partition floue, l'indice à minimiser se formule

$$(\downarrow)XBMW = \frac{\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^2 (\boldsymbol{x}_i - \boldsymbol{v}_j)^\top \boldsymbol{S}_j (\boldsymbol{x}_i - \boldsymbol{v}_j)}{n \min_{j,k\in[1,c],j\neq k} W_2(\omega_j, \omega_k)^2}. \tag{4.10}$$

XB est alors le cas particulier d'une partition avec des classes sphériques puisque les distances de Mahalanobis et de Wasserstein généralisent la distance euclidienne.

4.3 Expérimentations numériques

4.3.1 Méthodologie

Dans cette section, nous évaluons la performance de notre indice. Généralement dans la littérature, les indices sont comparés sur leur capacité à retrouver le bon nombre de classes des jeux de données [6, 86, 89].

Nous ne réalisons pas une comparaison exhaustive de tous les indices mais seulement entre XB et $XBMW$, sachant tout l'intérêt de XB, nous souhaitons déterminer ce qu'apporte concrètement $XBMW$. En théorie, notre indice doit mieux évaluer la partition issue d'algorithmes utilisant la distance de Mahalanobis.

Le protocole expérimental est donc basé sur la vérification de cette hypothèse. Dans un premier temps, nous appliquons les algorithmes de FCM et GK sur des jeux de données. Chaque algorithme est exécuté 10 fois avec différentes initialisations aléatoires des centroïdes et seule la partition minimisant la fonction objectif est conservée. Nous comparons les deux partitions obtenues selon un indice externe à maximiser, l'ARI (éq. 2.37) et les indices internes XB (éq. 2.48) et $XBMW$ (éq. 4.10) à minimiser. L'indice externe nous sert de référence puisqu'un meilleur partitionnement se traduit par une valeur de l'ARI plus élevée. Nous calculons ensuite un coefficient de correspondance simple SMC entre la différence d'ARI et d'indice interne. En vue de vérifier que le comportement de l'ARI, accentué ou diminué entre FCM et GK, est similaire au comportement de l'indice interne évalué.

$$SMC = \frac{VP + VN}{VP + VN + FP + FN}.$$

Lorsque l'ARI augmente et que l'indice diminue, il s'agit d'un vrai positif (VP), à l'inverse si l'indice augmente, il s'agit d'un faux négatif (FN). Si l'ARI diminue et que l'indice diminue, il s'agit d'un faux positif (FP), mais si l'indice augmente, il s'agit d'un vrai négatif (VN). Nous formons le tableau 4.3 de correspondance simple.

		Evaluation interne (XB - $XBMW$)	
		GK<FCM	GK>FCM
Evaluation externe (ARI)	GK>FCM	Vrai positif (VP)	Faux négatif (FN)
	GK<FCM	Faux positif (FP)	Vrai négatif (VN)

Tableau 4.3 – Correspondance simple.

Les expérimentations ont été réalisées à l'aide du logiciel MATLAB R2020a sur un ordinateur équipé d'un processeur Intel Core i5 de 10ème génération et de 16 Go de RAM sous Linux.

4.3.2 Jeux de données utilisés

Nous comparons les indices avec 17 jeux de données. Parmi ces jeux de données, six ont été créées pour illustrer théoriquement l'avantage de notre méthode. La génération de ces jeux de données synthétiques est donnée dans l'annexe A.2.1. Il est intéressant de regarder leurs représentations visuelles, figures A.2.1a-A.2.1f. Nous avons également utilisé 11 jeux couramment employés dans la littérature et pour lesquels FCM donne de bons partitions selon l'ARI. Les deux premiers présentés dans l'annexe A.3 sont deux jeux de données synthétiques. Les neuf autres sont issues de la bibliothèque de l'UCI [1], leurs origines et caractéristiques sont détaillées en annexe A.4. Une procédure de standardisation a été réalisé sur les jeux de données afin que chaque attributs ait une importance égale. Le détail du pré-traitement est présenté en annexe A.1.1.

4.3.3 Résultats

Le tableau 4.4 présente les coefficients de correspondance. De manière générale, nous constatons que notre indice affiche une précision de 76 %, tandis que celle de XB est seulement de 35 %.

	VP	VN	FP	FN	SMC
XB	1	**5**	0	11	0.35
$XBMW$	**11**	2	3	**1**	**0.76**

Tableau 4.4 – Correspondance entre ARI et XB, $XBMW$.

Les tableaux 4.5, 4.6 et 4.7 détaillent les résultats par jeu de données. Lorsque la distance euclidienne est employée, l'indice $XBMW$ est théoriquement équivalent à XB. C'est pourquoi nous observons que les indices ont la même valeur pour n'importe quel jeu de données appliqué à FCM.

Dans le but d'avoir une analyse plus fine, il est nécessaire de s'intéresser aux jeux synthétiques où la pertinence d'utiliser la distance de Mahalanobis est connue. Pour

1. https://archive.ics.uci.edu/

	FCM	GK	
ARI	0.42	1	
XB	0.18	0.61	FN
$XBMW$	0.18	0.21	FN

(a) T1.

	FCM	GK	
ARI	0.79	0.97	
XB	0.18	0.28	FN
$XBMW$	0.18	0.13	**VP**

(b) T2.

	FCM	GK	
ARI	0.26	0.86	
XB	0.72	33.3	FN
$XBMW$	0.72	0.005	**VP**

(c) T3.

	FCM	GK	
ARI	0.61	0.91	
XB	0.33	13.5	FN
$XBMW$	0.33	0.01	**VP**

(d) T4.

	FCM	GK	
ARI	0.41	0.93	
XB	0.33	0.68	FN
$XBMW$	0.33	0.31	**VP**

(e) T5.

	FCM	GK	
ARI	0.27	0.96	
XB	0.20	0.53	FN
$XBMW$	0.20	0.14	**VP**

(f) T6.

TABLEAU 4.5 – ARI, XB, $XBMW$ pour les jeux de données tests.

	FCM	GK	
ARI	0.89	0.96	
XB	0.09	0.12	FN
$XBMW$	0.09	0.06	**VP**

(a) Asymetric.

	FCM	GK	
ARI	0.65	0.99	
XB	0.24	0.66	FN
$XBMW$	0.24	0.06	**VP**

(b) Skewed.

TABLEAU 4.6 – ARI, XB, $XBMW$ pour les jeux de données synthétiques.

illustrer nos expérimentations numériques, dans l'annexe B une comparaison visuelle entre les regroupements obtenus par FCM et GK pour les jeux de données prototypes est présentée. Dans tous ces cas, XB se trompe et favorise la distance euclidienne pourtant moins performante. Parmi tous les jeux de données, Drybean est l'unique cas où XB favorise GK comme suggéré par l'ARI, +0.02. C'est l'unique fois sur 17 que XB choisit la distance de Mahalanobis.

Notre métrique est mieux adaptée pour sélectionner la distance optimale. En effet, dans nos expérimentations, $XBMW$ privilégie généralement la bonne distance. Évidemment, il est capable de choisir la distance de Mahalanobis pour laquelle il a été conçu, mais il peut également opter pour la distance euclidienne lorsque celle-ci est la plus pertinente. Nous notons particulièrement que cela se produit lorsque la différence d'ARI est significative, comme c'est le cas pour les jeux de données Wifi et Wine.

	FCM	GK	
ARI	0.34	0.54	
XB	0.35	0.38	FN
$XBMW$	0.35	0.01	**VP**

(a) AF.

	FCM	GK	
ARI	0.68	0.70	
XB	16.55	0.64	**VP**
$XBMW$	16.55	6.10^{-6}	**VP**

(b) DB.

	FCM	GK	
ARI	0.55	0.41	
XB	1.45	0.84	**VN**
$XBMW$	1.45	2.10^{-3}	FP

(c) Glass.

	FCM	GK	
ARI	0.63	0.74	
XB	0.22	0.79	FN
$XBMW$	0.22	0.16	**VP**

(d) Iris.

	FCM	GK	
ARI	0.04	0.26	
XB	7.06	1.15	FN
$XBMW$	7.06	0.10	**VP**

(e) IJL.

	FCM	GK	
ARI	0.77	0.72	
XB	0.21	0.22	**VN**
$XBMW$	0.21	0.01	FP

(f) Seed.

	FCM	GK	
ARI	0.68	0.41	
XB	0.48	2.16	**VN**
$XBMW$	0.48	0.02	FP

(g) WDBC.

	FCM	GK	
ARI	0.82	0.41	
XB	0.34	6.10^4	**VN**
$XBMW$	0.34	1.10^4	**VN**

(h) Wifi.

	FCM	GK	
ARI	0.90	0.33	
XB	0.47	70.0	**VN**
$XBMW$	0.47	4.19	**VN**

(i) Wine.

TABLEAU 4.7 – ARI, XB, $XBMW$ pour les jeux de données UCI.

4.3.4 Limites et discussions

Nous pouvons relever une sensibilité de notre indice au nombre d'attributs nd. En effet, les matrices de covariance $\mathbf{\Sigma} \in \mathbb{R}^{nd \times nd}$ présentes dans la fonction *trace* de la distance de Wasserstein, peuvent augmenter artificiellement la distance entre deux classes. C'est le cas pour le jeu de données WDBC où les centroïdes entre FCM et GK sont presque identiques mais les matrices des distances varient légèrement et faussent les résultats. Cette sensibilité est intrinsèque au modèle FCM.

En revanche, le cas T1 présente la limite la plus importante de notre indice. Dans cette situation, les trois classes ont la même forme, voir la figure B.0.1. Les $\boldsymbol{\mathcal{S}}$ matrices des distances sont identiques, ainsi la distance de Wasserstein est réduite à la distance euclidienne. $XBMW$ subit la même problématique que XB et conclut sur un faux négatif.

4.4 Conclusion

Dans cette étude, l'objectif était de tenir compte de la forme ellipsoïdale des classes pour mesurer la qualité de la partition d'un algorithme utilisant la distance de Mahalanobis. En effet, pour les critères d'évaluation interne, il est important de bien prendre en considération la spécificité du modèle de classification non supervisée. Or nous avons montré la limite des indices existants. Nous avons étendu celui proposé par Xie et Beni XB pour FCM. Nous choisissons de remplacer la distance euclidienne par la distance de Mahalanobis pour mesurer la compacité et par la distance de Wasserstein dans le but d'évaluer la séparabilité $XBMW$.

Avec une technique innovante d'analyse basée sur les critères d'évaluation externe, nous avons testé ces deux indices sur le partitionnement flou généré par FCM et GK. Les résultats sont satisfaisants, nous avons désormais 76% de conformité contrairement à 35% préalablement. Nous apportons un nouveau regard sur les mesures d'évaluation : notre indice constitue un outil idéal pour décider quelle métrique adopter entre la distance euclidienne et la distance de Mahalanobis. Ces résultats ont été présentés à la conférence EUSFLAT [2] [156].

Cette étude est encourageante et offre quelques perspectives. Tout d'abord, il serait intéressant d'approfondir encore l'étude de la séparabilité notamment en renforçant le lien entre les ellipses et les distributions gaussiennes. De plus, nous avons centré notre étude sur l'indice de Xie-Beni, mais il pourrait être intéressant d'adapter d'autres mesures de validation interne. Il est également attrayant d'étendre cette mesure pour l'évaluation des modèles de classification non supervisée évidentielle comme ECM. Enfin, il sera désormais possible de comparer deux méthodes de classification non supervisée qui utilisent toutes deux les distances adaptatives, type distance de Mahalanobis.

2. https://www.eusflat.org/

Chapitre 5

Optimisation par méthodes duales et proximales de FCM avec la distance de Mahalanobis

Contents

Ce chapitre occupe une place centrale dans ma thèse. L'objectif principal n'est pas de créer un nouveau modèle de classification non supervisée, mais plutôt de mieux résoudre son problème de minimisation. Dans le chapitre 3, nous avons identifié les limites de l'approche d'optimisation alternée. Ces limites ont été mises en évidence à travers un exemple simple impliquant deux variables. De plus, lorsque la décomposition des variables est étendue à trois blocs, aucune garantie de convergence n'est assurée. Cela devient particulièrement problématique dans le contexte de FCM avec la distance de Mahalanobis proposée par Gustafson et Kessel, où les variables sont justement réparties en trois blocs distincts : la partition, les centroïdes et les matrices de distance.

En vue d'augmenter les chances de trouver un minimum global, il est courant dans la littérature d'exécuter plusieurs fois les algorithmes de classification non supervisée avec différentes initialisations aléatoires, généralement entre 10 et 20 initialisations. Les solutions qui minimisent au mieux la fonction objectif sont alors conservées.

Dans ce chapitre, nous appliquons la méthode ADMM aux modèles de classification non supervisée floue, FCM et FCM-GK. Le choix d'ADMM est motivé par sa similarité avec l'approche d'optimisation alternée. En effet, les deux méthodes suivent le principe de Gauss-Seidel, ce qui permet de maintenir la pédagogie du modèle. L'application d'ADMM se déroule en deux étapes : d'abord, une reformulation astucieuse de la fonction objectif et du problème est effectuée, en introduisant judicieusement des variables auxiliaires. Ensuite, l'optimisation est réalisée en résolvant les conditions d'optimalité.

Nous présentons dans ce chapitre l'optimisation du problème selon ADMM (section 5.1.1-5.1.3) et l'optimisation des matrices de distance en utilisant l'accélération de Nes-

terov (section 5.1.4). La section suivante (section 5.2) regroupe les expérimentations numériques que nous avons menées. Enfin, nous conclurons par une analyse globale de notre démarche (section 5.3).

5.1 Optimisation de FCM-GK

5.1.1 Reformulation du problème

Rappelons le problème d'optimisation FCM-GK (2.31-2.32), nous recherchons $(\boldsymbol{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}})$ qui minimisent la somme des distance intra-classes :

$$J_{GK}(\boldsymbol{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}) = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m}(\boldsymbol{x}_i - \boldsymbol{v}_j)^{\top}\boldsymbol{S}_j(\boldsymbol{x}_i - \boldsymbol{v}_j), \tag{5.1}$$

sous les contraintes $\forall i, j \in [1, n] \times [1, c]$,

$$u_{ij} \geq 0, \sum_{j=1}^{c} u_{ij} = 1, \sum_{i=1}^{n} u_{ij} > 0, \tag{5.2}$$

$$\det(\boldsymbol{S}_j) = 1. \tag{5.3}$$

Et les deux ensembles des contraintes sont définis,

$$\mathcal{U} = \left\{ \forall i, j \in [1, n] \times [1, c], u_{ij} \geq 0, \quad \sum_{j=1}^{c} u_{ij} = 1, \quad \sum_{i=1}^{n} u_{ij} > 0 \right\},$$

$$\mathcal{S}_1 = \left\{ \forall j \in [1, c], \boldsymbol{S}_j \in \mathbb{R}^{n_d \times n_d} \text{ matrice symétrique positive}, \det(\boldsymbol{S}_j) = 1 \right\}.$$

Nous notons $I_{\mathcal{U}}, I_{\mathcal{S}_1}$ leur fonction caractéristique.
Nous souhaitons optimiser les variables originales du problème $\boldsymbol{U}, \boldsymbol{\mathcal{V}}$ et $\boldsymbol{\mathcal{S}}$ successivement, donc séparément. Pour ce faire, nous ajoutons deux nouvelles variables auxiliaires $\boldsymbol{\mathcal{P}}$ et $\boldsymbol{\mathcal{Q}}$ telles que

$$\boldsymbol{p}_{ij} = u_{ij}\boldsymbol{q}_{ij} = u_{ij}(\boldsymbol{x}_i - \boldsymbol{v}_j), \quad \forall i, j \in [1, n] \times [1, c].$$

$\boldsymbol{\mathcal{Q}}$ permet de séparer $\boldsymbol{U}$ de $\boldsymbol{\mathcal{V}}$, comme le paramètre de fuzzification vaut deux, $m = 2$, nous distribuons les deux u_{ij}. Grâce à $\boldsymbol{\mathcal{P}}$, $\boldsymbol{\mathcal{S}}$ se retrouve isolé dans la fonction objectif qui s'écrit désormais,

$$J(\boldsymbol{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{P}}) = \sum_{i=1}^{n}\sum_{j=1}^{c} \boldsymbol{p}_{ij}^{\top}\boldsymbol{S}_j\boldsymbol{p}_{ij}. \tag{5.4}$$

Pour simplifier l'écriture, nous notons :
$\mathbb{U} = (\boldsymbol{U}, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}})$ l'ensemble des variables du problème et $\mathbb{Q} = (\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{P}})$ l'ensemble des variables auxiliaires. Le problème de minimisation sous contraintes (5.1)-(5.3) devient

$$\min J(\mathbb{U}, \mathbb{Q}) + I_{\mathcal{U}}(\boldsymbol{U}) + I_{\mathcal{S}_1}(\boldsymbol{\mathcal{S}}). \tag{5.5}$$

sous les contraintes

$$\boldsymbol{q}_{ij} = \boldsymbol{x}_i - \boldsymbol{v}_j, \tag{5.6}$$

$$\boldsymbol{p}_{ij} = u_{ij}\boldsymbol{q}_{ij}. \tag{5.7}$$

Les contraintes du couplage linéaire sont définies pour garantir l'équivalence avec le problème original (5.1)-(5.3) en terme de solution. La fonction du Lagrangien augmenté associée au nouveau problème (5.5)-(5.7) est

$$\begin{aligned}\mathcal{L}_r(\mathbb{U},\mathbb{Q},\mathbb{Y}) &= J(\mathbb{U},\mathbb{Q}) + I_{\mathcal{U}}(\boldsymbol{U}) + I_{\mathcal{S}_1}(\boldsymbol{\mathcal{S}}) \\ &\quad + \sum_{i,j}\left[\boldsymbol{y}_{ij}^{\top}(\boldsymbol{q}_{ij} - \boldsymbol{x}_i + \boldsymbol{v}_j) + \boldsymbol{z}_{ij}^{\top}(\boldsymbol{p}_{ij} - u_{ij}\boldsymbol{q}_{ij})\right] \\ &\quad + \frac{r}{2}\sum_{i,j}\left[\parallel \boldsymbol{q}_{ij} - \boldsymbol{x}_i + \boldsymbol{v}_j \parallel^2 + \parallel \boldsymbol{p}_{ij} - u_{ij}\boldsymbol{q}_{ij} \parallel^2\right]. \end{aligned} \tag{5.8}$$

où $r > 0$ est le terme de pénalité, $\parallel \cdot \parallel$ est la norme euclidienne, $\boldsymbol{y}_{ij}$ et $\boldsymbol{z}_{ij}$ sont les multiplicateurs de Lagrange associés aux valeurs auxiliaires et leurs contraintes (5.6),(5.7). L'ensemble des multiplicateurs est noté $\mathbb{Y} = (\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Z}})$.

5.1.2 Optimisation

Nous appliquons la méthode ADMM, introduite au paragraphe 3.4.3, pour la résolution du problème de minimisation avec le Lagrangien augmenté (5.8) par l'algorithme itérative suivant. Partant avec $\mathbb{Q}^0 : (\boldsymbol{\mathcal{Q}}^0, \boldsymbol{\mathcal{P}}^0)$ et $\mathbb{Y}^0 :(\boldsymbol{\mathcal{Y}}^0,\boldsymbol{\mathcal{Z}}^0)$, nous calculons successivement $\mathbb{U}^k : (\boldsymbol{U}^k, \boldsymbol{\mathcal{V}}^k, \boldsymbol{\mathcal{S}}^k)$, $\mathbb{Q}^k : (\boldsymbol{\mathcal{Q}}^k, \boldsymbol{\mathcal{P}}^k)$ et $\mathbb{Y}^k :(\boldsymbol{\mathcal{Y}}^k,\boldsymbol{\mathcal{Z}}^k)$ par la procédure suivante (voir l'algorithme 7) :

$$\mathbb{U}^{k+1} = \arg\min_{\mathbb{U}} \mathcal{L}_r(\mathbb{U}, \mathbb{Q}^k, \mathbb{Y}^k), \tag{5.9}$$

$$\mathbb{Q}^{k+1} = \arg\min_{\mathbb{Q}} \mathcal{L}_r(\mathbb{U}^{k+1}, \mathbb{Q}, \mathbb{Y}^k), \tag{5.10}$$

$$\boldsymbol{y}_{ij}^{k+1} = \boldsymbol{y}_{ij}^k + r(\boldsymbol{q}_{ij}^{k+1} - \boldsymbol{x}_i + \boldsymbol{v}_j^{k+1}), \tag{5.11}$$

$$\boldsymbol{z}_{ij}^{k+1} = \boldsymbol{z}_{ij}^k + r(\boldsymbol{p}_{ij}^{k+1} - u_{ij}^{k+1}\boldsymbol{q}_{ij}^{k+1}). \tag{5.12}$$

5.1.2.1 Solution du sous-problème (5.9) en $\mathbb{U}$

Nous supposons que les variables auxiliaires $\mathbb{Q}^k$ et que les multiplicateurs $\mathbb{Y}^k$ sont fixés. Le problème (5.9) du Lagrangien augmenté (5.8) est découplé selon chaque variable de $\mathbb{U}$ grâce à notre travail préliminaire. Donc l'optimisation est séparée selon chaque

variable :

$$\boldsymbol{\mathcal{V}}^{k+1} = \arg\min_{\boldsymbol{\mathcal{V}}} \sum_{i=1}^{n}\sum_{j=1}^{c} (\boldsymbol{y}_{ij}^k)^\top (\boldsymbol{q}_{ij}^k - \boldsymbol{x}_i + \boldsymbol{v}_j) + \frac{r}{2} \parallel \boldsymbol{q}_{ij}^k - \boldsymbol{x}_i + \boldsymbol{v}_j \parallel^2, \quad (5.13)$$

$$\begin{aligned}\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} I_{\mathcal{U}}(\boldsymbol{U}) + \sum_{i=1}^{n}\sum_{j=1}^{c} (\boldsymbol{z}_{ij}^k)^\top (\boldsymbol{p}_{ij}^k - u_{ij}\boldsymbol{q}_{ij}^k) \\ + \frac{r}{2} \parallel \boldsymbol{p}_{ij}^k - u_{ij}\boldsymbol{q}_{ij}^k \parallel^2,\end{aligned} \quad (5.14)$$

$$\boldsymbol{\mathcal{S}}^{k+1} = \arg\min_{\boldsymbol{\mathcal{S}}} \sum_{i=1}^{n}\sum_{j=1}^{c} (\boldsymbol{p}_{ij}^k)^\top \boldsymbol{S}_j \boldsymbol{p}_{ij}^k + I_{\mathcal{S}_1}(\boldsymbol{\mathcal{S}}). \quad (5.15)$$

Les sous-problèmes (5.13)-(5.15) sont résolus en prenant les conditions d'optimalité à l'exemple de l'optimisation alternée. Cela explique pourquoi les deux méthodes donnent des formulations similaires.

Optimisation de $\boldsymbol{\mathcal{V}}$

Le problème en $\boldsymbol{\mathcal{V}}$ est sans contrainte, il suffit d'annuler le gradient :

$$\begin{aligned}&\frac{\partial \mathcal{L}_r(\mathbb{U}, \mathbb{Q}^k, \mathbb{Y}^k)}{\partial \boldsymbol{v}_j} = 0, \quad \forall j \in [1, c], \\ &\Longrightarrow \sum_{i=1}^{n} \boldsymbol{y}_{ij}^k + r(\boldsymbol{q}_{ij}^k - \boldsymbol{x}_i + \boldsymbol{v}_j) = 0, \\ &\Longrightarrow \boldsymbol{v}_j^{k+1} = \frac{1}{n}\sum_{i=1}^{n} \left(\boldsymbol{x}_i - \boldsymbol{q}_{ij}^k - \frac{1}{r}\boldsymbol{y}_{ij}^k \right).\end{aligned} \quad (5.16)$$

Optimisation de $\boldsymbol{U}$

La partition $\boldsymbol{U}$ est une partition probabiliste. Elle doit respecter les contraintes de $\mathcal{U}$ (5.2). Nous utilisons, ici, la même démarche que décrite usuellement avec l'optimisation alternée [5, 11, 50]. Seule la contrainte $\sum_{j=1}^{c} u_{ij} = 1, \forall i \in [1, n]$ est étudiée. La fonction caractéristique $I_{\mathcal{U}}(\boldsymbol{U})$ est remplacée pour tous les objets i par $\sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{c} u_{ij} - 1 \right)$. Regardons d'abord les variables,

$$\begin{aligned}&\frac{\partial \mathcal{L}_r(\mathbb{U}, \mathbb{Q}^k, \mathbb{Y}^k)}{\partial u_{ij}} = 0, \quad \forall i, j \in [1, n] \times [1, c] \\ &\Longrightarrow \lambda_i - (\boldsymbol{z}_{ij}^k)^\top \boldsymbol{q}_{ij}^k - r(\boldsymbol{q}_{ij}^k)^\top (\boldsymbol{p}_{ij}^k - u_{ij}\boldsymbol{q}_{ij}^k) = 0, \\ &\Longrightarrow u_{ij} = \frac{(\boldsymbol{z}_{ij}^k)^\top \boldsymbol{q}_{ij}^k + r(\boldsymbol{q}_{ij}^k)^\top \boldsymbol{p}_{ij}^k - \lambda_i}{r(\boldsymbol{q}_{ij}^k)^\top \boldsymbol{q}_{ij}^k}.\end{aligned}$$

D'un autre côté, en regardant l'annulation du gradient selon les contraintes,

$$\begin{aligned}
&\frac{\partial \mathcal{L}_r(\mathbb{U}, \mathbb{Q}^k, \mathbb{Y}^k)}{\partial \lambda_i} = 0, \quad \forall i \in [1, n], \\
&\Longrightarrow \sum_{\ell=1}^{c} u_{i\ell} - 1 = 0, \\
&\Longrightarrow \sum_{\ell=1}^{c} \frac{(\boldsymbol{z}_{i\ell}^k)^\top \boldsymbol{q}_{i\ell}^k + r(\boldsymbol{q}_{i\ell}^k)^\top \boldsymbol{p}_{i\ell}^k - \lambda_i}{r(\boldsymbol{q}_{i\ell}^k)^\top \boldsymbol{q}_{i\ell}^k} = 1, \\
&\Longrightarrow \lambda_i = \frac{\sum_{l=1}^{c} \frac{(\boldsymbol{z}_{i\ell}^k)^\top \boldsymbol{q}_{i\ell}^k + r(\boldsymbol{q}_{i\ell}^k)^\top \boldsymbol{p}_{i\ell}^k}{r(\boldsymbol{q}_{i\ell}^k)^\top \boldsymbol{q}_{i\ell}^k} - 1}{\sum_{\ell=1}^{c} \frac{1}{r(\boldsymbol{q}_{i\ell}^k)^\top \boldsymbol{q}_{i\ell}^k}}.
\end{aligned}$$

En posant,

$$\tilde{\boldsymbol{z}}_{ij}^k = \boldsymbol{z}_{ij}^k + r\boldsymbol{p}_{ij}^k, \quad \alpha_i^k = \frac{1}{r} \sum_{j=1}^{c} \frac{1}{\parallel \boldsymbol{q}_{ij}^k \parallel^2}.$$

Ainsi, la mise à jour des degrés d'appartenance est

$$u_{ij}^{k+1} = \frac{1}{r^2 \alpha_i^k \parallel \boldsymbol{q}_{ij}^k \parallel^2} \left[r\alpha_i^k (\boldsymbol{q}_{ij}^k)^\top \tilde{\boldsymbol{z}}_{ij}^k + 1 - \sum_{\ell=1}^{c} \frac{(\boldsymbol{q}_{i\ell}^k)^\top \tilde{\boldsymbol{z}}_{i\ell}^k}{\parallel \boldsymbol{q}_{i\ell}^k \parallel^2} \right]. \tag{5.17}$$

Optimisation de $\boldsymbol{\mathcal{S}}$

Les matrices de distance sont soumises, elles aussi à une contrainte (5.3). Nous utilisons la même démarche que Gustasfon et Kessel [61]. Nous remplaçons la fonction caractéristique $I_{\mathcal{S}_1}(\boldsymbol{\mathcal{S}})$ par $\lambda_j(1 - \det(\boldsymbol{S}_j))$ pour toutes les classes j.

$$\begin{aligned}
&\frac{\partial \mathcal{L}_r(\mathbb{U}, \mathbb{Q}^k, \mathbb{Y}^k)}{\partial \boldsymbol{S}_j} = 0, \quad \forall j \in [1, c], \\
&\Longrightarrow \sum_{i=1}^{n} \boldsymbol{p}_{ij}^k (\boldsymbol{p}_{ij}^k)^\top - \lambda_j \det(\boldsymbol{S}_j) \boldsymbol{S}_j^{-1} = 0,
\end{aligned}$$

Or, en supposant $\det(\boldsymbol{S}_j) = 1$,

$$\Longrightarrow \boldsymbol{S}_j^{-1} = \frac{1}{\lambda_j} \sum_{i=1}^{n} \boldsymbol{p}_{ij}^k (\boldsymbol{p}_{ij}^k)^\top.$$

La matrice $\boldsymbol{\Sigma}_j^k = \sum_{i=1}^{n} \boldsymbol{p}_{ij}^k (\boldsymbol{p}_{ij}^k)^\top$ est la matrice de covariance.
Pour vérifier l'hypothèse, $\det(\boldsymbol{S}_j) = 1$, nous devons prendre $\lambda_j = \det(\boldsymbol{\Sigma}_j^k)^{1/p}$. D'où, la formulation obtenue pour la mise à jour des matrices

$$\boldsymbol{S}_j^{k+1} = \det(\boldsymbol{\Sigma}_j^k)^{1/p} (\boldsymbol{\Sigma}_j^k)^{-1}. \tag{5.18}$$

5.1.2.2 Solution du sous-problème (5.10) en $\mathbb{Q}$

Nous fixons maintenant les variables $\mathbb{U}^{k+1}$ et les multiplicateurs $\mathbb{Y}^k$. Le sous-problème en $\mathbb{Q}$: $(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{P}})$ est un problème d'optimisation sans contrainte grâce au Lagrangien augmenté. Nous remarquons que $F(\mathbb{Q}) = \mathcal{L}_r(\mathbb{U}^{k+1}, \mathbb{Q}, \mathbb{Y}^k)$ est une fonction quadratique dont l'unique solution est obtenue en annulant le gradient :

$$\nabla F(\mathbb{Q}) = 0 \Longleftrightarrow \left\{ \begin{array}{ll} \frac{\partial \mathcal{L}_r(\mathbb{U}^{k+1}, \mathbb{Q}, \mathbb{Y}^k)}{\partial \boldsymbol{q}_{ij}} & = 0 \\ \frac{\partial \mathcal{L}_r(\mathbb{U}^{k+1}, \mathbb{Q}, \mathbb{Y}^k)}{\partial \boldsymbol{p}_{ij}} & = 0 \end{array} \right. , \forall i, j \in [1, n] \times [1, c].$$

Après ce simple calcul, on obtient donc le système linéaire en $(\boldsymbol{q}_{ij}, \boldsymbol{p}_{ij})$:

$$r(1 + (u_{ij}^{k+1})^2)\boldsymbol{q}_{ij} - r u_{ij}^{k+1} \boldsymbol{p}_{ij} = u_{ij}^{k+1} \boldsymbol{z}_{ij}^k - \boldsymbol{y}_{ij}^k + r(\boldsymbol{x}_i - \boldsymbol{v}_j^{k+1}), \tag{5.19}$$

$$-r u_{ij}^{k+1} \boldsymbol{q}_{ij} + (2\boldsymbol{S}_j^{k+1} + r\boldsymbol{I})\boldsymbol{p}_{ij} = -\boldsymbol{z}_{ij}^k. \tag{5.20}$$

A chaque itération, nous résolvons nc systèmes linéaires de taille $2n_d$

$$\boldsymbol{A}_{ij}^k \left[\begin{array}{c} \boldsymbol{q}_{ij} \\ \boldsymbol{p}_{ij} \end{array} \right] = \boldsymbol{b}_{ij}^k, \tag{5.21}$$

$$\boldsymbol{A}_{ij}^k = \left[\begin{array}{cc} r(1 + (u_{ij}^{k+1})^2)\boldsymbol{I} & -r u_{ij}^{k+1} \boldsymbol{I} \\ -r u_{ij}^{k+1} \boldsymbol{I} & 2\boldsymbol{S}_j^{k+1} + r\boldsymbol{I} \end{array} \right], \boldsymbol{b}_{ij}^k = \left[\begin{array}{c} u_{ij}^{k+1} \boldsymbol{z}_{ij}^k - \boldsymbol{y}_{ij}^k + r(\boldsymbol{x}_i - \boldsymbol{v}_j^{k+1}) \\ -\boldsymbol{z}_{ij}^k \end{array} \right],$$

avec $\boldsymbol{I}$ est la matrice identité $n_d \times n_d$.

5.1.3 Algorithme

La contrainte (5.3) concernant les déterminants n'est pas convexe. Dans ce cas particulier, comme expliqué au paragraphe 3.4.4, de manière à garantir la minimisation du Lagrangien augmenté, nous effectuons 5 mises à jour du bloc de relaxation (5.9)-(5.10) avant de mettre à jour les multiplicateurs, $it_a = 5$. Cependant, lorsque la distance euclidienne est utilisée, le problème est dépourvu de non-convexité, la convergence globale est assurée. Nous suggérons d'initialiser l'algorithme avec la distance de Mahalanobis par l'exécution de l'algorithme avec la distance euclidienne. Ainsi, il n'est plus nécessaire d'effectuer une dizaine d'exécutions avec différentes initialisations aléatoires. Une seule exécution suffit désormais.

Les multiplicateurs de Lagrange sont initialisés en résolvant la condition d'optimalité du premier ordre, dérivant le Lagrangien par rapport à $\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{P}}$. Pour rappel, le Lagrangien du problème est

$$\mathcal{L}(\mathbb{U}, \mathbb{Q}, \mathbb{Y}) = J(\mathbb{U}, \mathbb{Q}) + \sum_{i,j} \boldsymbol{y}_{ij}^\top (\boldsymbol{q}_{ij} - \boldsymbol{x}_i + \boldsymbol{v}_j) + \boldsymbol{z}_{ij}^\top (\boldsymbol{p}_{ij} - u_{ij} \boldsymbol{q}_{ij}).$$

En résolvant le système linéaire suivant,

$$\nabla\mathcal{L}(\mathbb{Q}) = 0 \Longleftrightarrow \begin{cases} \frac{\partial\mathcal{L}(\mathbb{U}^0,\mathbb{Q}^0,\mathbb{Y})}{\partial \boldsymbol{q}_{ij}} & = \boldsymbol{y}_{ij} - u_{ij}^0 \boldsymbol{z}_{ij} & = 0 \\ \frac{\partial\mathcal{L}(\mathbb{U}^0,\mathbb{Q}^0,\mathbb{Y})}{\partial \boldsymbol{p}_{ij}} & = 2\boldsymbol{S}_j^0 \boldsymbol{p}_{ij}^0 + \boldsymbol{z}_{ij} & = 0 \end{cases}, \forall i,j \in [1,n] \times [1,c],$$

nous déduisions que $\boldsymbol{z}_{ij}^0 = -2\boldsymbol{S}_j^0\boldsymbol{p}_{ij}^0$ et $\boldsymbol{y}_{ij}^0 = u_{ij}^0\boldsymbol{z}_{ij}^0, \forall i,j$.

L'algorithme 8 résume l'optimisation de FCM-GK par ADMM. Le critère d'arrêt est l'erreur relative sur toutes les variables primaires et duales inférieures à un seuil fixé à tol. Pour t itérations, la complexité de l'optimisation alternée est $O(t(nc^2n_d + ncn_d^2 + cn_d^3))$. Dans notre algorithme, le coût $O(ncn_d^3)$ de mise à jour des variables $\boldsymbol{\mathcal{Q}}^k, \boldsymbol{\mathcal{P}}^k$. La complexité de notre algorithme est $O(tnc(cn_d + n_d^2 + n_d^3))$. Les variables $\boldsymbol{\mathcal{Q}}^k, \boldsymbol{\mathcal{P}}^k$ sont de tailles cnn_d, la complexité mémoire est $O(ncn_d + cn_d^2)$.

Algorithme 8 FCM-GK par ADMM.

Entrée : $\boldsymbol{X}$ les données, c le nombre de classes et r le terme de pénalité.
Sortie : $\boldsymbol{U}^k$,$\boldsymbol{\mathcal{V}}^k$,$\boldsymbol{\mathcal{S}}^k$

$err = 0, k = 0$
$\boldsymbol{U}^0$ initialisation aléatoire (ou ADMM avec la distance euclidienne).
tant que $err > tol$ **faire**
    $k = k + 1$
    pour 1 jusqu'à 5 **faire**
        $\boldsymbol{\mathcal{V}}^k$,$\boldsymbol{\mathcal{S}}^k$ et $\boldsymbol{U}^k$ d'après respectivement (5.16), (5.18) et (5.17).
        $\boldsymbol{\mathcal{Q}}^k, \boldsymbol{\mathcal{P}}^k$ résolvant le système (5.21).
    fin pour
    $\boldsymbol{\mathcal{Y}}^k, \boldsymbol{\mathcal{Z}}^k$ d'après respectivement (5.11) et (5.12).
    $err = \| (\mathbb{U},\mathbb{Q})^k - (\mathbb{U},\mathbb{Q})^{k-1} \| / \| (\mathbb{U},\mathbb{Q})^k \|$
fin tant que

5.1.4 Accélération de Nesterov

Nous souhaitons nous concentrer spécifiquement sur l'optimisation de $\boldsymbol{\mathcal{S}}$. La contrainte sur le déterminant (5.3) est abandonnée pour privilégier une projection sur l'ensemble des matrices symétriques définies positives. L'objectif est de résoudre le problème de minimisation en $\boldsymbol{\mathcal{S}}$ à l'aide de la méthode du gradient proximal accéléré (Nesterov) décrite dans le paragraphe 3.3. Cette méthode ne résout qu'un problème univarié, ce qui implique qu'elle est utilisée à l'intérieur d'une autre méthode d'optimisation pour notre problème multivarié.

La contrainte sur le déterminant a été proposée pour éviter la solution triviale $\boldsymbol{S} = 0$. Nous proposons une approche différente, en exigeant que les matrices de $\boldsymbol{\mathcal{S}}$ soient symétriques définies positives. La longueur de rayon de l'ellipse l_j selon la direction du vecteur propre $\boldsymbol{d}_j$ est l'inverse de la racine de la valeur propre associée μ_j : $l_j = \frac{1}{\sqrt{\mu_j}}$.

Cela signifie que lorsque $\boldsymbol{S}$ est petit, l'ellipse associée est très grande. Nous fixons une valeur maximale à la longueur de l'ellipse $l_{\max}$ dans le but d'éviter la solution triviale.

Définition 5.1.1: Ensemble de définition des matrices

Soit l'ensemble $\mathscr{S}^*$ est défini comme l'ensemble des matrices définies positives avec des valeurs propres dans $I^* = \left[\mu_{min} = \frac{1}{\sqrt{l_{\max}}}, +\infty\right]$.
Nous notons $I_{\mathscr{S}^*}$ la fonction indicatrice associée et $\Pi_{\mathscr{S}^*}$ la projection dans cette ensemble.

Comme la fonction $J(\boldsymbol{\mathcal{S}}, \boldsymbol{P})$ est linéaire en $\boldsymbol{\mathcal{S}}$ son gradient est constant :

$$\nabla J(\boldsymbol{S}_j) = \sum_{i=1}^{n} \boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^{\top}.$$

Avec pour objectif d'appliquer la méthode de Nesterov, nous ajoutons deux termes à la fonction objectif qui devient

$$J_N(\boldsymbol{\mathcal{S}}) = \sum_{j=1}^{c}\sum_{i=1}^{n} \boldsymbol{p}_{ij}\boldsymbol{S}_j\boldsymbol{p}_{ij}^{\top} - \tau_1 \ln\det(\boldsymbol{S}_j) + \tau_2 \parallel \boldsymbol{S}_j \parallel_F^2, \tag{5.22}$$

où $\tau_1 \geq \tau_2 > 0$.

Le premier terme s'inspire du travail de Liu et al [62] qui ont relâché la contrainte du déterminant en le remplaçant par la fonction $-\ln\det(\boldsymbol{S}_j)$, c'est une fonction convexe qui permet de réguler le déterminant. Cependant, ce terme n'est pas strictement convexe et n'est pas stable :

$$\lim_{\boldsymbol{S}_j \to \infty} -\ln\det(\boldsymbol{S}_j) = -\infty.$$

Pour contrôler ce terme, nous ajoutons la norme de Frobinus de la matrice, à l'exemple de Rothman qui utilise la norme 1 [157].

Le problème s'écrit désormais,

$$\min_{\boldsymbol{\mathcal{S}}} J(\boldsymbol{\mathcal{S}}, \boldsymbol{P}) + I_{\mathscr{S}^*}(\boldsymbol{\mathcal{S}}) = \sum_{j=1}^{c}\sum_{i=1}^{n} \boldsymbol{p}_{ij}^{\top}\boldsymbol{S}_j\boldsymbol{p}_{ij} + I_{\mathscr{S}^*}(\boldsymbol{S}_j). \tag{5.23}$$

Nous remarquons que le problème est indépendant selon les classes j, minimiser $\boldsymbol{\mathcal{S}}$ revient à minimiser chacun des $\boldsymbol{S}_j$ séparément.

Afin d'appliquer la méthode de Nesterov, il est nécessaire d'étudier la fonction $\nabla J_N(\boldsymbol{S}_j)$.

Proposition 5.1.1: Constance de Lipschitz

Soit la fonction $\nabla J_N : \boldsymbol{S}_j \longrightarrow \sum_{i=1}^{n} \boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^{\top} - \tau_1 \boldsymbol{S}_j^{-1} + 2\tau_2 \boldsymbol{S}_j$, est une fonction L-lipschitzienne avec

$$L = \frac{\tau_1 n_d}{\mu_{\min}^2} + 2\tau_2 = \tau_1 n_d l_{\max} + 2\tau_2,$$

où n_d est le nombre d'attributs du jeu de données.

Démonstration. Soit $\| \ . \ \|_F$ la norme de Frobinus, démontrons le résultat de la propriété en vérifiant que pour n'importe quelle classe $j \in [1, n]$,

$$\forall \boldsymbol{S}_j, \boldsymbol{Z}_j \in \mathscr{S}^*, \| \nabla J_N(\boldsymbol{S}_j) - \nabla J_N(\boldsymbol{Z}_j) \|_F \leq (\frac{\tau_1 n_d}{\mu_{\min}^2} + 2\tau_2) \| \boldsymbol{S}_j - \boldsymbol{Z}_j \|_F .$$

Calculons la norme de la différence de gradient. Nous utilisons l'inégalité triangulaire et la propriété de la norme d'être sous multiplicative.

$$\begin{aligned}
\| \nabla J_N(\boldsymbol{S}_j) - \nabla J_N(\boldsymbol{Z}_j) \|_F &= \| -\tau_1(\boldsymbol{S}_j^{-1} - \boldsymbol{Z}_j^{-1}) + 2\tau_2(\boldsymbol{S}_j - \boldsymbol{Z}_j) \|_F \\
&= \| -\tau_1 \boldsymbol{S}_j^{-1}(\boldsymbol{I} - \boldsymbol{S}_j\boldsymbol{Z}_j^{-1}) + 2\tau_2(\boldsymbol{S}_j - \boldsymbol{Z}_j) \|_F \\
&= \| -\tau_1 \boldsymbol{S}_j^{-1}(\boldsymbol{Z}_j - \boldsymbol{S}_j)\boldsymbol{Z}_j^{-1} + 2\tau_2(\boldsymbol{S}_j - \boldsymbol{Z}_j) \|_F \\
&= \| \tau_1 \boldsymbol{S}_j^{-1}(\boldsymbol{S}_j - \boldsymbol{Z}_j)\boldsymbol{Z}_j^{-1} + 2\tau_2(\boldsymbol{S}_j - \boldsymbol{Z}_j) \|_F \\
&\leq \tau_1 \| \boldsymbol{S}_j^{-1}(\boldsymbol{S}_j - \boldsymbol{Z}_j)\boldsymbol{Z}_j^{-1} \|_F + 2\tau_2 \| \boldsymbol{S}_j - \boldsymbol{Z}_j \|_F \\
&\leq \tau_1 \| \boldsymbol{S}_j^{-1} \|_F \| \boldsymbol{S}_j - \boldsymbol{Z}_j \|_F \| \boldsymbol{Z}_j^{-1} \|_F + 2\tau_2 \| \boldsymbol{S}_j - \boldsymbol{Z}_j \|_F .
\end{aligned}$$

De plus, nous avons la relation entre la norme de Frobinus et la norme 2 :

$$\| \boldsymbol{S}_j^{-1} \|_F \leq \sqrt{n_d} \| \boldsymbol{S}_j^{-1} \|_2 = \frac{\sqrt{n_d}}{\mu_{\min}}.$$

Nous obtenons ainsi la majoration suivante,

$$\| \nabla J_N(\boldsymbol{S}_j) - \nabla J_N(\boldsymbol{Z}_j) \|_F \leq (\frac{\tau_1 n_d}{\mu_{\min}^2} + 2\tau_2) \| \boldsymbol{S}_j - \boldsymbol{Z}_j \|_F .$$

Ainsi $\nabla J_N(.)$ est bien L-Lipschtizienne avec $L = \frac{\tau_1 n_d}{\mu_{\min}^2} + 2\tau_2 = \tau_1 n_d l_{\max} + 2\tau_2$. D'où $\delta = \frac{1}{\tau_1 n_d l_{\max} + 2\tau_2}$. □

Nous définissons l'opérateur proximal $prox_{I_{\mathscr{S}^*}}$ comme étant la projection sur l'ensemble $\mathscr{S}^*$,$\Pi_{\mathscr{S}^*}$. Nous réalisons cette projection à l'aide d'un seuillage sur la décomposition spectrale dans l'intervalle $[\mu_{min}, +\infty]$. Pour une matrice symétrique $\boldsymbol{Z}$, nous avons la décomposition :

$$\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^{\top},$$

où $\boldsymbol{Q}$ est une matrice orthogonale, dont les colonnes sont des vecteurs propres de $\boldsymbol{Z}$, et où $\boldsymbol{D} = diag(\mu_1, ..., \mu_c)$ est une matrice diagonale dont les coefficients sont les valeurs propres associées.
La projection est réalisée grâce au seuillage des valeurs propres :

$$\Pi_{\mathscr{S}^*}(\boldsymbol{Z}) = \boldsymbol{Q}\boldsymbol{D}^+\boldsymbol{Q}, \tag{5.24}$$

avec $\boldsymbol{D}^+ = max(\boldsymbol{D}, \mu_{min})$. Nous confirmons que le terme projeté est bien symétrique puisque $\boldsymbol{Z}^k$ et $\nabla J_N(\boldsymbol{Z}^k)$ le sont. Ainsi, nous obtenons l'algorithme 9 de l'optimisation des distances par l'application de la méthode du gradient proximal accéléré de Nestrov.

L'algorithme est initialisé en $\boldsymbol{S}^0$ d'après la formule (3.22) : $\boldsymbol{S}^0 = det(\boldsymbol{\Sigma})^{\frac{1}{p}}\boldsymbol{\Sigma}^{-1}$, avec la matrice de variance-covariance $\boldsymbol{\Sigma}$.

Algorithme 9 APG pour l'optimisation des distances.

Itération $k = 0$: $\boldsymbol{Z}^0 = \boldsymbol{S}^0, t_0 = 1$ et $\delta > 0$
Itération $k \geq 1$:
1: $\boldsymbol{S}^{k+1} = \Pi_{\mathscr{S}^*}(\boldsymbol{Z}^k - \delta\nabla J_N(\boldsymbol{Z}^k))$
2: $t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$
3: $\boldsymbol{Z}^k = \boldsymbol{S}^{k+1} + (t_k - 1)(\boldsymbol{S}^{k+1} - \boldsymbol{S}^k)/t^{k+1}$

La décomposition de la matrice est aussi coûteuse que de l'inverser en $O(n_d^3)$, donc la complexité temporelle de la méthode de Nesterov est du même ordre de grandeur que celle de l'AO $O(t(nc^2n_d + ncn_d^2 + cn_d^3))$, de même pour sa complexité spatiale $O((nn_d + nc + cn_d^2)$.

5.2 Expérimentations numériques

5.2.1 Méthodologie

À travers ces expérimentations numériques, notre objectif est de comparer les performances des différentes méthodes d'optimisation :

— Pour la résolution du problème FCM-GK, nous confrontons les algorithmes 8 et 3 qui utilisent respectivement les méthodes d'optimisation ADMM et AO.
— Pour l'optimisation spécifique des distances de Mahalanobis, nous comparons l'approche basée sur la contrainte du déterminant de l'AO (algorithme 3) et l'approche par projection de la méthode du gradient proximal accéléré de Nesterov (APG) (algorithme 9).

Nous analysons la qualité du partitionnement à l'aide de plusieurs indices : l'indice ARI (éq. 2.37), l'indice $XBMW$ (éq. 4.10), le Partition Entropy PE (éq. 2.45) et le Fuzzy Silhouette FS (éq. 2.46). L'ARI permet une évaluation externe avec le partitionnement proposé par l'expert, sa valeur doit être maximisée dans son intervalle

[-1,1]. Pour l'évaluation externe, l'indice Partition Coefficient doit être minimisé dans son intervalle [0,ln(c)], $XBMW$ doit être minimisé vers 0 et FS maximisé dans [-1,1].

Nous examinons également la rapidité d'exécution en mesurant le temps CPU et le nombre d'itérations nécessaires pour atteindre la convergence.

Spécifiquement pour ADMM, nous devons d'abord vérifier que le problème d'optimisation reformulé (5.5)-(5.7) se rapproche suffisamment du problème d'origine de FCM-GK (5.1)-(5.3). Pour ce faire, nous contrôlons que les résidus $\forall i, j \in [1, n] \times [1, c], \quad \tau_{ij} = \Big(\| \boldsymbol{q}_{ij} - \boldsymbol{x}_i + \boldsymbol{v}_j \|, \| \boldsymbol{p}_{ij} - u_{ij}\boldsymbol{q}_{ij} \|\Big)$ sont suffisamment petits :

$$\tau = \max_{i,j \in [1,n]\times[1,c]} \tau_{ij} \leq 10^{-4},$$

pour une tolérance $tol = 10^{-4}$ dans l'algorithme 8. Les tolérances des autres algorithmes sont fixées à 10^{-3} conformément à la version originale de Bezdek [5].

Puisque la convergence globale n'est pas assurée, comme souligné dans le chapitre précédent, il est nécessaire de réaliser plusieurs initialisations aléatoires et de conserver les variables qui minimisent le mieux la fonction objectif. Néanmoins, avec la méthode ADMM restreinte au cas euclidien (sans les matrices $\boldsymbol{\mathcal{S}}$), le problème est convexe, ce qui assure la convergence vers un minimum local. Ainsi, nous n'avons plus besoin de réaliser plusieurs initialisations aléatoires, une seule suffit.

Bien qu'ADMM offre cet avantage, il est nécessaire de définir la valeur du terme de pénalité r. C'est le principal inconvénient de cette méthode, bien qu'il soit relativisé par l'existence d'un terme garantissant la convergence. En général, le terme de pénalité retenu est celui qui permet d'atteindre la convergence la plus rapide en termes d'itérations [158]. Ainsi, pour ADMM avec la distance euclidienne, nous avons obtenu $r = 2,5$ pour n'importe quel jeu de données, en fixant le nombre d'itérations maximal à 50.

Les calculs numériques ont été effectués à l'aide du logiciel MATLAB R2020a sur un ordinateur équipé d'un processeur Intel Core i5 de 10ème génération, 16 Go de RAM sous Linux.

5.2.2 Jeux de données

Nous comparons les méthodes sur 17 jeux couramment employés couramment dans la littérature avec FCM-GK, huit données synthétiques, A1, A3, Asymétrique, DIM32, DIM64 et Skewed, présentés dans l'annexe A.3 et neuf sont issues de la bibliothèque de l'UCI [1] dont les origines et caractéristiques sont détaillées en annexe A.4. Une procédure de normalisation a été réalisée sur les jeux de données pour que chaque attribut

1. https://archive.ics.uci.edu/

ait une importance égale et pour simplifier l'application de la méthode Nesterov. En effet, les attributs des données sont désormais compris entre [-1, 1] donc la longueur maximale est la longueur de la diagonale d'un carré de longueur 2 soit $l_{max} = 2\sqrt{2}$, en deux dimensions. Nous pouvons généralisé par la formule suivante $l_{max} = 2\sqrt{nd}$. Le détail du pré-traitement est présenté en annexe A.1.2.

Dans la présentation des résultats, par souci de compacité des tableaux, nous utilisons la notation e lorsque cela est nécessaire, comme suit : $10^n = e + n, 10^{-n} = e - n$.

5.2.3 ADMM vs AO

Les pénalités optimales r^* pour chaque jeu de données obtenues après une étude approfondie sont presentés dans le tableau 5.1.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
r^*	10^6	10^5	10^5	320	10^4	10^5	10^5	250	400

(a) Jeux de données UCI.

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
r^*	40	40	10	10^3	500	10^3	10^3	10

(b) Jeux de données synthétiques.

Tableau 5.1 – Pénalité optimale pour ADMM appliquée à FCM-GK.

Dans une procédure classique, le terme de pénalité optimale est celui qui obtient la convergence la plus rapide en terme de nombre d'itérations. Néanmoins, l'initialisation dans notre étude est particulière puisqu'elle recourt à la partition obtenue avec le modèle FCM (distance euclidienne). Dans certains cas, comme pour IRIS, l'évolution de la convergence en fonction du terme de pénalité n'est pas monotone, voir la figure 5.2.1. La convergence avec r très grand est rapide car la partition n'évolue plus. Les ellipses prendront forme, et à la deuxième itération l'algorithme s'arrêtera. Il est préférable de prendre la pénalité qui donnent la convergence la plus rapide mais qui permet également l'évolution des variables.

Tous les paramétrages des algorithmes sont fixés. Le tableau 5.2 présente le résidu maximal τ de chaque jeu de données. Nous vérifions que la convergence d'ADMM correspond à la résolution du problème original puisque le résidu maximal vaut 10^{-6} pour les jeux de données de l'UCI et 10^{-5} synthétiques.

Le tableau 5.3 présente la qualité du partitionnement évaluée par l'ARI. Dans la majorité des cas, ADMM obtient de meilleurs résultats. L'amélioration est particulièrement visible pour les jeux de données de l'UCI. Pour mieux comprendre les conditions dans lesquelles ADMM surpasse l'AO, nous examinons les données synthétiques. Plus précisément, les jeux de données A et S se caractérisent par un nombre élevé de classes

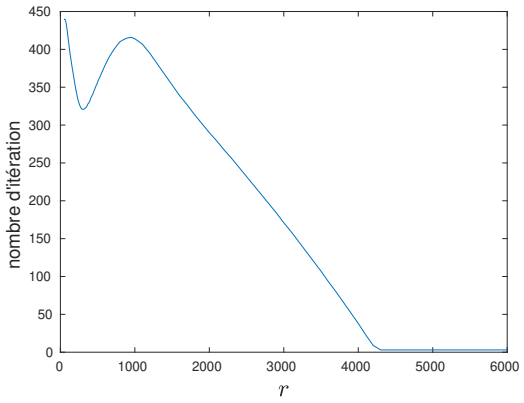


FIGURE 5.2.1 – Nombre d'itérations en fonction de la pénalité r pour Iris.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
ADMM	1e-10	6e-9	8e-9	1e-6	1e-9	4e-10	6e-10	1e-6	1e-6

(a) Jeux de données UCI.

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
ADMM	5e-6	6e-6	1e-5	4e-8	3e-8	8e-9	7e-8	2e-5

(b) Jeux de données synthétiques.

TABLEAU 5.2 – Vérification de la convergence (τ).

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**0.54**	**0.69**	0.58	0.74	0.32	0.72	0.41	0.41	0.33
ADMM	0.34	0.32	**0.62**	**0.92**	**0.33**	**0.73**	**0.76**	**0.76**	**0.81**

(a) Jeux de données UCI.

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	**0.90**	**0.90**	0.96	0.51	0.20	**0.99**	**0.65**	0.99
ADMM	0.28	0.14	**0.97**	**0.52**	**0.55**	0.30	0.27	0.99

(b) Jeux de données synthétiques.

TABLEAU 5.3 – Comparaison d'AO et d'ADMM par ARI.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**9.9e-4**	**2.6e-7**	**2.8e-3**	**5.2e-2**	**9.0e-3**	**8.0e-3**	**8.9e-4**	6.1e+2	8.0e-1
ADMM	7.1e-3	1.6e-4	1.1e-2	6.2e-2	6.2e+2	1.0e-2	1.7e-3	**7.6e-2**	**1.4e-1**

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	**1.4e-1**	**6.0e-2**	7.4e-2	**3.3e-5**	**2.0e-3**	**3.1e-2**	**4.6e-1**	4.6e-2
ADMM	6.4e+3	4.4e+4	**7.4e-2**	1.5e+0	4.6e-2	1.5e+4	1.1e+3	**4.3e-2**

TABLEAU 5.4 – Comparaison d'AO et d'ADMM par $XBMW$.

mais un faible nombre d'attributs ($n_d = 2$). Dans ces cas, l'AO parvient à trouver le bon nombre de classes et leur position, tandis qu'ADMM propose un partitionnement à deux classes. Toutefois, lorsque le nombre d'attributs augmente, ADMM devient plus performant tandis que l'AO montre des résultats moins satisfaisants. Cela est illustré par les jeux de données DIM32 et DIM64, où le nombre d'attributs passe de 32 à 64 pour un nombre de classes de 16.

L'analyse des résultats des indices internes est souvent plus délicate, car elle examine le regroupement de l'intérieur avec un point de vue spécifique. Les indices $XBMW$ et PE donnent globalement les mêmes résultats, comme présenté dans les tableaux 5.4 et 5.5. Selon ces indices, l'optimisation alternée (AO) est la méthode qui offre le meilleur regroupement dans la majorité des cas.

L'indice FS, Fuzzy Silhouette, offre une autre perspective sur la qualité des partitions, comme le reflète le tableau 5.6. Les résultats obtenus par FS sont généralement cohérents avec ceux de l'ARI, ces deux indices partageant le même intervalle de définition [-1, 1]. De plus, les valeurs dans cet intervalle offrent un plus d'informations sur la qualité de la partition : de mauvaise, moyenne à haute qualité pour -1, 0 et 1.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**0.32**	**0.42**	**0.40**	**0.67**	**1.21**	**0.61**	0.93	1.96	1.54
ADMM	0.69	2.51	0.46	0.66	1.58	0.82	**0.68**	**1.82**	**1.43**

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	**1.51**	**2.00**	0.59	**2.72**	**3.56**	**0.88**	**1.64**	0.50
ADMM	4.31	5.64	**0.58**	4.00	4.00	3.90	3.89	0.50

Tableau 5.5 – Comparaison d'AO et d'ADMM par PE.

Nous constatons globalement qu'ADMM est plus performante selon l'indice FS qu'AO. En particulier, pour les jeux de données réels, notre méthode d'optimisation surpasse l'AO sept fois sur neuf. Cela souligne l'efficacité d'ADMM dans un contexte plus proche de la réalité.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
OA	0.30	**0.37**	0.46	0.52	**0.14**	0.47	0.24	0.35	0.31
ADMM	**0.41**	0.06	**0.61**	**0.57**	0.05	**0.49**	**0.43**	**0.39**	**0.37**

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
OA	**0.62**	0.63	0.69	0.57	**0.64**	**0.75**	**0.56**	0.59
ADMM	0.46	**0.80**	**0.69**	**0.61**	0.63	0.66	0.16	0.59

Tableau 5.6 – Comparaison d'AO et d'ADMM par FS.

L'analyse à la fois interne et externe des partitions nous offre une vision plus complète et nuancée de la qualité des résultats. En conclusion, l'optimisation réalisée avec notre méthode se présente comme une alternative solide, se révélant souvent meilleure, notamment pour les jeux de données réelles.

Par ailleurs, pour information, le tableau 5.7 présente les temps d'exécution CPU pour chaque jeu de données et chaque méthode. Ces temps ne sont pas toujours significatifs, car la convergence vers un minimum local donnant de mauvais regroupements est souvent plus rapide (Iris, IJL, WDBC, Wifi, Wine, A et S). Il est intéressant de noter que dans les cas où les deux méthodes convergent vers le même partitionnement, Skewed et Asymmetric, la méthode d'optimisation alternée est généralement plus rapide. Cette différence s'explique notamment par sa moindre complexité spatiale.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**3.0e-2**	**1.6e+1**	3.0e-2	**4.0e-2**	**9.2e-1**	4.0e-2	**1.0e-1**	2.3e+0	**2.5e-1**
ADMM	4.0e-2	3.5e+1	**5.0e-2**	7.6e-1	1.0e+0	4.0e-2	7.0e-1	4.9e+1	8.2e+0

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	6.8e+0	1.8e+2	**1.0e-1**	**4.4e+0**	**2.2e+1**	3.9e+0	7.0e+0	1.2e-1
ADMM	**3.4e+0**	**5.3e+0**	1.2e+0	1.0e+1	5.5e+1	**1.1e+0**	**1.1e+0**	6.8e+0

TABLEAU 5.7 – Temps CPU de l'exécution d'AO et d'ADMM.

5.2.4 AO-APG vs AO

Nous conservons toujours l'initialisation avec la méthode ADMM appliquée à FCM (euclidien), ainsi nous réalisons une seule exécution. Dans le but d'appliquer la méthode de Nesterov, il est nécessaire de fixer les hyperparamètres τ_1 et τ_2.

La fonction objectif de FCM-GK est contrôlée par la partie de τ_1 qui est elle même régulée par la partie de τ_2. Or la fonction objectif et la partie de τ_2 favorisent la solution triviale $\boldsymbol{S}_j = 0$, ce qui grâce à la projection revient à $\boldsymbol{S}_j = \mu_{\min}\boldsymbol{I}$. Donc pour avoir une convergence intéressante de l'algorithme, nous devons avoir $\tau_1 > \tau_2$. Rothman suggère de prendre $\tau_2 = \tau_1 \times 10^{-3}$ [157]. Sur nos jeux de données, il suffit de prendre $\tau_2 = \tau_1 \times 10^{-2}$. Cela permet de contrôler τ_1 comme prévu avec τ_2.

De plus, pour éviter systématiquement la solution triviale, la valeur de τ_1 ne doit pas être trop faible par rapport à la fonction objectif. Tandis qu'un τ_1 trop grand ne favorise pas non plus la convergence de l'algorithme puisque $\| \boldsymbol{S}_j \|_F \longrightarrow +\infty \Longrightarrow J_N \longrightarrow -\infty$. Rothman a fixé $\tau_1 = 10^{-4}$ dans son étude, mais cette valeur est trop petite pour notre fonction objectif. Pour assurer une convergence, il faut prendre $\tau_1 = 10^{-1}$. Cette adaptation de τ_1 permet de garantir une convergence efficace de l'algorithme tout en maintenant le contrôle souhaité sur τ_2 pour ne pas favoriser les $\boldsymbol{S}_j = 0$.

La méthode de Nesterov APG est utilisée exclusivement pour l'optimisation de $\boldsymbol{\mathcal{S}}$, le reste de l'algorithme est celui de l'optimisation alternée. Par conséquent, dans la suite, nous notons cette méthode AO-APG.

Le tableau 5.8 renseigne sur le nombre d'itération nécessaire à la convergence d'AO-APG et la moyenne déterminant $moy(\boldsymbol{\mathcal{S}}) = \frac{1}{c}\sum_{j=1}^{c} \boldsymbol{S}_j$. Dans les cas d'Glass, Asymmetric, S1 et S3, le volume moyen des classes est proche de ceux obtenus par AO et ADMM pour lequelles le volume est contraint à 1, (5.3) contrainte du problème FCM-GK.

En analysant les valeurs de l'ARI présentées dans tableau 5.9, nous remarquons deux tendances différents selon l'origine des données. Pour les jeux de données UCI, la méthode de Nesterov améliore généralement l'adéquation à la partition de référence.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
It	16	83	7	9	2	10	9	2	9
$moy(\boldsymbol{\mathcal{S}})$	3.9e+2	1.4e+18	2.2e+1	7.2e-2	1.9e-6	2.4e+2	1.6e+6	2.9e-3	2.6e-6

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
It	37	22	2	2	2	32	24	23
$moy(\boldsymbol{\mathcal{S}})$	5.0e+14	1.4e+15	3.5e-1	2.2e+14	4.4e+34	3.5e-1	3.5e-1	3.5e-1

Tableau 5.8 – Convergence d'AO-APG.

En revanche, pour les données synthétiques, les performances sur les jeux de données synthétiques sont moins bonnes que l'AO.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**0.54**	**0.69**	0.58	0.74	0.32	**0.72**	0.41	0.41	0.33
AO-APG	0.37	0.36	**0.60**	**0.85**	**0.34**	0.70	**0.75**	**0.83**	**0.82**

(a) Jeux de données UCI.

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	**0.90**	**0.90**	**0.96**	**0.51**	0.20	**0.99**	0.65	**0.99**
AO-APG	0.82	0.74	0.92	0.27	**0.38**	0.89	**0.70**	0.65

(b) Jeux de données synthétiques.

Tableau 5.9 – Comparaison d'AO et d'AO-APG par ARI.

Les tableaux 5.10-5.11 présentent les valeurs des indices $XBMW$ et PE. L'analyse de la comparaison entre l'AO et l'AO-APG est similaire à celle de l'AO et de l'ADMM. En effet, de nouveau, dans la majorité des cas, l'optimisation alternée obtient un regroupement selon ces deux indices.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**9.9e-4**	**2.6e-7**	**2.8e-3**	5.2e-2	**9.0e-3**	**8.0e-3**	**8.9e-4**	6.1e+2	8.0e-1
AO-APG	2.7e-2	5.3e-1	3.4e-2	**3.7e-2**	7.4e+2	8.8e-2	4.9e-2	**1.4e-1**	**1.8e-01**

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	**1.4e-1**	**6.0e-2**	7.4e-2	**3.3e-5**	**2.0e-3**	**3.1e-2**	**4.6e-1**	**4.6e-2**
AO-APG	2.2e-1	8.7e-1	**5.8e-2**	2.2e+5	1.1e+6	2.43e-1	5.5e-2	1.4e-1

Tableau 5.10 – Comparaison d'AO et d'AO-APG par $XBMW$.

Du point de vue de la Fuzzy Silhouette, la méthode avec projection, AO-APG, permet

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**0.32**	**0.42**	**0.40**	**0.67**	**1.21**	**0.61**	0.93	1.96	1.54
AO-APG	0.55	1.38	0.48	0.70	1.58	0.78	**0.70**	**1.42**	**1.21**

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	**1.51**	**2.00**	**0.59**	**2.72**	**3.56**	**0.88**	**1.64**	**0.50**
AO-APG	1.71	2.27	0.61	4	4	1.10	1.60	1.13

Tableau 5.11 – Comparaison d'AO et d'AO-APG par PE.

d'obtenir un meilleur regroupement dans la majorité des cas, comme le montre le tableau 5.12. Cette performance est significative.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
OA	0.30	**0.37**	0.46	0.52	**0.14**	0.47	0.24	0.35	0.31
AO-APG	**0.40**	0.33	**0.61**	**0.57**	-0.02	**0.49**	**0.43**	**0.38**	**0.35**

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
OA	**0.62**	**0.63**	0.69	**0.57**	**0.64**	**0.75**	0.56	**0.59**
AO-APG	0.60	0.57	0.69	0.45	0.49	0.72	**0.59**	0.54

Tableau 5.12 – Comparaison d'AO et d'AO-APG par FS.

De plus, nous observons la grande différence qui peut se manifester entre les différents indices. Cela souligne l'importance de choisir judicieusement l'indice approprié en fonction des objectifs spécifiques de l'analyse du partitionnement.

Enfin, le temps CPU est donné par le tableau 5.13. La comparaison entre les deux méthodes est possible, en particulier pour les données de l'UCI. En effet, AO-APG est souvent plus rapide et sa convergence est aussi souvent meilleure selon ARI et FS. Nous pouvons donc déduire ici l'intérêt de l'accélération adaptée à la méthode de Nesterov, car elle s'avère être une méthode efficiente.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
AO	**3.0e-2**	**1.6e+1**	3.0e-2	4.0e-2	9.2e-1	**4.0e-2**	1.0e-1	2.3e+0	**2.5e-1**
AO-APG	1.5e-1	2.3e+1	**2.0e-2**	1.1e-1	**5.2e-1**	**2.0e-2**	**5.0e-2**	**2.6e-1**	3.1e-1

	A1	A3	Asymmetric	DIM32	DIM64	S1	S3	Skewed
AO	**6.8e+0**	1.8e+2	1.0e-1	4.4e+0	2.2e+1	**3.9e+0**	7.0e+0	1.2e-1
AO-APG	2.4e+1	**7.7e+1**	**6.0e-2**	**6.1e-1**	**1.3e+0**	9.5e+0	**6.6e+0**	**5.3e-1**

Tableau 5.13 – Temps CPU de l'exécution d'AO et d'AO-APG.

5.3 Conclusion

Dans ce chapitre, nous avons étudié deux nouvelles approches pour optimiser le problème FCM avec la distance de Mahalanobis, FCM-GK (2.31-2.32). L'objectif était de compenser les lacunes de la méthode optimisation alternée pour trouver un meilleur partitionnement des données.

La première approche consiste à optimiser le problème par la méthode ADMM. En ajoutant des nouvelles variables, nous avons décomposé le problème en sous-problème plus facile à résoudre. Ainsi cette reformulation rend le problème mieux séparable.

La deuxième approche consiste à relâcher la contrainte sur le volume des classes (2.32) en utilisant une projection sur un ensemble introduit dans la définition 5.1.1. Cette contrainte est non convexe, le défi est de pouvoir relâcher la contrainte tout en évitant la solution triviale $\boldsymbol{S}_j = 0$, que cette contrainte empêchait d'atteindre.

Nous avons pu réaliser des expériences numériques et les performances de ces méthodes d'optimisation sont intéressantes. Elles offrent de nouvelles possibilités. Notamment la propriété de convergence globale d'ADMM dans le cas où la distance est euclidienne (FCM), nous permet d'initialiser l'algorithme qu'une seule fois. La méthode ADMM se révèle particulièrement efficace lorsque le nombre d'attributs est suffisamment élevé par rapport au nombre de classes. Ces performances ont été présentées à deux conférences : EGC [2] [159] et OLA [3] [160].

Nous nous sommes confrontés à la difficulté d'analyser les méthodes d'apprentissage non supervisé. Il est nécessaire d'évaluer le regroupement à l'aide de plusieurs indices de validité. Ainsi d'après $XBMW$ et PE, la méthode d'optimisation alternée obtient en général un meilleur partitionnement. D'un autre côté selon l'indice FS, nos deux propositions améliorent dans la plupart des cas le partitionnement. Ce que confirme l'indice ARI pour les données réelles (UCI). De plus, d'après l'évaluation externe, nous observons une tendance générale : plus le nombre d'attributs est important par rapport au nombre de classes, plus l'optimisation par ADMM est intéressante.

Ces méthodes offrent une alternative crédible. Le calibrage de leurs hyperparamètres (r, τ_1, τ_2) est un véritable défi, leur choix est primordiale pour permettre une bonne convergence. Le tableau 5.14 reprend les caractéristiques, les avantages et inconvénients des différentes méthodes. Une étude de la complexité a été réalisée, ADMM est plus coûteux en temps et en mémoire. La méthode de Nesterov présente la même complexité spatiale et temporelle que l'optimisation alternée puisqu'elle intervient seulement dans

2. https://www.egc.asso.fr/
3. https://ola2023.sciencesconf.org/

la mise à jour des matrices $\boldsymbol{\mathcal{S}}$. Cependant, nous constatons que la méthode de Nesterov a souvent une convergence plus rapide due à son principe d'accélération par l'inertie.

En conclusion de ce chapitre, l'idée de notre approche est intéressante théoriquement et les résultats numériques montrent l'intérêt d'utiliser des méthodes d'optimisation plus robustes malgré la complexité de leur application.

Il serait intéressant d'améliorer la fixation des pénalités de manière plus automatique en fonction de chaque jeu de données. Pour confirmer ces bons résultats, nous souhaitons appliquer notre méthode à un jeu de données de biologie, où les objets à classer ont un grand nombre d'attributs. Combiner la méthode de Nesterov et ADMM permettrait de rejoindre nos deux approches. Enfin, notre étude ouvre des perspectives d'utilisation des méthodes d'optimisation pour résoudre d'autres problèmes de classification non supervisées notamment ECM.

Méthodes	Optimisation alternée	Directions alternées	Gradient proximal accéléré
Nature des fonctions	$\min_{\boldsymbol{x}\in R^n} f(\boldsymbol{x})$ f convexe et différentiable.	$\min_{\boldsymbol{x}\in R^n} f(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x})$ f convexe avec $f_1$différentiable mais pas nécessairement f_2.	
Outil de résolution	Lagrangien	Lagrangien augmenté	Opérateur proximal
Algorithme	2 AO	7 ADMM	5 APG
Modularité (Variables)	+	+	-
↪ Sensibilité (Décomposition)	-	-	
↪ Coordination (Variables)	-	+	
Séparabilité (Fonction)	-	+	+
Convergence (Convexe)	+	+	+
Convergence (Non-convexe)	-	+	-
Rapidité			+
Mise en oeuvre	+	-	+
↪ Parallélisme	+	+	-
↪ Sensibilité (Hyperparamètres)		-	+
Complexité temporelle	$O(tc(ncn_d + nn_d^2 + n_d^3))$	$O(tnc(cn_d + n_d^2 + n_d^3))$	$O(tc(ncn_d + nn_d^2 + n_d^3))$
Complexité mémoire	$O(nn_d + nc + cn_d^2)$	$O(ncn_d + cn_d^2)$	$O(nn_d + nc + cn_d^2)$

TABLEAU 5.14 – Comparaison théorique des méthodes.

Chapitre 6

Adaptation d'ECM pour la détection des zones d'imprécisions induites par la distance de Mahalanobis

Contents

Le formalisme de *k-means* repose sur un concept simple : chaque classe est représentée par un objet "type" appelé centroïde ou centre de gravité. Chaque objet est associé à la classe dont le centroïde est le plus proche en terme de distance. Dans sa version évidentielle, les sous-ensembles sont aussi caractérisés par leur centre de gravité. Ces centroïdes sont définis en fonction de ceux des classes qui composent ces sous-ensembles. Masson et al. [11] utilisent un calcul barycentrique adapté à la distance euclidienne.

Cependant cette approche peut ne pas être adaptée lorsque des distances adaptatives sont employées. Elle peut conduire à une mauvaise localisation de la zone d'imprécision modélisée par les sous-ensembles. Or, l'intérêt d'un modèle évidentiel est de pouvoir caractériser les zones d'imprécision, pour lesquelles un expert peut apporter de nouvelles informations.

Dans ce chapitre, nous abordons en premier lieu la problématique, illustrant les limitations de l'approche actuelle (section 6.1). Ensuite, nous introduisons notre proposition (section 6.2). Enfin, nous analysons les performances obtenues par notre approche (section 6.3).

6.1 Problématique

Dans le paragraphe 2.3.2 qui introduit le concept d'ECM, nous avons présenté la formule de barycentre proposée (2.29) par Masson et al. [11] pour définir les centroïdes $(\boldsymbol{v}_j)$ de tous les sous-ensembles $(\mathcal{A}_j \neq \emptyset, \in 2^\Omega)$ en fonction de ceux des classes $(\boldsymbol{v}_\ell, \forall \ell \in [1, c])$. De même, dans CECM, Antoine [50] utilise une formulation barycentrique similaire (éq. 6.1).

Pour rappel, voici ces formules :

$$\boldsymbol{v}_j = \overline{\boldsymbol{v}}_j \triangleq \frac{1}{|\mathcal{A}_j|} \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{v}_\ell = \frac{1}{|\mathcal{A}_j|} \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{v}_\ell,$$

$$\boldsymbol{S}_j = \overline{\boldsymbol{S}}_j \triangleq \frac{1}{|\mathcal{A}_j|} \sum_{\omega_l \in \mathcal{A}_j} \boldsymbol{S}_\ell = \frac{1}{|\mathcal{A}_j|} \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{S}_\ell. \tag{6.1}$$

Les formules barycentriques étendues à la distance de Mahalanobis ne permettent pas toujours une bonne représentation de la zone d'imprécision. Nous prenons l'exemple 6.1.1 pour illustrer la problématique.

Exemple 6.1.1: Détection de la zone d'imprécision

Soient deux classes ω_1, ω_2 définies par leur centroïde $\boldsymbol{v}_1 = \begin{pmatrix} 5 \\ 1 \end{pmatrix}, \boldsymbol{v}_2 = \begin{pmatrix} 10 \\ 6 \end{pmatrix}$ et leur ellipse de longueurs d'axe $a_1 = a_2 = 7$ et $b_1 = b_2 = 2$ et d'angle $\theta_1 = 0°, \theta_2 = 90°$, par construction avec les formules barycentriques, le sous-ensemble $\omega_{1\cup 2}$ a pour centroïde et ellipse,

$$\boldsymbol{v}_{1\cup 2} = \begin{pmatrix} 7.5 \\ 3.5 \end{pmatrix}, a_{1\cup 2} = b_{1\cup 2} = 4.5, \theta_{1\cup 2} = 0°.$$

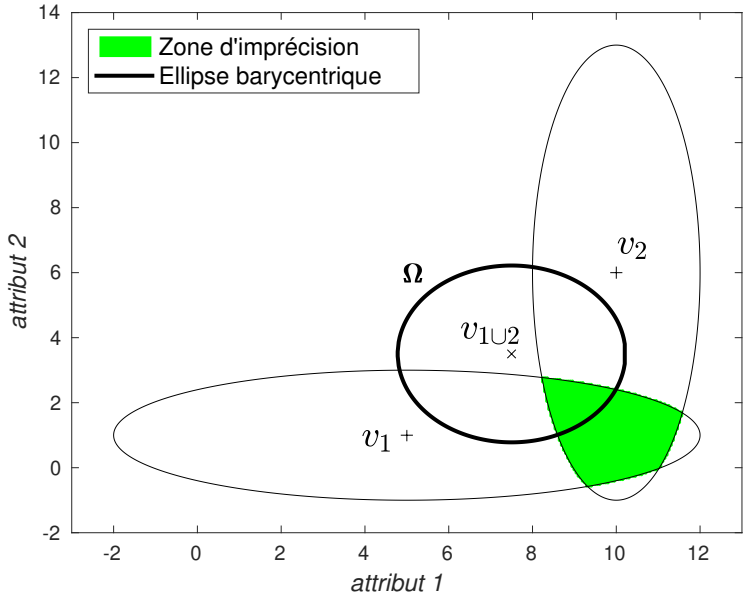


FIGURE 6.1.1 – Zone d'imprécision.

D'après la figure 6.1.1, nous observons que le centroïde de sous-ensemble $\boldsymbol{v}_{1\cup2}$ n'est pas dans la zone de recouvrement des deux ellipses où se situe la véritable zone d'imprécision, en vert. De même, la forme de son ellipse n'est visiblement pas judicieuse. Cette étude est nécessaire pour trouver une formulation appropriée du centroïde et des matrices de variance-covariance pour chaque sous-ensemble.

Ce problème a déjà été exploré par Davis [161] dans le contexte de la combinaison d'ellipses d'erreurs. Dans cette section, nous détaillons les formules à appliquer spécifiquement dans le cadre des partitions crédales avec la distance de Mahalanobis.

6.2 Formulation

Nous présentons deux approches différentes aboutissant aux mêmes formules. Dans un premier temps, nous étudions le problème d'un point de vue statistique puis géométrique.

6.2.1 Analyse statistique

A chaque itération d'ECM, les centroïdes et les matrices de covariance des classes sont obtenus par l'optimisation alternée. Le défi réside dans l'élaboration d'une formule permettant de relier le centroïde et la matrice de covariance d'un sous-ensemble $\mathcal{A}_j$, pour lequel $|\mathcal{A}_j| > 1$, aux centroïdes et aux matrices des singletons inclus dans $\mathcal{A}_j$.

Soit $\boldsymbol{\mu}_j \in \mathbb{R}^p$ et $\boldsymbol{\Sigma}j \in \mathbb{R}^{p\times p}$ la moyenne réelle et la matrice de covariance réelle d'un sous-ensemble $\mathcal{A}_j$ tel que $|\mathcal{A}_j| > 1$. Les valeurs attendues E pour $\boldsymbol{\mu}_j$ et $\boldsymbol{\Sigma}_j$ sont :

$$\begin{aligned} \boldsymbol{\mu}_j &= E[\boldsymbol{Y}_j], \\ \boldsymbol{\Sigma}_j &= Var[\boldsymbol{Y}_j] = E[(\boldsymbol{Y}_j - E[\boldsymbol{Y}_j])(\boldsymbol{Y}_j - E[\boldsymbol{Y}_j])^T], \end{aligned}$$

où $\boldsymbol{Y}_j$ représente une variable aléatoire associée à l'échantillon $\{\boldsymbol{v}_\ell\}, \forall \omega_\ell \in \mathcal{A}_j$. Nous supposons qu'elle suit la distribution normale $\boldsymbol{Y}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$.

Proposition 6.2.1: Définition des estimateurs

Soit $\overline{\boldsymbol{v}}_j$ un estimateur de $\boldsymbol{\mu}_j$ sans biais :

$$\overline{\boldsymbol{v}}_j = \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell \boldsymbol{v}_\ell, \tag{6.2}$$

avec $\boldsymbol{W}_\ell \in \mathbb{R}^{p\times p}$ une matrice de poids, telle que

$$\sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell = \boldsymbol{I}, \tag{6.3}$$

avec $\boldsymbol{I}$ la matrice identité.
Et soit $\overline{\boldsymbol{Z}}_j$ un estimateur de $\boldsymbol{\Sigma}_j$ associé à $\overline{\boldsymbol{v}}_j$:

$$\overline{\boldsymbol{Z}}_j = \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T. \tag{6.4}$$

Démonstration. Soit l'estimateur $\overline{\boldsymbol{v}}_j$ de $\boldsymbol{\mu}_j$,

$$\begin{aligned} E(\overline{\boldsymbol{v}}_j) &= E(\sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell \boldsymbol{v}_\ell) = \sum_{\omega_\ell \in \mathcal{A}_j} E(\boldsymbol{W}_\ell \boldsymbol{v}_\ell), \\ &= \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell E(\boldsymbol{v}_\ell) = \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell \boldsymbol{\mu}_j. \end{aligned}$$

Avec la définition 6.3 pour les matrices des poids, nous vérifions que l'estimateur est sans biais,

$$E(\overline{\boldsymbol{v}}_j) = \boldsymbol{\mu}_j.$$

Comme les centroïdes des singletons sont indépendants,

$$Var[\overline{\boldsymbol{v}}_j] = Var\left[\textstyle\sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell \boldsymbol{v}_\ell\right] = \textstyle\sum_{\omega_\ell \in \mathcal{A}_j} Var\left[\boldsymbol{W}_\ell \boldsymbol{v}_\ell\right].$$

Avec $Var\left[\boldsymbol{v}_\ell\right] = \boldsymbol{Z}_\ell$, nous obtenons l'estimateur de $\boldsymbol{\Sigma}_j$ associé à $\overline{\boldsymbol{v}}_j$,

$$\overline{\boldsymbol{Z}}_j = Var[\overline{\boldsymbol{v}}_j] = \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T.$$

□

Pondération barycentrique :
Si nous apparions la définition de l'estimateur (6.2) avec la formulation barycentrique originelle pour les centres de gravité, $\overline{\boldsymbol{v}}_j = \frac{1}{|\mathcal{A}_j|} \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{v}_\ell \quad \forall j \in [1, 2^c]$, nous déduisons que les matrices des poids sont $\boldsymbol{W}_\ell = \frac{1}{|\mathcal{A}_j|} \boldsymbol{I}, \forall \ell \in [1, c]$. Rigoureusement nous devrions définir les matrices de covariances, $\overline{\boldsymbol{Z}}_j = \frac{1}{|\mathcal{A}_j|^2} \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{Z}_\ell$.

Pondération optimale :
Nous souhaitons trouver les matrices des poids $\boldsymbol{\mathcal{W}}_j = \{\boldsymbol{W}_\ell | \omega_\ell \in A_j\}$ qui minimisent la zone d'imprécision $\overline{\boldsymbol{S}}_j = \overline{\boldsymbol{Z}}_j^{-1}$:

$$\min_{\boldsymbol{\mathcal{W}}_j} \det(\overline{\boldsymbol{S}}_j), \forall \mathcal{A}_j \subseteq 2^{\Omega},$$
$$\text{avec la contrainte(6.3).}$$

La fonction déterminant n'étant pas convexe, nous réécrivons le problème avec une combinaison par la fonction ln,

$$\min_{\boldsymbol{\mathcal{W}}_j} -\ln(\det(\overline{\boldsymbol{Z}}_j)), \forall \mathcal{A}_j \subseteq 2^{\Omega}. \tag{6.5}$$
$$\text{avec la contrainte(6.3).}$$

Or les matrices de distances de Mahalanobis sont symétriques et définies positives, $\overline{\boldsymbol{S}}_j \succ 0$. Par conséquent, par construction, $\boldsymbol{\mathcal{W}}_j$ est inversible et la nouvelle fonction est convexe. La contrainte (6.3) permet de borner l'ensemble des solutions, et ainsi d'obtenir un problème bien défini.

Proposition 6.2.2: Pondération optimale

Pour le sous-ensemble $\mathcal{A}_j$, les matrices des poids minimisant la zone d'imprécision sont les suivants , $\forall \ell \in [1, c], \omega_\ell \in \mathcal{A}_j$,

$$\boldsymbol{W}_\ell = \left(\sum_{\omega'_\ell \in \mathcal{A}_j} \boldsymbol{Z}_{\ell'}^{-1} \right)^{-1} \boldsymbol{Z}_\ell^{-1}. \tag{6.6}$$

Le sous-ensemble $\mathcal{A}_j$ est caractérisé par sa matrice de variance-covariance optimale $\overline{\boldsymbol{Z}}_j^+$ (ou sa matrice de distance $\overline{\boldsymbol{S}}_j^+$) et son centre de gravité $\overline{\boldsymbol{v}}_j^+$ définis par les formulations :

$$\overline{\boldsymbol{Z}}_j^+ = \left(\sum_{\omega'_\ell \in \mathcal{A}_j} \boldsymbol{Z}_{\ell'}^{-1} \right)^{-1},$$
$$\overline{\boldsymbol{S}}_j^+ = \sum_{\omega'_\ell \in \mathcal{A}_j} \boldsymbol{Z}_{\ell'}^{-1} = \sum_{\omega'_\ell \in \mathcal{A}_j} \boldsymbol{S}_{\ell'}, \tag{6.7}$$

$$\begin{aligned} \overline{\boldsymbol{v}}_j^+ &= \sum_{\omega_\ell \in \mathcal{A}_j} \left(\sum_{\omega'_\ell \in \mathcal{A}_j} \boldsymbol{Z}_{\ell'}^{-1} \right)^{-1} \boldsymbol{Z}_\ell^{-1} \boldsymbol{v}_\ell, \\ &= \overline{\boldsymbol{Z}}_j^+ \sum_{\omega_\ell \in \mathcal{A}_j} \boldsymbol{S}_\ell \boldsymbol{v}_\ell. \end{aligned} \tag{6.8}$$

Démonstration. La fonction de Lagrange associée à fonction objectif du problème (6.5) est,

$$\mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda}) = -ln(det(\overline{\boldsymbol{Z}}_j)) + < \boldsymbol{\Lambda}, (\sum_{\omega_\ell \in A_j} \boldsymbol{W}_\ell) - \boldsymbol{I} >_F .$$

avec $\boldsymbol{\Lambda}$ le multiplicateur de Lagrange, une matrice réelle carrée de dimension $n_d \times n_d$. Comme la fonction est convexe, la solution optimale est le point qui annule le gradient du Lagrangien :

$$\begin{aligned} \frac{\partial \mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})}{\partial \boldsymbol{W}_\ell} &= 0, \\ \frac{\partial \mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})}{\partial \boldsymbol{\Lambda}} &= 0. \end{aligned}$$

Calculs des dérivées

Pour simplifier les calculs, supposons la décomposition de $\mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})$ en deux termes :

$$\begin{aligned} \mathcal{L}_1(\boldsymbol{\mathcal{W}}_j) &= -ln(det(\overline{\boldsymbol{Z}}_j)), \\ \mathcal{L}_2(\boldsymbol{\Lambda}, \boldsymbol{\mathcal{W}}_j) &= < \boldsymbol{\Lambda}, (\sum_{\omega_\ell \in A_j} \boldsymbol{W}_\ell) - \boldsymbol{I} >_F, \end{aligned}$$

tels que $\mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda}) = \mathcal{L}_1(\boldsymbol{\mathcal{W}}_j) + \mathcal{L}_2(\boldsymbol{\Lambda}, \boldsymbol{\mathcal{W}}_j)$.

Pour le premier terme, nous utilisons la règle de dérivée d'une fonction matricielle composée d'autres matrices voir l'équation 137 dans *Matrix cook book*[1], [162, éq. 137] :

$$\frac{\partial}{\partial \boldsymbol{W}_\ell} \mathcal{L}_1(\boldsymbol{\mathcal{W}}_j) = -Tr\left(\frac{\partial ln(det(\overline{\boldsymbol{Z}}_j))}{\partial \overline{\boldsymbol{Z}}_j} \frac{\partial \overline{\boldsymbol{Z}}_j}{\partial \boldsymbol{W}_\ell}\right) = -Tr\left(\overline{\boldsymbol{Z}}_j^{-1} \frac{\partial \overline{\boldsymbol{Z}}_j}{\partial \boldsymbol{W}_\ell}\right)$$

d'après [162, éq. 57] et grâce à la propriété de symétrie de $\overline{\boldsymbol{Z}}$. Nous continuons par la dérivée partielle, en l'étudiant terme à terme (m, s), $(\boldsymbol{W}_\ell)_{ms}$ de $\boldsymbol{W}_\ell$:

$$\begin{aligned} \frac{\partial \overline{\boldsymbol{Z}}_j}{\partial (\boldsymbol{W}_\ell)_{ms}} &= \frac{\partial}{\partial (\boldsymbol{W}_\ell)_{ms}} \sum_{\omega_{\ell'} \in \mathcal{A}_j} \boldsymbol{W}_{\ell'} \boldsymbol{Z}_{\ell'} \boldsymbol{W}_{\ell'}^T, \\ &= \frac{\partial}{\partial (\boldsymbol{W}_\ell)_{ms}} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T, \\ &= \boldsymbol{W}_\ell \boldsymbol{Z}_\ell \boldsymbol{J}^{sm} + \boldsymbol{J}^{ms} \boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T, \end{aligned}$$

avec $\boldsymbol{J}^{ms}$ une matrice à une seule entrée ayant une valeur de 1 à (m,s) et des valeurs nulles ailleurs voir [162, éq. 80]. Ainsi, par linéarité de la trace et selon les [162, éq. 450 et 452], nous obtenons les relations suivantes :

$$\begin{aligned} \frac{\partial \mathcal{L}_1(\boldsymbol{\mathcal{W}}_j)}{\partial (\boldsymbol{W}_\ell)_{ms}} &= -Tr(\overline{\boldsymbol{Z}}_j^{-1} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell J^{sm}) - Tr(\overline{\boldsymbol{Z}}_j^{-1} J^{ms} \boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T), \\ &= -(\overline{\boldsymbol{Z}}_j^{-1} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell)_{sm}^T - (\boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T \overline{\boldsymbol{Z}}_j^{-1})_{sm}, \\ &= -(\boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T \overline{\boldsymbol{Z}}_j^{-T})_{sm} - (\boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T \overline{\boldsymbol{Z}}_j^{-1})_{sm}. \end{aligned}$$

1. https://www.math.uwaterloo.ca/ hwolkowi/matrixcookbook.pdf

Puisque la matrice $\overline{\boldsymbol{Z}}_j$ est symétrique défini positif $\overline{\boldsymbol{Z}}_j^{-1} = \overline{\boldsymbol{Z}}_j^{-T}$, nous avons,

$$\frac{\partial \mathcal{L}_1(\boldsymbol{\mathcal{W}}_j)}{\partial (\boldsymbol{W}_\ell)_{ms}} = -2(\boldsymbol{Z}_\ell \boldsymbol{W}_\ell^T \overline{\boldsymbol{Z}}_j^{-1})_{sm}.$$

Nous pouvons reconstruire la dérivée partielle de la première partie de la fonction $\mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})$ à partir des dérivées partielles terme à terme,

$$\frac{\partial \mathcal{L}_1(\boldsymbol{\mathcal{W}}_j)}{\partial \boldsymbol{W}_\ell} = -2\overline{\boldsymbol{Z}}_j^{-1} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell.$$

Pour la seconde partie, nous utilisons la linéarité de la trace et [162, éq. 101] :

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{W}_\ell} \mathcal{L}_2(\boldsymbol{\Lambda}, \boldsymbol{\mathcal{W}}_j) &= \frac{\partial}{\partial \boldsymbol{W}_\ell} Tr(\boldsymbol{\Lambda}((\sum_{\omega_{\ell'} \in \mathcal{A}_j} \boldsymbol{W}_{\ell'}) - \boldsymbol{I})), \\
&= \frac{\partial}{\partial \boldsymbol{W}_\ell} Tr(\boldsymbol{\Lambda} \boldsymbol{W}_\ell), \\
&= \boldsymbol{\Lambda}^T.
\end{aligned}$$

En réunissant les dérivées partielles des deux parties, $\mathcal{L}_1$ et $\mathcal{L}_2$, nous écrivons la dérivée partielle du Lagrangien selon la matrice des poids :

$$\begin{aligned}
\frac{\partial \mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})}{\partial \boldsymbol{W}_\ell} &= \frac{\partial \mathcal{L}_{11}(\boldsymbol{\mathcal{W}}_j)}{\partial \boldsymbol{W}_\ell} + \frac{\partial \mathcal{L}_{12}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})}{\partial \boldsymbol{W}_\ell}, \\
&= -2\overline{\boldsymbol{Z}}_j^{-1} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell + \boldsymbol{\Lambda}^T.
\end{aligned}$$

Ainsi, le système définissant les conditions d'optimalité est désormais le système,

$$\frac{\partial \mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})}{\partial \boldsymbol{W}_\ell} = 0 \Longleftrightarrow 2\overline{\boldsymbol{Z}}_j^{-1} \boldsymbol{W}_\ell \boldsymbol{Z}_\ell = \boldsymbol{\Lambda}^\top, \tag{6.9}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\mathcal{W}}_j, \boldsymbol{\Lambda})}{\partial \boldsymbol{\Lambda}} = 0 \Longleftrightarrow \sum_{\omega_\ell \in A_j} \boldsymbol{W}_\ell = \boldsymbol{I}. \tag{6.10}$$

Nous isolons $\boldsymbol{W}_\ell$ dans (6.9),

$$\boldsymbol{W}_\ell = \frac{1}{2} \overline{\boldsymbol{Z}}_j \boldsymbol{\Lambda}^\top \boldsymbol{Z}_\ell^{-1}. \tag{6.11}$$

Nous injectons cette formule dans l'équation (6.10), nous déduisons la valeur du multiplicateur $\boldsymbol{\Lambda}$:

$$\boldsymbol{\Lambda}^\top = 2\overline{\boldsymbol{Z}}_j^{-1} \left(\sum_{\omega_{\ell'} \in \mathcal{A}_j} \boldsymbol{Z}_{\ell'}^{-1} \right)^{-1}. \tag{6.12}$$

En substituant (6.12) dans (6.11) nous obtenons,

$$\boldsymbol{W}_\ell = \left(\sum_{\omega'_\ell \in \mathcal{A}_j} \boldsymbol{Z}_{\ell'}^{-1} \right)^{-1} \boldsymbol{Z}_\ell^{-1}. \tag{6.13}$$

Nous remarquons que tous les $\boldsymbol{W}_\ell$ sont inversibles donc $\boldsymbol{\mathcal{W}}_j$ aussi. Nous sommes assurés de minimiser le problème (6.5). □

6.2.2 Analyse géométrique

D'un point de vue géométrique, le centre de gravité optimal du sous-ensemble $\boldsymbol{\mathcal{A}}_j$ est le point de l'espace $\boldsymbol{v}_j$ qui minimise la moyenne de Fréchet définie comme la somme des distances de Mahalanobis de chaque classe :

$$\min_{\boldsymbol{v}} F(\boldsymbol{v}) = \sum_{\omega_\ell \subseteq \mathcal{A}_j} (\boldsymbol{v} - \boldsymbol{v}_\ell)^T \boldsymbol{Z}_\ell^{-1} (\boldsymbol{v} - \boldsymbol{v}_\ell), \tag{6.14}$$

Comme F est la somme de fonction quadratique, elle est convexe. Donc le vecteur $\boldsymbol{v}$ qui annule le gradient, est le minimum de la fonction.
La dérivée de la fonction est,

$$\frac{\partial F(\boldsymbol{v})}{\partial \boldsymbol{v}} = \frac{1}{2} \sum_{\omega_\ell \subseteq \mathcal{A}_j} \frac{\partial}{\partial \boldsymbol{v}} (\boldsymbol{v} - \boldsymbol{v}_\ell)^T \boldsymbol{Z}_\ell^{-1} (\boldsymbol{v} - \boldsymbol{v}_\ell) = \sum_{\omega_\ell \subseteq \mathcal{A}_j} \boldsymbol{Z}_\ell^{-1} (\boldsymbol{v} - \boldsymbol{v}_\ell).$$

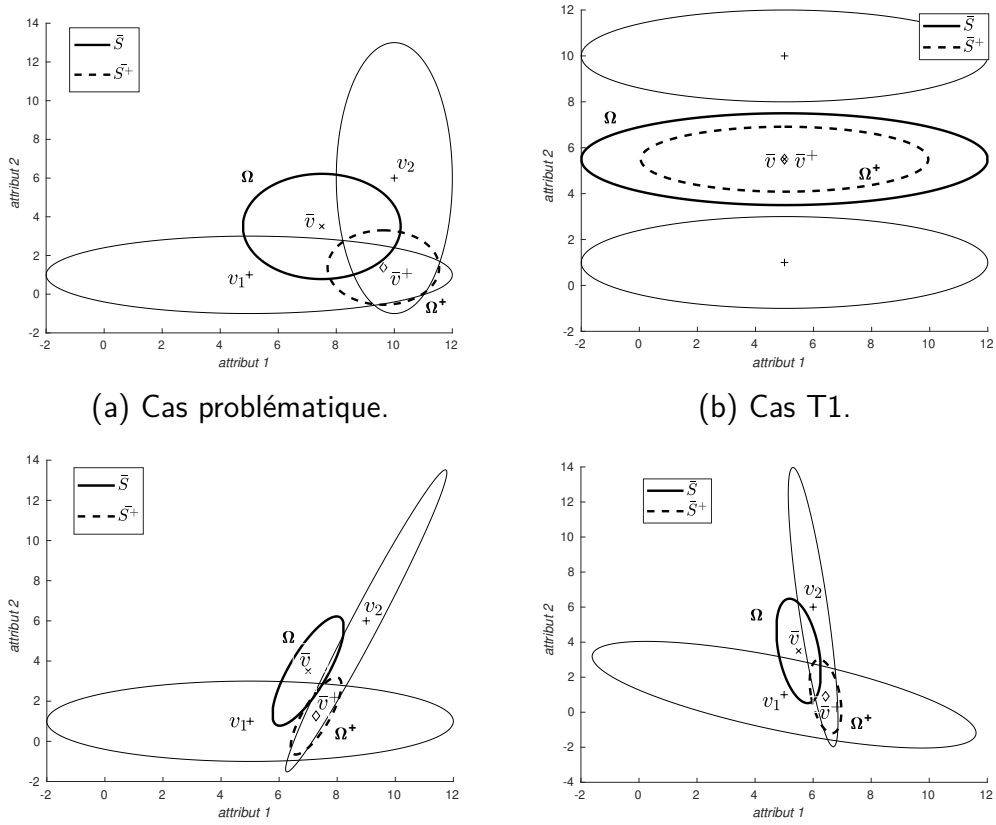
Nous l'annulons,

$$\begin{aligned} \frac{\partial F(\boldsymbol{v})}{\partial \boldsymbol{v}} = 0 \quad &\Rightarrow \quad \sum_{\omega_\ell \subseteq \mathcal{A}_j} \boldsymbol{Z}_\ell^{-1} \boldsymbol{v} = \sum_{\omega_\ell \subseteq \mathcal{A}_j} \boldsymbol{\Sigma}_\ell^{-1} \boldsymbol{v}_\ell, \\ &\Rightarrow \quad \boldsymbol{v} = \left(\sum_{\omega_\ell \subseteq \mathcal{A}_j} \boldsymbol{Z}_\ell^{-1} \right)^{-1} \sum_{\omega_\ell \subseteq \mathcal{A}_j} \boldsymbol{Z}_\ell^{-1} \boldsymbol{v}_\ell. \end{aligned}$$

Nous avons bien la même formule $\boldsymbol{v} = \overline{\boldsymbol{v}}_j^+$. La géométrie confirme les résultats obtenus par les statistiques.

6.2.3 Illustrations

Nous souhaitons comparer visuellement notre modèle, nommé ECM+ $(\overline{\boldsymbol{v}}_j^+, \overline{\boldsymbol{S}}_j^+)$ et le modèle l'orignal ECM $(\overline{\boldsymbol{v}}_j, \overline{\boldsymbol{S}}_j)$ [50]. Pour cela, nous reprenons l'exemple 6.1.1 ainsi que trois autres exemples types définis dans l'annexe A.2.2. Les quatres figures 6.2.1 présentent les partitionnements obtenus par ces deux modèles. Les formules d'ECM+ permettent de caractériser la zone d'imprécision avec justesse et précision.



(a) Cas problématique. (b) Cas T1.

(c) Cas T2. (d) Cas T3.

FIGURE 6.2.1 – Différence théorique entre ECM et ECM+.

La plus grosse différence se situe sur l'estimation du centre de gravité. Dans le cas où les classes ont la même forme comme dans le cas $T1$, la formule barycentrique est correcte. Pour la forme du sous-ensemble j, nous avons une relation intéressante $\overline{\boldsymbol{S}}_j = \frac{1}{|\mathcal{A}_j|}\overline{\boldsymbol{S}}_j^+$. Nous en déduisons que les ellipses partagent la même orientation mais ont des volumes différents. Nous avons la relation suivante : $\det(\overline{\boldsymbol{S}}_j) = \frac{1}{|\mathcal{A}_j|^{1/n_d}} \det(\overline{\boldsymbol{S}}_j^+)$. Pour tous les cas, nous observons cette différence.

6.2.4 Algorithme

L'algorithme 10 est celui de la méthode d'optimisation alternée appliquée au modèle ECM+. Il reprend l'algorithme 4 du modèle ECM en remplaçant les formules barycentriques par les formules optimales.
Nous n'avons pas ajouté ou supprimé de variable donc la complexité spatiale reste en $O(nn_d + n2^{c-1} + 2^{c-1}n_d^2 + ncn_d^2)$. Le coût supplémentaire du calcul de $\boldsymbol{\mathcal{V}}_j^k$ engendré par la somme des $\boldsymbol{S}_\ell$ $O(cn_d^2)$ et son inversion $O(n_d^3)$, n'augmentent pas la complexité temporelle totale en $O(t(n2^{2c-2}n_d^2 + (cn_d)^3))$ pour t itérations.

Algorithme 10 ECM+ par AO.

Entrée : $\boldsymbol{X}$ les données, c le nombre de classes, α,β,δ.
Sortie : $\boldsymbol{M}^k$,$\boldsymbol{\mathcal{V}}^k$,$\boldsymbol{\mathcal{S}}^k$

1: $err = 1, k = 0$,
2: $\boldsymbol{M}^0$ initialisation aléatoire ou FCM.
3: **tant que** $err > 10^{-3}$ **faire**
4: $k = k + 1$
5: Calcul de $\boldsymbol{\mathcal{V}}_\ell^k$ (3.20) *(résolution du système linéaire)* :

$$\boldsymbol{G}^{k-1}\boldsymbol{v}^k = \boldsymbol{F}^{k-1}\boldsymbol{x}.$$

6: Calcul de $\boldsymbol{\mathcal{V}}_j^k$ (6.8) *(nouvelle formule)* :

$$\overline{\boldsymbol{v}}_j^k = \left(\sum_{\omega_\ell \subseteq \mathcal{A}_j} \boldsymbol{S}_\ell^{k-1}\right)^{-1} \sum_{\omega_\ell \subseteq \mathcal{A}_j} \boldsymbol{S}_\ell^{k-1}\boldsymbol{v}_\ell^k.$$

7: Calcul de $\boldsymbol{\mathcal{S}}_\ell^k$ (3.22) :

$$\boldsymbol{\Sigma}_\ell^k = \sum_{i=1}^{n} \sum_{\mathcal{A}_j \neq \emptyset} s_{\ell j} |\mathcal{A}_j|^{\alpha-1} \left(m_{ij}^{k-1}\right)^\beta (\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j^k)(\boldsymbol{x}_i - \overline{\boldsymbol{v}}_j^k)^\top,$$

$$\boldsymbol{S}_\ell^k = det(\boldsymbol{\Sigma}_\ell^k)^{\frac{1}{p}} \left(\boldsymbol{\Sigma}_\ell^k\right)^{-1}.$$

8: Calcul de $\boldsymbol{\mathcal{S}}_j^k$ (6.7) *(nouvelle formule)* :

$$\overline{\boldsymbol{S}}_j^k = \sum_{\ell=1}^{c} s_{\ell j} \boldsymbol{S}_\ell^k.$$

9: Calcul de $\boldsymbol{M}^k$ (3.19) :

$$\forall i, \mathcal{A}_j \neq \emptyset, m_{ij}^k = \frac{|\mathcal{A}_j|^{\frac{-\alpha}{\beta-1}} d_{ij}^{\frac{-2}{\beta-1}}}{\sum_{\mathcal{A}_\ell \neq \emptyset} |\mathcal{A}_\ell|^{\frac{-\alpha}{\beta-1}} d_{i\ell}^{\frac{-2}{\beta-1}} + \delta^{-\frac{2}{\beta-1}}}, \quad m_{i\emptyset}^k = 1 - \sum_j m_{ij}^k.$$

10: $err = \| \boldsymbol{M}^k - \boldsymbol{M}^{k-1} \|$
11: **fin tant que**

6.3 Expérimentations numériques

6.3.1 Méthodologie

Grâce aux illustrations de la figure 6.2.1, nous avons visualisé la différence théorique entre ces deux modèles. À présent, nous souhaitons comparer les algorithmes ECM et ECM+, en évaluant la qualité de leur regroupement. Il existe très peu de mesures d'évaluation pour la classification non supervisée évidentielle. Nous disposons d'un indice interne, l'entropie appelée *nonspecificity* N^* (éq. 2.49), qui doit être minimisée vers 0, ainsi que deux indices externes : CRI (éq. 2.39) et ARI (éq. 2.37) à maximiser vers 1. L'Adjusted Rand Index compare deux partitions dures. Il est nécessaire de transformer les partitions crédales en partitions dures. Au lieu d'appliquer la transformation pignistique, voir sa définition 2.3.14, et le principe de défuzzification pour obtenir une partition dure, nous considérons la partition crédale dure. Elle est définie en attribuant à chaque objet le sous-ensemble ayant la masse la plus élevée. La partition crédale dure permet de détecter les objets qui peuvent être affectés sans ambiguïté à une classe, c'est-à-dire ceux dont le sous-ensemble d'affectation est un singleton, donc une classe. C'est ainsi que nous comparons la partition dure de référence avec la partition dure crédale, restreinte aux objets affectés à une classe. Nous donnons le pourcentage des objets conservés pour l'évaluation par l'ARI.

Pour n'avoir besoin que d'une seule initialisation, nous avons opté pour l'algorithme ADMM de FCM, algorithme 8 avec la distance euclidienne. Le terme de pénalité a été fixé à $r = 2,5$ et le nombre d'itérations maximal à 50. En ce qui concerne les hyperparamètres d'ECM et ECM+, ils n'ont pas fait l'objet d'une étude particulière. Nous avons conservé les valeurs standards proposées par Masson et al. [11], soit $\alpha = 1$, $\beta = 2$, et $\delta = 10$. Ils proposent de réduire la complexité computationnelle des algorithmes en se focalisant uniquement sur les sous-ensembles de cardinalité inférieure ou égale à deux, à savoir les singletons, les doublons, l'ensemble vide $\emptyset$ et l'ensemble Ω caractérisant l'incertitude totale.

Les expérimentations ont été réalisés en utilisant le logiciel MATLAB R2020a sur un ordinateur équipé d'un processeur Intel Core i5 de 10ème génération, 16 Go de RAM, fonctionnant sous le système d'exploitation Linux.

6.3.2 Jeux de données

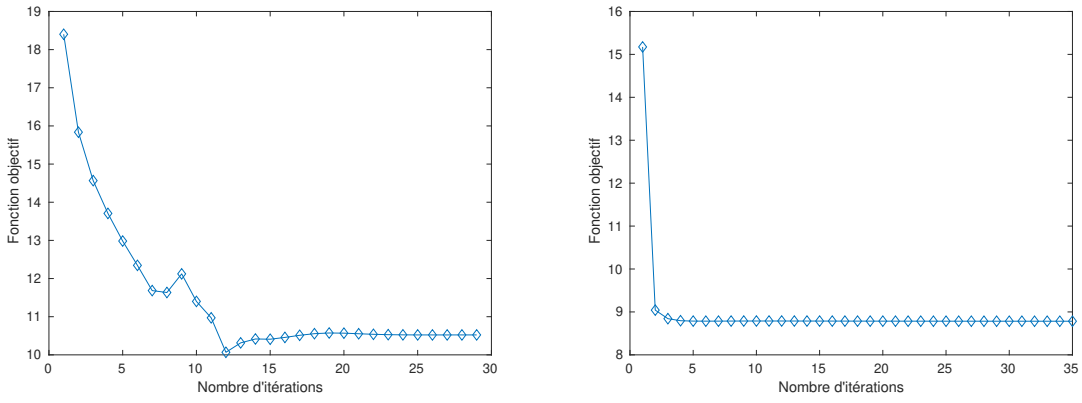
Pour choisir un jeu de données sur lequel appliquer ECM, nous devons être vigilant au nombre de classes c, même si nous examinerons seulement c^2+2 sous-ensembles et non 2^c. En effet, la limite spatiale peut être atteinte avec des jeux de données comportant un grand nombre de classes. C'est pourquoi nous n'appliquons pas les algorithmes aux données A1 et A3, qui ont respectivement 20 et 50 classes. En revanche, nous testons les six autres jeux de données synthétiques : Asymétrique, DIM32, DIM64 et Skewed, S1 et S3 présentés dans l'annexe A.3. Nous reprenons également les neuf jeux

de données réelles de l'annexe A.4 issus de la bibliothèque de l'UCI [2]. De plus, l'annexe A.2.2 détaille la construction de trois jeux de données prototypes correspondant aux cas T1, T2 et T3 de la figure 6.2.1. Ces trois proptotypes sont spécialement conçus pour l'étude d'ECM+. Ainsi, au total nous disposons de 18 jeux de données.

Dans le but de garantir que chaque attribut ait une importance égale, nous normalisons les jeux de données selon la procédure décrite dans l'annexe A.1.2.

6.3.3 ECM+ vs ECM

Tout d'abord, il est important de préciser que dans l'algorithme d'ECM+ les modifications ont été apportées uniquement au niveau des formules de mise à jour des centres de gravité des sous-ensembles et de leurs matrices associées, c'est-à-dire $\boldsymbol{\mathcal{V}}_j$ et $\boldsymbol{\mathcal{S}}_j$. Ces variables sont mises à jour en minimisant la zone d'imprécision plutôt que la fonction objectif J_{ECM} (2.30). D'un autre côté, les mises à jour des centres de gravité des classes et de leurs matrices, $\boldsymbol{\mathcal{V}}_\ell$ et $\boldsymbol{\mathcal{S}}_\ell$, n'ont pas été modifiées, et elles reposent toujours sur la formulation barycentrique. Cette minimisation est donc approximative et manque de cohérence, ce qui signifie que la monotonie de la fonction objectif J_{ECM} n'est pas garantie. Cependant, nos expérimentations ont toujours montré une convergence de la méthode. Dans certains cas, comme pour le jeu de données Skewed illustré dans la figure 6.3.1a, la fonction peut présenter des rebonds. Cependant, pour d'autres cas tels qu'Iris, comme le montre la figure 6.3.1b, la minimisation de la fonction reste monotone. Ainsi, dans cette étude préliminaire, nous avons supposé que les éventuels rebonds n'empêchent pas l'algorithme de converger.



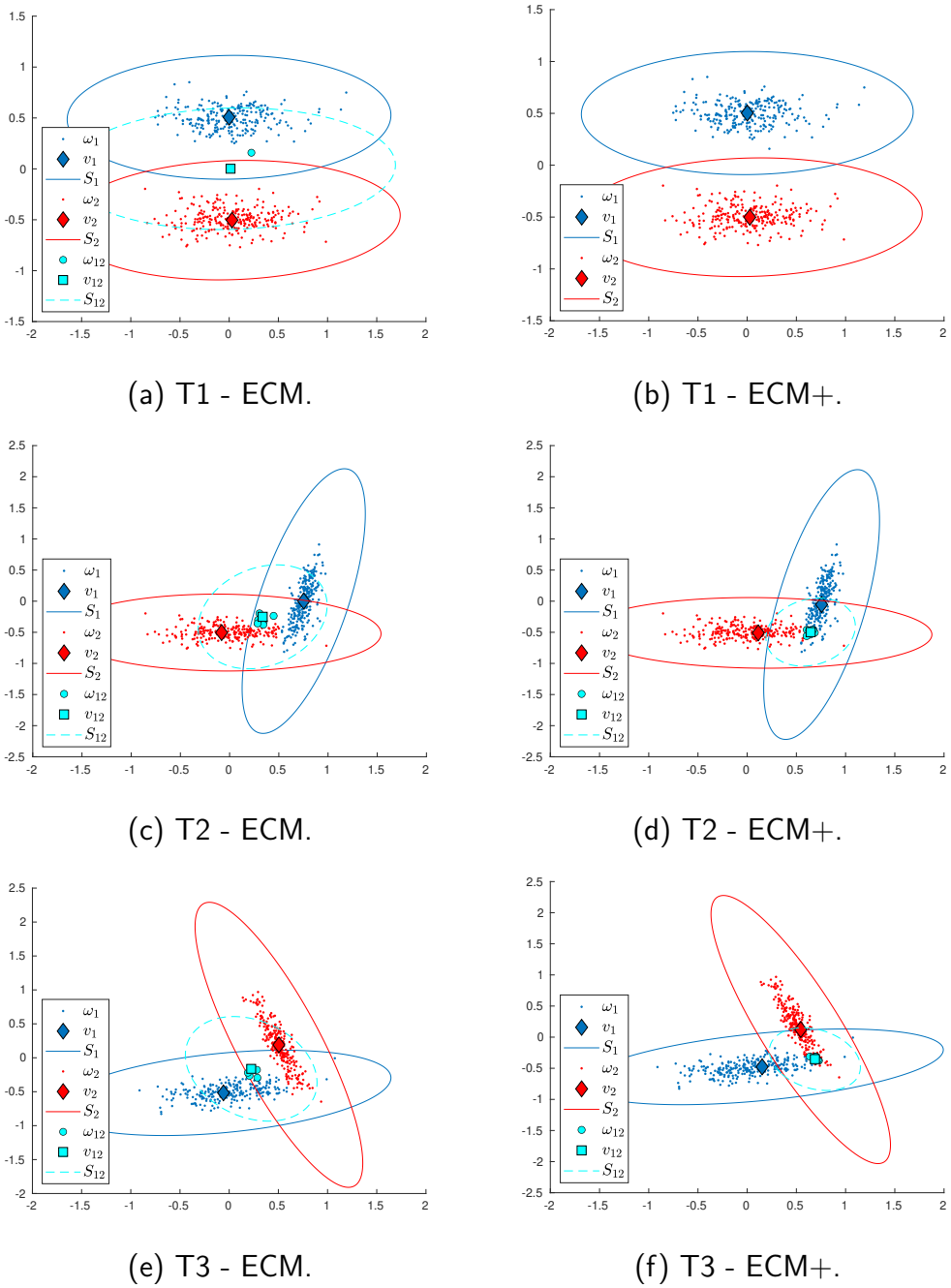
(a) Skewed. (b) Iris.

FIGURE 6.3.1 – Monotonie de la fonction objectif pour ECM+.

Les données prototypes ont été créées dans le but de mettre en évidence les problématiques rencontrées et les avantages d'une méthode par rapport à l'autre. La figure

2. https://archive.ics.uci.edu/

6.3.2 présente les regroupements obtenus par ECM et ECM+ pour les trois jeux de données T1, T2 et T3. Ces graphiques confirment les résultats théoriques obtenus précédemment. Nous pouvons constater que la zone d'imprécision est mieux ciblée par ECM+ que par ECM et qu'elle est également plus petite.



(a) T1 - ECM. (b) T1 - ECM+.

(c) T2 - ECM. (d) T2 - ECM+.

(e) T3 - ECM. (f) T3 - ECM+.

FIGURE 6.3.2 – Différence pratique entre ECM et ECM+.

Le tableau 6.1 donne le score *ARI* de la partition dure crédale, restreinte aux objets

affectés à une classe. Nous précisons le pourcentage des objets conservés quand il n'est pas égal à 100%. Les résultats obtenus vont dans le sens des observations visuelles. En effet, pour la majorité des jeux de données (11/18), les données sont affectées à une classe puisque la croyance maximale est allouée à un singleton. Pour 70% de ces cas, ECM+ obtient un score meilleur ou identique à ECM. D'autre part, pour la minorité des jeux de données (7/18), nous remarquons que le nombre d'objets retirés est plus important pour ECM.

	T1	T2	T3
ECM	1.00 (99.8%)	0.85 (98.2%)	0.94 (97.8%)
ECM+	1.00	**0.91** (96.6%)	**0.99** (96.6%)

(a) Jeux de données prototypes.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
ECM	0.53	**0.72**	**0.77**	0.70 (98%)	0.29	0.76	0.26	**0.37**	0.36
ECM+	**0.55**	0.71	0.58	**0.85**	**0.30**	**0.78**	**0.42**	0.28	0.36

(b) Jeux de données UCI.

	Asymmetric	DIM32	DIM64	S1	S3	Skewed
ECM	**0.99** (86.9%)	0.09	0.06	**0.97** (57.3%)	**0.75** (32.8%)	0.99 (75.6%)
ECM+	0.96 (89.4%)	**0.23**	**0.11**	0.62 (52.4%)	0.46 (32.8%)	0.99 (95%)

(c) Jeux de données synthétiques.

Tableau 6.1 – Comparaison d'ECM et d'ECM+ par ARI (%).

	T1	T2	T3
ECM	0.84	0.75	0.78
ECM+	**0.90**	**0.82**	**0.85**

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
ECM	0.64	0.87	0.56	0.63	0.48	0.48	0.43	0.54	0.48
ECM+	**0.70**	**0.89**	**0.71**	**0.76**	**0.56**	**0.73**	**0.48**	**0.59**	**0.52**

	Asymmetric	DIM32	DIM64	S1	S3	Skewed
ECM	0.82	0.82	0.81	**0.87**	**0.83**	0.79
ECM+	**0.85**	**0.85**	**0.83**	0.83	0.81	**0.89**

Tableau 6.2 – Comparaison d'ECM et d'ECM+ par CRI.

Afin de réaliser une observation plus complète, nous avons décidé de comparer la partition crédale avec la partition dure de référence en utilisant l'indice CRI. Les résultats sont présentés dans le tableau 6.2. L'analyse ne laisse place à aucune ambiguïté : dans 89 % des jeux de données, le modèle ECM+ aboutit à un meilleur partitionnement. Seuls les cas S1 et S3 ne montrent pas d'amélioration.

D'un point de vue de la non-spécificité, dont les valeurs sont retranscrites dans le tableau 6.3, ECM+ est meilleure. En effet, cette mesure n'évalue que l'entropie des sous-ensembles différents des singletons. Or, nous venons de voir théoriquement et également dans nos exemples qu'ECM+ présente en général des zones d'imprécisions plus petite, par conséquent, le nombre d'objets dans ces zones sont généralement moins important. Seul S1 présente une valeur légèrement meilleure pour ECM.

	T1	T2	T3
ECM	0.10	0.12	0.11
ECM+	**0.05**	**0.09**	**0.07**

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
ECM	0.13	0.03	0.23	0.19	0.27	0.26	0.33	0.24	0.27
ECM+	**0.06**	**0.02**	**0.09**	**0.07**	**0.13**	**0.07**	**0.19**	**0.15**	**0.15**

	Asymmetric	DIM32	DIM64	S1	S3	Skewed
ECM	0.10	0.19	0.20	**0.14**	0.19	0.14
ECM+	**0.08**	**0.14**	**0.16**	0.15	0.19	**0.06**

TABLEAU 6.3 – Comparaison d'ECM et d'ECM+ par N^*.

Pour compléter l'analyse de l'entropie, il est important d'étudier celle des classes. Pour ce faire, nous considérons seulement les singletons de l'ensemble puissance. Nous évaluons, avec l'indice Partition Entropy PE (éq. 2.45), la partition crédale avec les mêmes objets utilisés pour l'ARI. Les résultats, présentés dans le tableau 6.4, montrent sans surprise qu'ECM+ obtient une entropie inférieure aux exceptions $S1$ et $S3$. Nous pouvons donc conclure que notre modèle produit un meilleur regroupement selon l'entropie.

	T1	T2	T3
ECM	0.32	0.39	0.35
ECM+	**0.24**	**0.29**	**0.25**

(a) Jeux de données prototypes.

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
ECM	0.49	0.76	0.77	1.07	1.58	1.57	0.99	2.00	1.58
ECM+	**0.36**	**0.52**	**0.48**	**0.78**	**1.38**	**0.81**	**0.97**	2.00	1.58

(b) Jeux de données UCI.

	Asymmetric	DIM32	DIM64	S1	S3	Skewed
ECM	0.81	3.91	4	**1.73**	**2.09**	1.00
ECM+	**0.74**	**3.40**	**3.93**	2.34	2.95	**0.76**

(c) Jeux de données synthétiques.

TABLEAU 6.4 – Comparaison d'ECM et d'ECM+ par PE.

Le temps d'exécution CPU est renseigné dans le tableau 6.5. Les deux modèles utilisant la même méthode d'optimisation ont la même complexité temporelle en théorie. Nos expérimentations numériques vont dans ce sens puisqu'il n'y a pas un modèle qui a systématiquement un temps significativement plus faible.

	T1	T2	T3
ECM	**7.0e-2**	2.1e-1	1.1e-1
ECM+	1.0e-1	**9.0e-2**	**8.0e-2**

	AG	DB	Glass	Iris	IJL	Seed	WDBC	Wifi	Wine
ECM	2.2e-1	3.2e+2	**1.5e-1**	3.3e-1	**8.6e-1**	1.8e+0	9.5e-1	**3.3e+0**	**2.7e-1**
ECM+	**2.1e-1**	**2.5e+2**	2.7e-1	**2.5e-1**	1.0e+1	**4.2e-1**	**7.0e-1**	6.1e+0	9.5e-1

	Asymmetric	DIM32	DIM64	S1	S3	Skewed
ECM	**7.4e-1**	**1.2e+3**	**2.8e+2**	**8.4e+2**	**1.4e+3**	**5.8e+0**
ECM+	2.5e+0	4.5e+3	3.8e+3	6.9e+3	6.7e+3	1.0e+2

Tableau 6.5 – Temps CPU de l'exécution d'ECM et d'ECM+.

6.4 Conclusion

Au début de ce chapitre, nous avons mis en exergue le défi d'appliquer la distance de Mahalanobis avec le modèle évidentiel ECM. La formulation barycentrique, pour construire les centroïdes et les ellipsoïdes des sous-ensembles en fonction de ceux des classes, est pertinente avec une distance euclidienne. Cependant, elle délimite mal les zones d'imprécisions pour la distance de Mahalanobis.

En nous inspirant des travaux de Davis [161], nous proposons une nouvelle formulation obtenue à la fois par une approche statistique et par une approche géométrique. D'un point de vue statistique, nous affinons la moyenne et l'écart-type correspondant aux centroïdes et à la matrice de variance-covariance en minimisant la zone d'imprécision grâce au calcul des poids qui la réduisent au maximum. Du point de vue géométrique, nous recherchons le point qui minimise la somme des distances.

Les résultats obtenus théoriquement montrent qu'ECM+ identifie mieux la zone d'imprécision en terme de position, les centroïdes, ainsi qu'en terme de forme, en assurant une meilleure adéquation entre l'ellipse et cette zone. De plus, le volume de l'ellipsoïde est généralement plus petit, car le déterminant de l'inverse de la matrice de variance-covariance évidentielle d'ECM+ est plus petit que celui d'ECM.

Les expérimentations numériques ont confirmé ces résultats. Les zones d'imprécision sont mieux cernées et le nombre d'objets inclus dans cette zone est souvent plus faible.

Les améliorations en terme de partitionnement selon l'entropie, N^* et PE, et le CRI sont significatives et sans équivoque : ECM+ parvient à obtenir un meilleur regroupement pour un temps de calcul similaire.

Dans cette étude, nous avons été limités dans notre analyse en raison du nombre restreint de mesures d'évaluation interne disponibles. Il serait important de développer de nouveaux indices permettant de décrire la qualité des partitions évidentielles, en particulier en tenant compte de l'impact des sous-ensembles.

Nos expérimentations ont révélé que malgré la précision de la délimitation des zones d'imprécision, celles-ci ne sont pas pleinement exploitées. En effet, il y a très peu de données qui ont une fonction de croyance maximale pour un des sous-ensembles hors singletons. D'un autre côté, nous savons qu'avec la distance euclidienne, l'importance des sous-ensembles est plus significative. Il serait donc intéressant d'étudier de manière approfondie l'impact concret des sous-ensembles dans le cas de la distance de Mahalanobis, en comparant notamment ECM avec FCM. L'objectif est de tirer le meilleur parti possible de ce modèle. Les zones de chevauchement des ellipses correspondent en réalité aux zones d'imprécision, il est essentiel d'en tenir compte.

Pour compléter cette étude, nous souhaitons intégrer la nouvelle formulation pour optimiser les centres de gravité des classes et leurs matrices de distance associées, afin d'obtenir une optimisation plus cohérente. Il serait envisageable d'étendre les nouvelles formules d'ECM+ au modèle avec contraintes sur les paires d'objets, CECM [50], à partir duquel la problématique de ce chapitre est née.

Chapitre 7

Conclusion et perspectives

L'utilisation de la distance de Mahalanobis offre de nombreuses opportunités pour le modèle de classification non supervisée HCM et ses variantes. Néanmoins, elle soulève également de nouveaux défis. Nous venons d'explorer trois défis majeurs, les deux premiers concernant la variante floue FCM, tandis que le dernier concerne la variante évidentielle ECM.

En premier lieu, il est essentiel de développer des indices capables d'évaluer une partition basée sur la distance de Mahalanobis. Pour ce faire, nous avons étendu la mesure d'évaluation interne de Xie-Beni (XB). Nous avons remplacé la distance euclidienne, par la distance de Mahalanobis pour mesurer la compacité, et par la distance de Wasserstein pour évaluer la séparabilité. Ces distances nous permettent de prendre en compte la forme des classes. D'après nos expérimentations numériques, nous avons observé que notre indice $XBMW$ améliore significativement la précision. Notre méthodologie s'appuie sur la vérification des indices internes par les indices externes. Cette démarche est novatrice. Enfin, nous avons constaté que ce n'était pas seulement une mesure d'évaluation, mais également un outil d'aide à la décision pour déterminer quelle distance est préférable d'employer : la distance euclidienne ou la distance de Mahalanobis.

Le point central de cette étude concerne l'optimisation du modèle FCM avec la distance de Mahalanobis. L'objectif est d'obtenir un meilleur regroupement en trouvant un meilleur minimum local du problème. En effet, l'utilisation de cette distance complexifie l'optimisation, ce qui nous pousse à explorer des méthodes plus robustes. Tout d'abord, pour résoudre le problème global, nous proposons d'appliquer la méthode des directions alternées, l'Alternating Direction Method of Multipliers (ADMM). Dans nos expériences numériques, nous avons observé que notre méthode conduit à un partitionnement différent de celui obtenu par l'optimisation alternée et qu'elle est meilleure lorsque le nombre d'attributs est suffisamment élevé par rapport au nombre de classes. Dans un second temps, pour la mise à jour des matrices de la distance de Mahalanobis, nous optons pour l'application de la méthode du gradient projeté accéléré (APG) de Nesterov. Cette approche permet de rendre le sous-problème convexe et les résultats que nous obtenons sont intéressants, générant un partitionnement encore différent. En somme, notre étude apporte un regard nouveau sur la classification non supervisée.

En dernier lieu, la gestion des zones d'imprécision dans ECM représente un enjeu majeur. Néanmoins, la formulation barycentrique, qui est adaptée à la distance euclidienne, n'est pas pertinente dans le contexte de la distance de Mahalanobis. Pour y remédier, nous avons proposé une nouvelle formulation, ECM+, spécifiquement conçue pour la distance de Mahalanobis. Nos expérimentations numériques sont en adéquation avec les propriétés théoriques. ECM+ est tout simplement bien plus précise et performante.

À la fin de ces travaux, plusieurs perspectives s'offrent à nous. Nous envisageons

de développer de nouvelles mesures d'évaluation pour les modèles de classification non supervisée basés sur la distance de Mahalanobis et d'appliquer nos méthodes d'optimisation à d'autres modèles.

Mesures internes :

— Tout d'abord, en ce qui concerne FCM, il est nécessaire d'envisager des extensions supplémentaires des mesures internes pour tenir compte de la forme des classes générées par FCM-GK. Une piste intéressante serait l'indice FS (Fuzzy Silhouette), qui propose une approche différente et complémentaire par rapport à XB. L'objectif sous-jacent est de fournir un outil complet permettant de choisir la métrique la plus adaptée en fonction de cet indice.

— L'analyse des méthodes de classification non supervisée évidentielle est également confrontée au faible nombre de mesures internes disponibles pour évaluer les partitions obtenues. Nous pouvons envisager d'étendre certaines mesures internes floues existantes pour les adapter au contexte évidentiel. Cependant, il serait également nécessaire de créer de nouvelles mesures spécifiques permettant d'évaluer la pertinence des sous-ensembles du modèle évidentiel.

Méthodes d'optimisation :

— Afin que le plus grand nombre puisse exploiter ADMM et AO-APG, il est essentiel de développer un paramétrage adaptatif qui ne nécessite pas d'intervention de l'utilisateur. De plus, l'intégration d'APG à ADMM pour former ADMM-APG pourrait constituer un véritable atout en combinant les avantages de ces deux méthodes. Toutefois, il convient de noter que le choix des hyperparamètres pourrait représenter un défi majeur.

— L'optimisation d'ECM par ADMM semble être la suite logique à cette étude. Son application devrait être simple, si nous reprenons la décomposition des variables proposées pour FCM. Cette approche prometteuse pourrait également être étendue à ECM+ ou à d'autres modèles de classification non supervisée.

— Une autre perspective serait d'explorer l'optimisation de fonctions bi-objectif, en utilisant par exemple la formulation proposée par Fukuyama et al. [163]. Cela permettrait à la fois d'optimiser la compacité et la séparabilité des classes.

— Il serait également très intéressant d'explorer l'application de la nouvelle formulation d'ECM+ à des problèmes d'apprentissage semi-supervisé sous contrainte, par exemple CECM [50].

Annexe A

Jeux de données

Contents

A.1 Mise à l'échelle

Dans cette étude, nous supposons que les données utilisées sont propres, elles n'ont pas de bruit. Il reste un seul pré-traitement à effectuer, la mise à l'échelle des différents attributs. En effet, l'ordre de grandeur peut varier significativement d'un attribut à un autre. Pour éviter tout impact sur la classification non supervisée, il est nécessaire de standardiser ou normaliser les données. L'avantage de la normalisation est qu'elle ne requiert aucune hypothèse particulière, contrairement à la standardisation qui suppose une distribution gaussienne des données. De plus, avec la normalisation, toutes les valeurs des données sont ramenées dans l'intervalle [-1, 1].

A.1.1 Standardisation

Soit le jeu de données représenté par une matrice réelle $\boldsymbol{X}$ de taille $n \times n_d$ (nombre d'objets par nombre d'attributs). La standardisation signifie centrer la variable à zéro et standardiser la variance à 1. La procédure consiste à soustraire la moyenne de chaque observation, puis à diviser par l'écart type. Les valeurs redimensionnées des objets par attribut ont les propriétés d'une distribution normale centrée réduite.

Définition A.1.1: Standardisation

Soit le jeu de données X, et sa standardisation $\boldsymbol{X}_s$ se défini selon chaque attribut,

$$\forall \ell \in [1, n_d], \quad \boldsymbol{X}_s(., \ell) = \frac{\boldsymbol{X}(., \ell) - mean_\ell(\boldsymbol{X})}{std_\ell(\boldsymbol{X})}, \tag{A.1}$$

avec $mean_\ell(\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{c} \boldsymbol{X}(i, \ell)$ et $std_\ell(\boldsymbol{X}) = \sqrt{\frac{1}{n}\sum_{i=1}^{c} \left(\boldsymbol{X}(i, \ell) - mean_\ell(\boldsymbol{X})\right)^2}$.

A.1.2 Normalisation

Il existe la normalisation min-max qui transforme les données dans [0,1]. Mais nous utilisons la normalisation moyenne (*mean normalization*) qui recentre les données en 0 et réduit l'intervalle à [-1, 1].

Définition A.1.2: Normalisation

Soit le jeu de données X, et sa normalisation X_n se défini selon chaque attribut,

$$\forall \ell \in [1, n_d], \quad X_n(., \ell) = \frac{X(., \ell) - mean_\ell(X)}{max(X(., \ell)) - min(X(., \ell))}. \tag{A.2}$$

A.2 Jeux de données prototypes

Pour mettre en évidence les caractéristiques théoriques des indices ou des modèles nous avons besoin de créer des prototypes.

A.2.1 Pour l'évaluation de XBMW

Nous souhaitons obtenir des classes de formes ellipsoïdales variées, en modifiant à la fois leur rotation et leur longueurs d'axe. Plus particulièrement, dans le but d'analyser les mesures d'évaluation interne, nous avons besoin de tester différents scénarios. Par exemple, lorsque deux centroïdes sont très proches les uns des autres, mais que leurs ellipses associées sont différentes (cas T3). Ou encore, lorsque deux jeux de données distincts partagent les mêmes centres de gravité, mais diffèrent uniquement par l'orientation de leurs ellipses (cas T1-T6). Enfin, nous considérons également le cas où chaque classe possède la même forme ellipsoïdale (cas T1).

Ainsi, nous avons créé six jeux de données, chacun étant une combinaison de plusieurs classes. Chaque classe ω est caractérisée par une ellipse dont le tableau A.1 fournit le centre $\boldsymbol{v}$, les longueurs des axes a et b, ainsi que l'angle de rotation θ. Les données ont été générées en utilisant la fonction *mvrand* de MATLAB, avec 100 objets par classe.

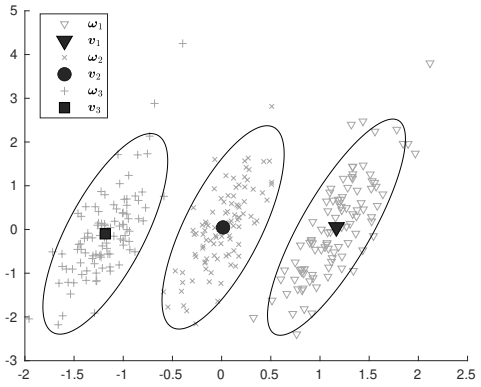
On note $-\omega$, la classe dont son centre $-\boldsymbol{v}$ est l'opposé du centre de la classe ω et qui partage les mêmes paramètres a, b, θ. Voici la liste de composition de chaque jeu de données.

- T1 : $\{\omega_1, \omega_2, -\omega_1\}$,
- T2 : $\{\omega_1, \omega_2, \omega_3\}$,
- T3 : $\{\omega_4, \omega_5\}$,
- T4 : $\{\omega_4, \omega_5, \omega_6, -\omega_6, \omega_7, -\omega_7\}$,
- T5 : $\{\omega_1, \omega_8, \omega_9, \omega_{10}, \omega_{11}\}$,
- T6 : $\{\omega_{12}, \omega_{13}, -\omega_1\}$.

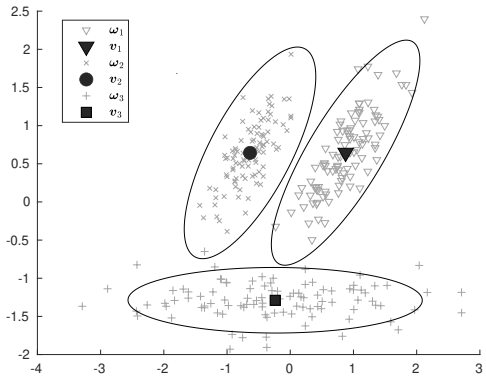
	$\boldsymbol{v}$	a	b	θ ($^\circ$)
ω_1	(0 ;3/5)	1/6	1/18	30
ω_2	(0 ;0)	1/6	1/18	30
ω_3	(-2/5 ;0)	1/2	1/18	0
ω_4	(0 ;0)	2	1/10	0
ω_5	(0 ;0)	1/3	1/9	10
ω_6	(-3 ;3)	1	1	0
ω_7	(3 ;3)	2	1/4	0
ω_8	(1.2 ;0)	1/6	1/18	-30
ω_9	(-1/2 ;-1/3)	1/12	1/12	0
ω_{10}	(-0.9 ;-1/3)	1/12	1/12	0
ω_{11}	(0 ;-1/6)	1/6	1/12	45
ω_{12}	(3/5 ;0)	1/6	1/18	-30
ω_{13}	(0 ;0)	1/6	1/18	0

TABLEAU A.1 – Caractéristiques des classes ellipsoïdales (Étude XBMW).

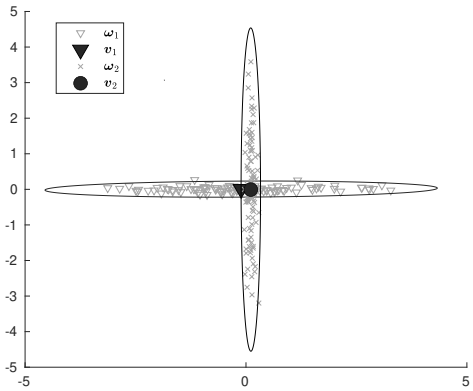
La figure A.2.1 présente les 6 jeux de données prototypes standardisés.



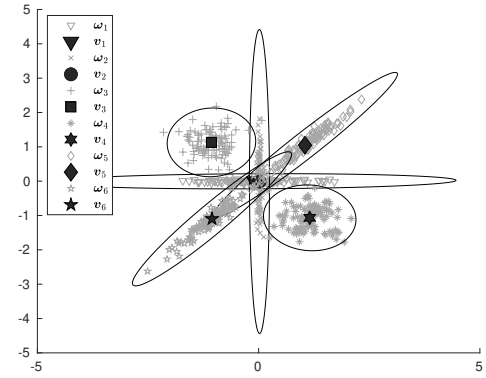
(a) T1.



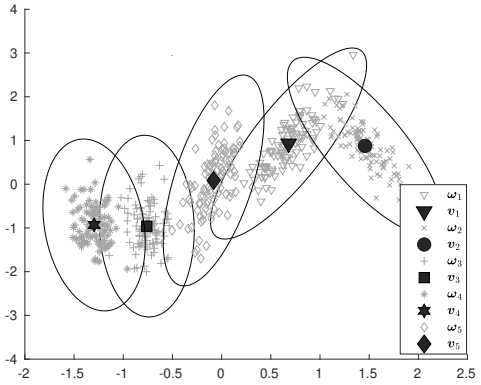
(b) T2.



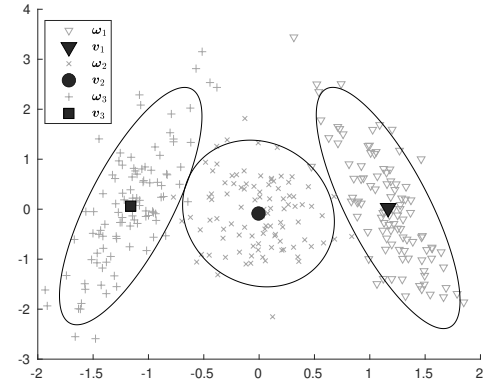
(c) T3.



(d) T4.



(e) T5.



(f) T6.

FIGURE A.2.1 – Jeux de données prototypes (Étude XBMW).

A.2.2 Pour l'étude de la classification évidentielle

Nous souhaitons étudier la gestion de la zone d'incertitude par ECM et ECM+ dans différents scénarios. Nous avons créé trois jeux de données, chacun composé de deux classes, ω_1 et ω_2. Tout d'abord, nous avons considéré le cas où la formule barycentrique est appropriée pour définir le sous-ensemble $\omega_{1\cup2}$ (cas T1). Ensuite, nous proposons les cas où le barycentre est plus éloigné (cas T2) ou plus proche (cas T3) de la zone réelle d'incertitude. Comme pour les jeux de données prototypes précédents, les classes sont définies par des ellipses. Le tableau A.2 fournit leurs caractéristiques, les centroïdes $\boldsymbol{v}$, les longueurs des axes a et b, ainsi que l'angle de rotation θ. Nous avons utilisé la fonction *mvrand* pour générer 250 objets par classe.

	$\boldsymbol{v}$	a	b	θ ($^\circ$)
ω_1	(0 ;-0.5)	1/3	1/9	0
ω_2	(0 ;0.5)	1/3	1/9	0
ω_3	(0.75 ;0)	1/3	1/18	80
ω_4	(0.5 ;0.2)	1/3	1/18	-70
ω_5	(0 ;-0.5)	1/3	1/9	10

Tableau A.2 – Caractéristiques des classes ellipsoïdales (Etude ECM+).

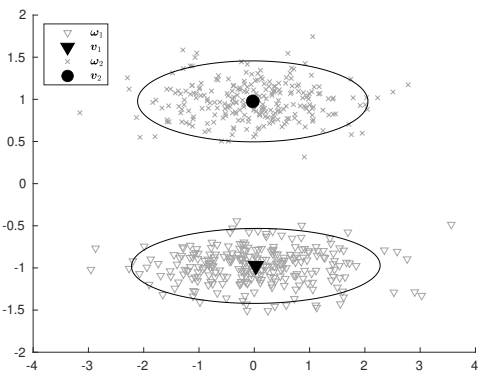
La figure A.2.2 donne la représentation des trois jeux de données prototypes composés des classes selon la liste suivante :
— T1 : $\{\omega_1, \omega_2\}$,
— T2 : $\{\omega_1, \omega_3\}$,
— T3 : $\{\omega_4, \omega_5\}$.

A.3 Jeux de données synthétiques

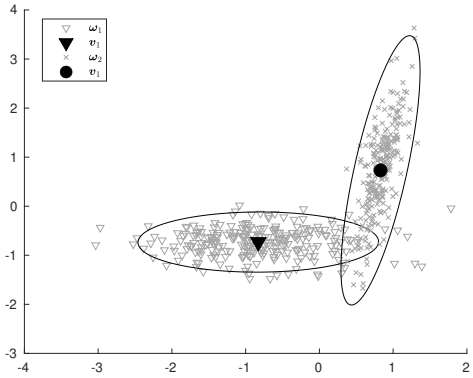
Tous les jeux de données synthétiques proviennent des études de Fränti [1]. Dans le but d'évaluer les propriétés de *k-means*, Fränti et Sieranoja [164] ont créé A-sets, S-sets et DIM-sets. Les deux premiers permetent d'évaluer la robustesse du modèle par rapport au nombre de classes. De plus S-sets propose également différents taux de chevauchement entre les classes. Enfin, DIM-sets offre la possibilité de vérifier la mise à l'échelle de la méthode, car le nombre de dimensions varie pour un nombre fixe de classes et d'objets. Plus récemment, Rezaei et Fränti [165] ont créé d'autres données synthétiques, notamment Asymmetric et Skewed, qui permettent d'évaluer les modèles face à l'asymétrie des données.

Le tableau A.3 présente les caractéristiques de chaque jeu de données synthétiques : le nombre de classes c, le nombre d'objets n et le nombre d'attributs n_d.

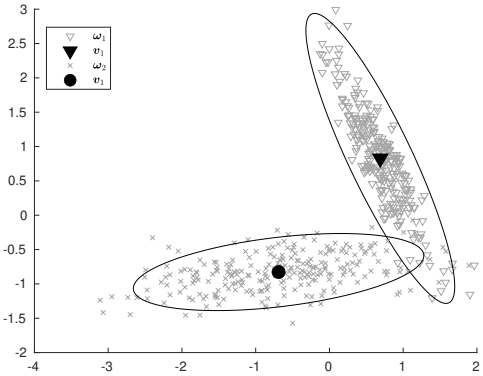
1. https://cs.joensuu.fi/sipu/datasets/



(a) T1.



(b) T2.



(c) T3.

FIGURE A.2.2 – Jeux de données prototypes (Etude ECM+).

	c	n	n_d
A1	20	3000	2
A3	50	7500	2
S1	15	5000	2
S3	15	5000	2
DIM032	16	1024	32
DIM064	16	1024	64
Asymetric	5	1000	2
Skewed	6	1000	2

TABLEAU A.3 – Caractéristiques des jeux de données synthétiques.

A.4 Jeux de données UCI

La bibliothèque de l'UCI [2] recense de nombreux jeux de données pour la classification non supervisée, la classification et la régression. Dans cette étude, nous avons retenu les jeux de données pour lesquels appliquer HCM ou ses variantes de *k-means* est pertinent. Les jeux de données proviennent de domaines variés, ce qui démontre l'interdisciplinarité des applications de la classification non supervisée.

Nous listons ci-dessous les neuf jeux de données retenus, par ordre alphabétique : Algerian forest (AF) [166], Dry bean(DB) [167] , Glass [168], Iris [169], classes I, J, et L de Letters (IJL) [170], Seeds [171], WDBC [172] , Wifi [173], Wine [174]. Depuis les travaux de Bilenko et al. [175], il est courant de se restreindre aux lettres les plus ressemblantes, à savoir I, J et L. Pour le jeu de données Glass, nous utilisons la version à deux classes, verres flottants ou non, plutôt que la version à sept classes, par type de verre.

Le tableau A.4 répertorie les caractéristiques des données de l'UCI : le nombre de classes c, le nombre d'objets n et le nombre d'attributs n_d.

	c	n	n_d
AG	2	243	10
DB	7	13611	16
Glass	2	214	9
Iris	3	150	4
IJL	3	2263	16
Seed	3	210	7
WDBC	2	569	30
Wifi	4	2000	7
Wine	3	178	13

Tableau A.4 – Caractéristiques des jeux de données de l'UCI.

2. https://archive.ics.uci.edu/

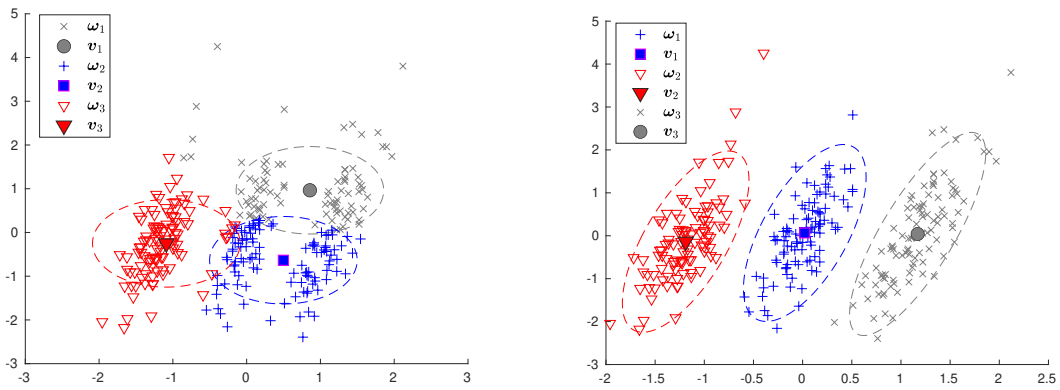
Annexe B

Résultats détaillés pour XBMW

Dans cette section, nous détaillons les résultats obtenus dans le chapitre 4. Nous présentons visuellement le partitionnement obtenu par FCM et GK pour les jeux de données prototypes spécialement conçus pour cette analyse. L'objectif est d'illustrer l'intérêt de $XBMW$.

Prototype T1

Commençons par observer le cas T1, les regroupements obtenus par les modèles sont présentés dans les figures B.0.1a et B.0.1b. Il est évident que GK trouve le bon partitionnement, contrairement à FCM. Cependant, XB ainsi que $XBMW$ inquent l'inverse, ils suggèrent que la métrique euclidienne est meilleure. Puisque les trois classes ont la même ellipse, la distance de Wasserstein est réduite à la distance euclidienne, et $XBMW$ ne prédit pas correctement la meilleure métrique.

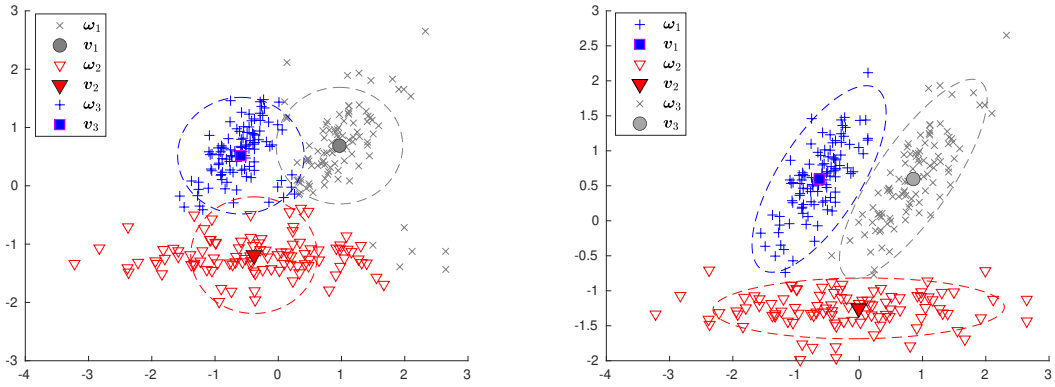


(a) FCM. (b) GK.

FIGURE B.0.1 – Partitionnement flou de T1.

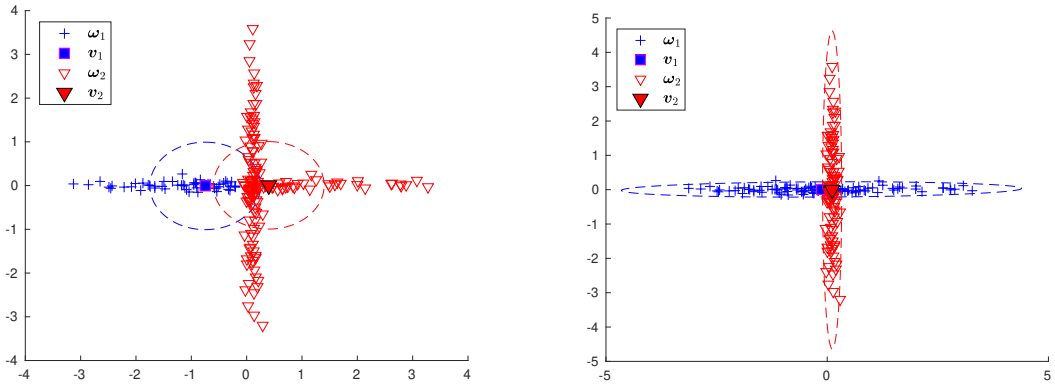
Prototype T2 à T6

Pour les autres jeux de données, nous ne faisons pas de distinction, car pour chacun d'entre eux, $XBMW$ détermine correctement qu'il est préférable d'utiliser la distance de Mahalanobis. En témoignent les figures B.0.2-B.0.6, GK retrouve le partitionnement d'origine.



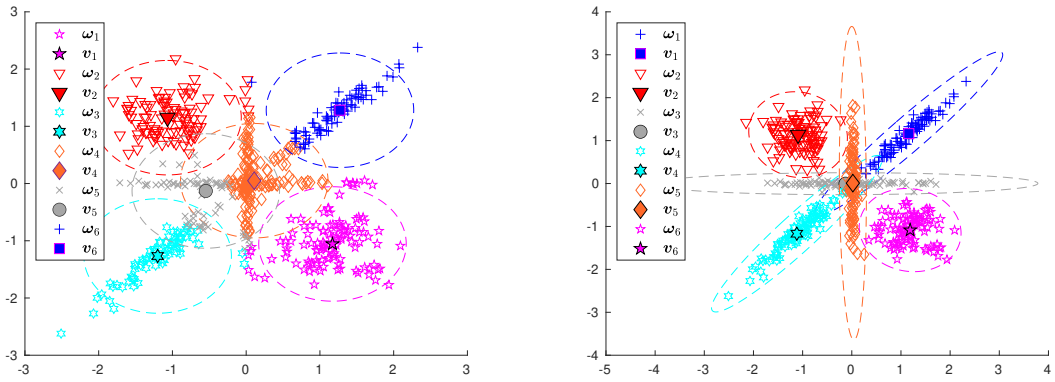
(a) FCM. (b) GK.

FIGURE B.0.2 – Partitionnement flou de T2.



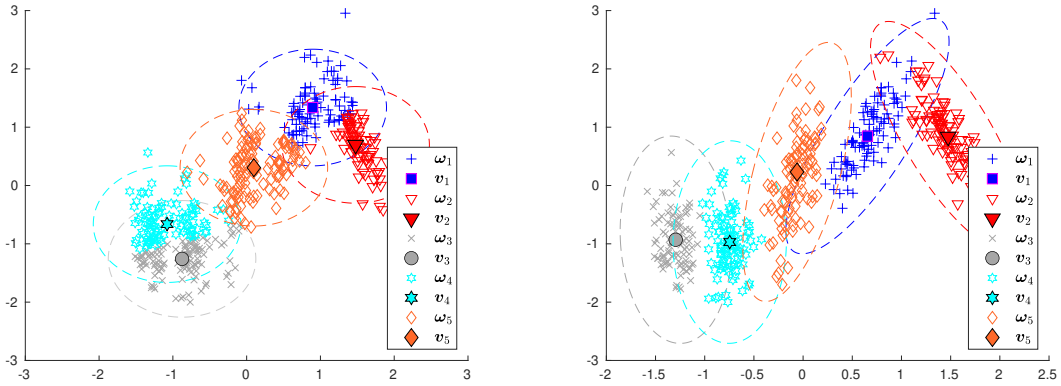
(a) FCM. (b) GK.

FIGURE B.0.3 – Partitionnement flou de T3.



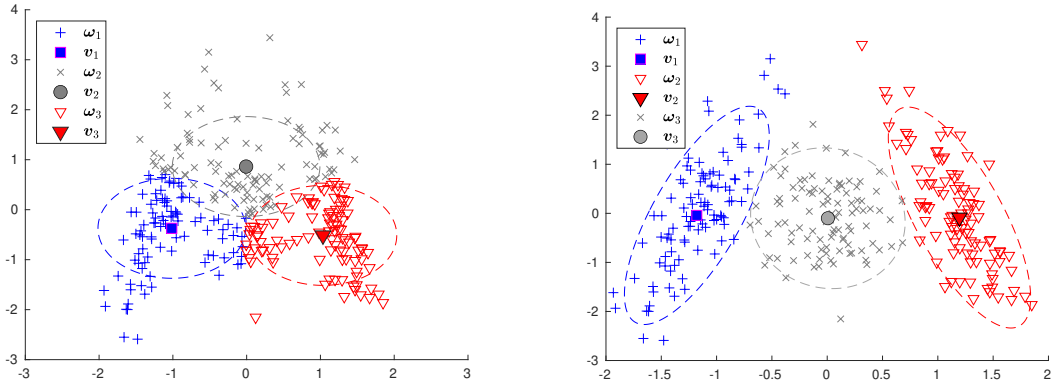
(a) FCM. (b) GK.

FIGURE B.0.4 – Partitionnement flou de T4.



(a) FCM. (b) GK.

FIGURE B.0.5 – Partitionnement flou de T5.



(a) FCM. (b) GK.

FIGURE B.0.6 – Partitionnement flou de T6.

Bibliographie

[1] D. J. Hand, "Principles of data mining," Drug safety, vol. 30, pp. 621–622, 2007.

[2] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin, "A review of clustering techniques and developments," Neurocomputing, vol. 267, pp. 664–681, 2017.

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering : a review," ACM computing surveys (CSUR), vol. 31, no. 3, pp. 264–323, 1999.

[4] A. Ghosal, A. Nandy, A. K. Das, S. Goswami, and M. Panday, "A short review on different clustering techniques and their applications," Emerging technology in modelling and graphics, pp. 69–83, 2020.

[5] J. C. Bezdek, Fuzzy Mathematics in pattern classification. Cornell University, 1973.

[6] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 13, no. 08, pp. 841–847, 1991.

[7] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique, vol. 9, no. 2, pp. 41–76, 1975.

[8] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," Computers & mathematics with applications, vol. 2, no. 1, pp. 17–40, 1976.

[9] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate o (1/kˆ 2)," in Dokl. akad. nauk Sssr, vol. 269, 1983, pp. 543–547.

[10] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM journal on imaging sciences, vol. 2, no. 1, pp. 183–202, 2009.

[11] M.-H. Masson and T. Denoeux, "Ecm : An evidential version of the fuzzy c-means algorithm," Pattern Recognition, vol. 41, no. 4, pp. 1384–1397, 2008.

[12] A. K. Jain and R. C. Dubes, Algorithms for clustering data. Prentice-Hall, Inc., 1988.

[13] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering : an overview," Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery, vol. 2, no. 1, pp. 86–97, 2012.

[14] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering : an overview, ii," Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery, vol. 7, no. 6, p. e1219, 2017.

[15] M. E. Celebi, Partitional clustering algorithms. Springer, 2014.

[16] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," Swarm and Evolutionary computation, vol. 16, pp. 1–18, 2014.

[17] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," Wiley interdisciplinary reviews : data mining and knowledge discovery, vol. 1, no. 3, pp. 231–240, 2011.

[18] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms," Frontiers of Computer Science, vol. 15, pp. 1–27, 2021.

[19] C. Bouveyron and C. Brunet-Saumard, "Model-based clustering of high-dimensional data : A review," Computational Statistics & Data Analysis, vol. 71, pp. 52–78, 2014.

[20] A. K. Jain, "Data clustering : 50 years beyond k-means," Pattern recognition letters, vol. 31, no. 8, pp. 651–666, 2010.

[21] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms : A comprehensive review, variants analysis, and advances in the era of big data," Information Sciences, 2022.

[22] P. Govender and V. Sivakumar, "Application of k-means and hierarchical clustering techniques for analysis of air pollution : A review (1980–2019)," Atmospheric pollution research, vol. 11, no. 1, pp. 40–56, 2020.

[23] W. Shi and W. Zeng, "Application of k-means clustering to environmental risk zoning of the chemical industrial area," Frontiers of Environmental Science & Engineering, vol. 8, pp. 117–127, 2014.

[24] S. Madhukumar and N. Santhiyakumari, "Evaluation of k-means and fuzzy c-means segmentation on mr images of brain," The Egyptian Journal of Radiology and Nuclear Medicine, vol. 46, no. 2, pp. 475–479, 2015.

[25] K. G. Al-Hashedi and P. Magalingam, "Financial fraud detection applying data mining techniques : A comprehensive review from 2009 to 2019," Computer Science Review, vol. 40, p. 100402, 2021.

[26] T. Kansal, S. Bahuguna, V. Singh, and T. Choudhury, "Customer segmentation using k-means clustering," in 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS). IEEE, 2018, pp. 135–139.

[27] J. Yanase and E. Triantaphyllou, "A systematic survey of computer-aided diagnosis in medicine : Past and present developments," Expert Systems with Applications, vol. 138, p. 112821, 2019.

[28] I. K. Vlachos and G. D. Sergiadis, "Intuitionistic fuzzy information–applications to pattern recognition," Pattern Recognition Letters, vol. 28, no. 2, pp. 197–206, 2007.

[29] J. MacQueen, "Classification and analysis of multivariate observations," in 5th Berkeley Symp. Math. Statist. Probability. University of California Los Angeles LA USA, 1967, pp. 281–297.

[30] S. Lloyd, "Least squares quantization in pcm," IEEE transactions on information theory, vol. 28, no. 2, pp. 129–137, 1982.

[31] K. Sabo and R. Scitovski, "Interpretation and optimization of the k-means algorithm," Applications of mathematics, vol. 59, pp. 391–406, 2014.

[32] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is np-hard," Theoretical Computer Science, vol. 442, pp. 13–21, 2012.

[33] L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8, no. 3, pp. 338–353, 1965.

[34] J. Bezdek and J. Dunn, "Optimal fuzzy partitions : A heuristic for estimating the parameters in a mixture of normal distributions," IEEE Transactions on Computers, vol. 100, no. 8, pp. 835–838, 1975.

[35] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," IEEE Transactions on Fuzzy systems, vol. 3, no. 3, pp. 370–379, 1995.

[36] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," Fuzzy sets and systems, vol. 1, no. 1, pp. 3–28, 1978.

[37] E. H. Ruspini, J. C. Bezdek, and J. M. Keller, "Fuzzy clustering : A historical perspective," IEEE Computational Intelligence Magazine, vol. 14, no. 1, pp. 45–55, 2019.

[38] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," IEEE transactions on fuzzy systems, vol. 1, no. 2, pp. 98–110, 1993.

[39] R. Krishnapuram and J. M. Keller, "The possibilistic c-means algorithm : insights and recommendations," IEEE transactions on Fuzzy Systems, vol. 4, no. 3, pp. 385–393, 1996.

[40] N. R. Pal, K. Pal, and J. C. Bezdek, "A mixed c-means clustering model," in Proceedings of 6th international fuzzy systems conference, vol. 1. IEEE, 1997, pp. 11–21.

[41] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," IEEE transactions on fuzzy systems, vol. 13, no. 4, pp. 517–530, 2005.

[42] Z. Pawlak, "Rough sets," International journal of computer & information sciences, vol. 11, pp. 341–356, 1982.

[43] P. Lingras and C. West, "Interval set clustering of web users with rough k-means," Journal of Intelligent Information Systems, vol. 23, pp. 5–16, 2004.

[44] P. Maji and S. K. Pal, "Rough set based generalized fuzzy *c*-means algorithm and quantitative indices," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 6, pp. 1529–1540, 2007.

[45] P. Lingras and G. Peters, "Applying rough set concepts to clustering," Rough Sets : Selected Methods and Applications in Management and Engineering, pp. 23–37, 2012.

[46] G. Shafer, A mathematical theory of evidence. Princeton university press, 1976, vol. 42.

[47] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping." Ann. Math. Statist., vol. 38, pp. 325–339, 1967.

[48] P. Smets and R. Kennes, "The transferable belief model," Classic Works of the Dempster-Shafer Theory of Belief Functions, pp. 693–736, 2008.

[49] R. R. Yager, "On the normalization of fuzzy belief structures," International Journal of Approximate Reasoning, vol. 14, no. 2-3, pp. 127–153, 1996.

[50] V. Antoine, B. Quost, M.-H. Masson, and T. Denoeux, "Cecm : Constrained evidential c-means algorithm," Computational Statistics & Data Analysis, vol. 56, no. 4, pp. 894–914, 2012.

[51] R. N. Dave, "Characterization and detection of noise in clustering," Pattern Recognition Letters, vol. 12, no. 11, pp. 657–664, 1991.

[52] E. Schubert, "Stop using the elbow criterion for k-means and how to choose the number of clusters instead," ACM SIGKDD Explorations Newsletter, vol. 25, no. 1, pp. 36–42, 2023.

[53] D. Pelleg and A. W. Moore, "X-means : Extending k-means with efficient estimation of the number of clusters." in Icml, vol. 1, 2000, pp. 727–734.

[54] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats ?" Pattern Recognition, vol. 93, pp. 95–112, 2019.

[55] S. Kapil and M. Chawla, "Performance evaluation of k-means clustering algorithm with various distance metrics," in 2016 IEEE 1st international conference on power electronics, intelligent control and energy systems (ICPEICES). IEEE, 2016, pp. 1–4.

[56] R. C. De Amorim and B. Mirkin, "Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering," Pattern Recognition, vol. 45, no. 3, pp. 1061–1075, 2012.

[57] N. Gueorguieva, I. Valova, and G. Georgiev, "M&mfcm : fuzzy c-means clustering with mahalanobis and minkowski distance metrics," Procedia computer science, vol. 114, pp. 224–233, 2017.

[58] J. Arora, K. Khatter, and M. Tushir, "Fuzzy c-means clustering strategies : A review of distance measures," Software Engineering : Proceedings of CSI 2015, pp. 153–162, 2019.

[59] P. C. Mahalanobis, "On the generalized distance in statistics." National Institute of Science of India, 1936.

[60] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," Chemometrics and intelligent laboratory systems, vol. 50, no. 1, pp. 1–18, 2000.

[61] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in 1978 IEEE conference on decision and control including the 17th symposium on adaptive processes. IEEE, 1979, pp. 761–766.

[62] H.-C. Liu, B.-C. Jeng, J.-M. Yih, and Y.-K. Yu, "Fuzzy c-means algorithm based on standard mahalanobis distances," in Proceedings. The 2009 International Symposium on Information Processing (ISIP 2009). Citeseer, 2009, p. 422.

[63] A. Rammal, E. Perrin, V. Vrabie, I. Bertrand, and B. Chabbert, "Weighted-covariance factor fuzzy c-means clustering," in 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE). IEEE, 2015, pp. 144–149.

[64] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," IEEE Transactions on pattern analysis and machine intelligence, vol. 11, no. 7, pp. 773–780, 1989.

[65] N. Kumar, H. Kumar, and K. Sharma, "Extension of fcm by introducing new distance metric," SN Applied Sciences, vol. 2, pp. 1–21, 2020.

[66] B. N. Prasad, M. Rathore, G. Gupta, and T. Singh, "Performance measure of hard c-means, fuzzy cmeans and alternative c-means algorithms," International Journal of Computer Science and Information Technologies, vol. 7, no. 2, pp. 878–883, 2016.

[67] D. Krasnov, D. Davis, K. Malott, Y. Chen, X. Shi, and A. Wong, "Fuzzy c-means clustering : A review of applications in breast cancer detection," Entropy, vol. 25, no. 7, p. 1021, 2023.

[68] X. Zhao, Y. Li, and Q. Zhao, "Mahalanobis distance based on fuzzy clustering algorithm for image segmentation," Digital Signal Processing, vol. 43, pp. 8–16, 2015.

[69] P. O. Brown, M. C. Chiang, S. Guo, Y. Jin, C. K. Leung, E. L. Murray, A. G. Pazdor, and A. Cuzzocrea, "Mahalanobis distance based k-means clustering," in International Conference on Big Data Analytics and Knowledge Discovery. Springer, 2022, pp. 256–262.

[70] T. Wang, L. Zhang, and W. Hu, "Bridging deep and multiple kernel learning : A review," Information Fusion, vol. 67, pp. 3–13, 2021.

[71] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural computation, vol. 10, no. 5, pp. 1299–1319, 1998.

[72] Y. Yao, Y. Li, B. Jiang, and H. Chen, "Multiple kernel k-means clustering by selecting representative kernels," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 4983–4996, 2020.

[73] W. M. Rand, "Objective criteria for the evaluation of clustering methods," Journal of the American Statistical association, vol. 66, no. 336, pp. 846–850, 1971.

[74] L. Hubert and P. Arabie, "Comparing partitions," Journal of classification, vol. 2, pp. 193–218, 1985.

[75] D. Steinley, G. Hendrickson, and M. J. Brusco, "A note on maximizing the agreement between partitions : A stepwise optimal algorithm and some properties," Journal of Classification, vol. 32, no. 1, pp. 114–126, 2015.

[76] R. J. Campello, "A fuzzy extension of the rand index and other related indexes for clustering and classification assessment," Pattern Recognition Letters, vol. 28, no. 7, pp. 833–841, 2007.

[77] T. Denoeux, S. Li, and S. Sriboonchitta, "Evaluating and comparing soft partitions : An approach based on dempster–shafer theory," IEEE Transactions on Fuzzy Systems, vol. 26, no. 3, pp. 1231–1244, 2017.

[78] K. Zhou, A. Martin, Q. Pan, and Z.-g. Liu, "Median evidential c-means algorithm and its application to community detection," Knowledge-Based Systems, vol. 74, pp. 69–88, 2015.

[79] R. Quéré, "Quelques propositions pour la comparaison de partitions non strictes," Ph.D. dissertation, Université de La Rochelle, 2012.

[80] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[81] D. L. Davies and D. W. Bouldin, "A cluster separation measure," IEEE transactions on pattern analysis and machine intelligence, no. 2, pp. 224–227, 1979.

[82] H. Qiao and B. Edwards, "A data clustering tool with cluster validity indices," in 2009 International Conference on Computing, Engineering and Information. IEEE, 2009, pp. 303–309.

[83] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 28, no. 3, pp. 301–315, 1998.

[84] J. C. Bezdek, "Numerical taxonomy with fuzzy sets," Journal of mathematical biology, vol. 1, no. 1, pp. 57–71, 1974.

[85] R. N. Dave, "Validating fuzzy partitions obtained through c-shells clustering," Pattern recognition letters, vol. 17, no. 6, pp. 613–623, 1996.

[86] K.-L. Wu and M.-S. Yang, "A cluster validity index for fuzzy clustering," pattern recognition letters, vol. 26, no. 9, pp. 1275–1291, 2005.

[87] H. Mittal and M. Saraswat, "A new fuzzy cluster validity index for hyperellipsoid or hyperspherical shape close clusters with distant centroids," IEEE Transactions on Fuzzy Systems, vol. 29, no. 11, pp. 3249–3258, 2020.

[88] L. Vendramin, M. Naldi, and R. J. G. B. Campello, "Fuzzy clustering algorithms and validity indices for distributed data," Partitional Clustering Algorithms, pp. 147–192, 2015.

[89] R. J. Campello and E. R. Hruschka, "A fuzzy extension of the silhouette width criterion for cluster analysis," Fuzzy Sets and Systems, vol. 157, no. 21, pp. 2858–2875, 2006.

[90] A. Antoniou and W.-S. Lu, Practical optimization : algorithms and engineering applications. Springer, 2007, vol. 19.

[91] D. G. Luenberger, Y. Ye et al., Linear and nonlinear programming. Springer, 1984, vol. 2.

[92] R. J. Vanderbei et al., Linear programming. Springer, 2020.

[93] D. P. Bertsekas, "Nonlinear programming," Journal of the Operational Research Society, vol. 48, no. 3, pp. 334–334, 1997.

[94] G. M. Lee, N. N. Tam, N. D. Yen et al., Quadratic programming and affine variational inequalities : a qualitative study. Springer, 2005.

[95] L. A. Wolsey, Integer programming. John Wiley & Sons, 2020.

[96] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, Genetic programming : an introduction : on the automatic evolution of computer programs and its applications. Morgan Kaufmann Publishers Inc., 1998.

[97] J. Bezdek and R. Hathaway, "Some notes on alternating optimization," Advances in Soft Computing—AFSS 2002, pp. 187–195, 2002.

[98] P. Wolfe, "A duality theorem for non-linear programming," Quarterly of applied mathematics, vol. 19, no. 3, pp. 239–244, 1961.

[99] I. Ekeland and R. Temam, Convex analysis and variational problems. SIAM, 1999.

[100] J. P. Crouzeix, A. Keraghel, and W. Sosa, "Programación matemática diferenciable," Universidad Nacional de Ingenieria, 2011.

[101] H. Maurer and J. Zowe, "First and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems," Mathematical programming, vol. 16, pp. 98–110, 1979.

[102] J.-B. Hiriart-Urruty and C. Lemaréchal, Fundamentals of convex analysis. Springer Science & Business Media, 2004.

[103] K. J. Arrow, L. Hurwicz, H. Uzawa, and H. B. Chenery, Studies in linear and non-linear programming. Stanford University Press, 1958, vol. 2.

[104] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear gauss–seidel method under convex constraints," Operations research letters, vol. 26, no. 3, pp. 127–136, 2000.

[105] Y. Hu and R. J. Hathaway, "On efficiency of optimization in fuzzy c-means." Neuronal Parallel and Scientific Computations, vol. 10, no. 2, pp. 141–156, 2002.

[106] R. Babuka, P. Van der Veen, and U. Kaymak, "Improved covariance estimation for gustafson-kessel clustering," in 2002 IEEE World Congress on Computational Intelligence, vol. 2. IEEE, 2002, pp. 1081–1085.

[107] S. Ghosh and S. K. Dubey, "Comparative analysis of k-means and fuzzy c-means algorithms," International Journal of Advanced Computer Science and Applications, vol. 4, no. 4, 2013.

[108] F. Hoppner and F. Klawonn, "A contribution to convergence theory of fuzzy c-means and derivatives," IEEE Transactions on fuzzy systems, vol. 11, no. 5, pp. 682–694, 2003.

[109] J. C. Bezdek, "A convergence theorem for the fuzzy isodata clustering algorithms," IEEE transactions on pattern analysis and machine intelligence, no. 1, pp. 1–8, 1980.

[110] L. Groll and J. Jakel, "A new convergence proof of fuzzy c-means," IEEE Transactions on Fuzzy Systems, vol. 13, no. 5, pp. 717–720, 2005.

[111] F. Iutzeler and J. Malick, "On the proximal gradient algorithm with alternated inertia," Journal of Optimization Theory and Applications, vol. 176, no. 3, pp. 688–710, 2018.

[112] Y. Nesterov, "Smooth minimization of non-smooth functions," Mathematical programming, vol. 103, pp. 127–152, 2005.

[113] I. Necoara, Y. Nesterov, and F. Glineur, "Linear convergence of first order methods for non-strongly convex optimization," Mathematical Programming, vol. 175, pp. 69–107, 2019.

[114] B. O'donoghue and E. Candes, "Adaptive restart for accelerated gradient schemes," Foundations of computational mathematics, vol. 15, pp. 715–732, 2015.

[115] H. Xiong, Y. Chi, B. Hu, and W. Zhang, "Analytical convergence regions of accelerated gradient descent in nonconvex optimization under regularity condition," Automatica, vol. 113, p. 108715, 2020.

[116] J.-H. Park, A. J. Salgado, and S. M. Wise, "Preconditioned accelerated gradient descent methods for locally lipschitz smooth objectives with applications to the solution of nonlinear pdes," Journal of Scientific Computing, vol. 89, no. 1, p. 17, 2021.

[117] A. Aberdam and A. Beck, "An accelerated coordinate gradient descent algorithm for non-separable composite optimization," Journal of Optimization Theory and Applications, vol. 193, no. 1-3, pp. 219–246, 2022.

[118] T. T. Truong and H.-T. Nguyen, "Backtracking gradient descent method and some applications in large scale optimisation. part 2 : Algorithms and experiments," Applied Mathematics & Optimization, vol. 84, no. 3, pp. 2557–2586, 2021.

[119] M. R. Hestenes, "Multiplier and gradient methods," Journal of optimization theory and applications, vol. 4, no. 5, pp. 303–320, 1969.

[120] M. J. Powell, "A method for nonlinear constraints in minimization problems," Optimization, pp. 283–298, 1969.

[121] D. P. Bertsekas, Constrained optimization and Lagrange multiplier methods. Academic press, 2014.

[122] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends in Machine learning, vol. 3, no. 1, pp. 1–122, 2011.

[123] A. Miele, E. Cragg, R. Iyer, and A. Levy, "Use of the augmented penalty function in mathematical programming problems, part 1," Journal of optimization Theory and Applications, vol. 8, pp. 115–130, 1971.

[124] A. Miele, E. Cragg, and A. Levy, "Use of the augmented penalty function in mathematical programming problems, part 2," Journal of Optimization Theory and Applications, vol. 8, pp. 131–153, 1971.

[125] J. Douglas and H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," Transactions of the American mathematical Society, vol. 82, no. 2, pp. 421–439, 1956.

[126] J. Koko, "Uzawa block relaxation method for the unilateral contact problem," Journal of computational and applied mathematics, vol. 235, no. 8, pp. 2343–2356, 2011.

[127] R. Glowinski, Lectures on numerical methods for non-linear variational problems. Springer Science & Business Media, 2008.

[128] J. Koko, "A survey on dual decomposition methods," SeMA Journal, vol. 62, no. 1, pp. 27–59, 2013.

[129] J. Wang and L. Zhao, "Convergence and applications of admm on the multi-convex problems," in Advances in Knowledge Discovery and Data Mining : 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part II. Springer, 2022, pp. 30–43.

[130] D.-R. Han, "A survey on some recent developments of alternating direction method of multipliers," Journal of the Operations Research Society of China, pp. 1–52, 2022.

[131] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," Journal of Scientific Computing, vol. 78, pp. 29–63, 2019.

[132] M. Yashtini, "Multi-block nonconvex nonsmooth proximal admm : Convergence and rates under kurdyka–łojasiewicz property," Journal of Optimization Theory and Applications, vol. 190, no. 3, pp. 966–998, 2021.

[133] A. Themelis and P. Patrinos, "Douglas–rachford splitting and admm for nonconvex optimization : Tight convergence results," SIAM Journal on Optimization, vol. 30, no. 1, pp. 149–181, 2020.

[134] J. Koko, "Parallel uzawa method for large-scale minimization of partially separable functions," Journal of Optimization Theory and Applications, vol. 158, pp. 172–187, 2013.

[135] M. Fortin and R. Glowinski, Augmented Lagrangian Methods : Applications to the Numerical Solution of Boundary-Value Problems. Amsterdam : North-Holland, 1983.

[136] T. Lin, S. Ma, and S. Zhang, "Global convergence of unmodified 3-block admm for a class of convex minimization problems," Journal of Scientific Computing, vol. 76, no. 1, pp. 69–88, 2018.

[137] B. Wohlberg, "Admm penalty parameter selection by residual balancing," arXiv preprint arXiv :1704.06209, 2017.

[138] R. Glowinski, Numerical Methods for Nonlinear Variational Problems. Berlin, New York : Springer-Verlag, 1984.

[139] R. Glowinski and P. Le Tallec , Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics, ser. Studies in Applied Mathematics. SIAM, 1989.

[140] J. Koko, "Uzawa block relaxation domain decomposition method for the two-body contact problem with Tresca friction," Comput. Methods. Appl. Mech. Engrg., vol. 198, pp. 420–431, 2008.

[141] N. Parikh, S. Boyd et al., "Proximal algorithms," Foundations and trends® in Optimization, vol. 1, no. 3, pp. 127–239, 2014.

[142] J. Koko and S. Jehan-Besson, "An augmented lagrangian method for tv g+ l 1-norm minimization," Journal of Mathematical Imaging and Vision, vol. 38, pp. 182–196, 2010.

[143] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu, "A comprehensive survey of clustering algorithms : State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," Engineering Applications of Artificial Intelligence, vol. 110, p. 104743, 2022.

[144] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in 2010 IEEE international conference on data mining. IEEE, 2010, pp. 911–916.

[145] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity index for crisp and fuzzy clusters," Pattern recognition, vol. 37, no. 3, pp. 487–501, 2004.

[146] E. R. Hruschka, R. J. Campello, A. A. Freitas et al., "A survey of evolutionary algorithms for clustering," IEEE Transactions on systems, man, and cybernetics, Part C (applications and reviews), vol. 39, no. 2, pp. 133–155, 2009.

[147] S. Kullback and R. A. Leibler, "On information and sufficiency," The annals of mathematical statistics, vol. 22, no. 1, pp. 79–86, 1951.

[148] T. Van Erven and P. Harremos, "Rényi divergence and kullback-leibler divergence," IEEE Transactions on Information Theory, vol. 60, no. 7, pp. 3797–3820, 2014.

[149] E. Hellinger, "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen." Journal für die reine und angewandte Mathematik, vol. 1909, no. 136, pp. 210–271, 1909.

[150] V. González-Castro, R. Alaiz-Rodríguez, and E. Alegre, "Class distribution estimation based on the hellinger distance," Information Sciences, vol. 218, pp. 146–164, 2013.

[151] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," Sankhyā : The Indian Journal of Statistics (1933-1960), vol. 7, no. 4, pp. 401–406, 1946.

[152] F. C. Schweppe, "State space evaluation of the bhattacharyya distance between two gaussian processes," Information and Control, vol. 11, no. 3, pp. 352–372, 1967.

[153] L. V. Kantorovich, "Mathematical methods of organizing and planning production," Management science, vol. 6, no. 4, pp. 366–422, 1960.

[154] L. N. Vaserstein, "Markov processes over denumerable products of spaces, describing large systems of automata," Problemy Peredachi Informatsii, vol. 5, no. 3, pp. 64–72, 1969.

[155] M. Soloveitchik, T. Diskin, E. Morin, and A. Wiesel, "Conditional frechet inception distance," arXiv preprint arXiv :2103.11521, 2021.

[156] S. Deng, B. Albert, V. Antoine, and J. Koko, "A specialized xie-beni measure for clustering with adaptive distance," Conference of the European Society for Fuzzy Logic and Technology, pp. 713–724, 2023.

[157] A. J. Rothman, "Positive definite estimators of large covariance matrices," Biometrika, vol. 99, no. 3, pp. 733–740, 2012.

[158] Z. Xu, M. Figueiredo, and T. Goldstein, "Adaptive admm with spectral penalty parameter selection," in Artificial Intelligence and Statistics. PMLR, 2017, pp. 718–727.

[159] B. Albert, V. Antoine, and J. Koko, "Optimisation de fuzzy c-means (fcm) clustering par la méthode des directions alternées (admm)," Extraction et Gestion des Connaissances : Actes de la conférence EGC'2023, vol. 39, 2023.

[160] B. Albert, V. Antoine, and J. Koko, "Optimization of fuzzy c-means with alternating direction method of multipliers," International Conference on Optimization and Learning, pp. 277–286, 2023.

[161] J. Davis, "Combining error ellipses," MIT Kavli Institute (MKI), Tech. Rep., 2007.

[162] K. B. Petersen, M. S. Pedersen et al., "The matrix cookbook," Technical University of Denmark, vol. 7, no. 15, p. 510, 2008.

[163] Y. Fukuyama, "A new method of choosing the number of clusters for fuzzy c-means method," in Proc. 5th Fuzzy System Symp., 1989, pp. 247–250.

[164] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," Applied intelligence, vol. 48, pp. 4743–4759, 2018.

[165] M. Rezaei and P. Fränti, "Can the number of clusters be determined by external indices ?" IEEE Access, vol. 8, pp. 89 239–89 257, 2020.

[166] "Algerian Forest Fires Dataset," UCI Machine Learning Repository, 2019. https://doi.org/10.24432/C5KW4N

[167] "Dry Bean Dataset," UCI Machine Learning Repository, 2020. https://doi.org/10.24432/C50S4B

[168] B. German, "Glass Identification," UCI Machine Learning Repository, 1987. https://doi.org/10.24432/C5WW2P

[169] R. A. Fisher, "Iris," UCI Machine Learning Repository, 1988. https://doi.org/10.24432/C56C76

[170] D. Slate, "Letter Recognition," UCI Machine Learning Repository, 1991. https://doi.org/10.24432/C5ZP40

[171] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. Kowalski, and S. Lukasik, "Seeds," UCI Machine Learning Repository, 2012. https://doi.org/10.24432/C5H30K

[172] W. Wolberg, O. Mangasarian, N. Street, and W. Street, "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1995. https://doi.org/10.24432/C5DW2B

[173] R. Bhatt, "Wireless Indoor Localization," UCI Machine Learning Repository, 2017. https://doi.org/10.24432/C51880

[174] S. Aeberhard and M. Forina, "Wine," UCI Machine Learning Repository, 1991. https://doi.org/10.24432/C5PC7J

[175] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in Proceedings of the twenty-first international conference on Machine learning, 2004, p. 11.