RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mira Abderrahmane de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Clustering : Approche par la théorie des jeux

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Recherche Opérationnelle

HAMIDOUCHE Saddek & IDJERAOUI Tayeb

Président du jury :	M^{me} . TOUCHE Aicha	Maître assistant	Université de Béjaïa.
Examinateurs :	M^{me} . BARACHE Fatiha	Maître Assistant	Université de Béjaïa.
	M^{me} KENDI Salima	Maître Assistant	Université de Béjaïa.
Promoteur :	M^{lle} . BOUCHAMA Kahina	Maître assistant	Université de Béjaïa.
Co-promoteur :	M^{r} . RADJEF M-Said	Professeur	Université de Béjaïa.

Promotion 2013

"Ils ne savaient que c'était impossible,
alors ils l'ont fait".

Mark Twain

"Je m'efforce de tout comprendre
et de ne rien condamner.

Marcel Proust

Dédicaces

Nous dédions ce travail, à nos chère parents qui nous ont soutenus
durant toutes ces années d'études.
À nos frères et sœurs.
À nos familles.
À tous nos amis.

Remerciements

En premier lieu, nous remercions Dieu qui nous a procuré cette réussite. Nous remercions M^{lle} K. BOUCHAMA pour sa disponibilité, sa patience, ses précieux conseils et ses remarques constructives. Nous remercions également M^r M.S. RADJEF pour son aide précieuse en terme de documentation et de disponibilité. Nous remercions M^{me} A.TOUCHE pour avoir accepté de présider le jury. Nous remercions M^{me} F.BARACHE et M^{me} S.KENDI pour avoir accepté de faire part du jury. Nous remercions également tous ceux qui nous ont aidé à la réalisation de ce projet.

Sadek & Tayeb

Département RO
Université de Béjaia
Béjaia, Le 27/06/2013

Table des matières

Table des figures

Introduction générale

La création de groupes au sein d'un ensemble d'éléments est une opération omniprésente dans notre société. Les groupes peuvent posséder plusieurs significations, comme par exemple une signification sociale quand ils décrivent un ensemble de personnes qui partagent des caractéristiques communes. Il est naturel de faire appel aux groupes quand il s'agit de structurer, d'organiser ou de résumer un ensemble d'éléments. Ainsi, dans la vie quotidienne, la notion de groupe est utilisée pour l'organisation et la description, comme par exemple des familles de plantes, des genres de musique, des groupes de produits, des groupes sanguins, etc.

Un groupe peut être défini de manière informelle comme un ensemble d'éléments qui sont rassemblés en raison d'une relation particulière entre ces éléments. La problématique consistant à former de tels groupes de manière automatique se pose dans de nombreux domaines. En marketing, on va chercher à regrouper des personnes ayant des comportements de consommation similaires pour cibler par exemple une campagne publicitaire. Dans l'étude des réseaux sociaux, on cherche à regrouper les différents membres du réseau pour y faire émerger des communautés. En biologie, on cherche à identifier des groupes de gènes ayant le même comportement pour mettre en place des thérapies géniques.

Deux types d'approches existent et sont à considérer quand la tâche est de regrouper des éléments de manière automatique. La première approche consiste à faire émerger des groupes au sein d'un ensemble d'éléments sans aucune information a priori. Dans ce cas, cette tâche est appelée, selon les domaines, classification non supervisée, classification automatique ou encore en utilisant l'anglicisme clustering. Les groupes créés sont appelés clusters. L'objectif de cette approche est de découvrir la structure sous-jacente des données pour en extraire de l'information.

La seconde approche intervient quand les groupes existent a priori, et le problème est de créer un modèle permettant d'assigner des éléments à ces groupes. Dans ce cas, la tâche

est appelée classification supervisée et les groupes sont appelés des classes et possèdent une étiquette qui correspond au nom de la classe. La classification supervisée nécessite cependant, contrairement à la classification non supervisée, un ensemble d'exemples, c'est-à-dire un ensemble d'éléments dont la classe est connue a priori. L'objectif, à partir de ces exemples, est de découvrir un modèle des classes pouvant être généralisé à un ensemble de données plus large sous la forme d'un modèle prédictif.

Ce processus comporte deux étapes : une étape de construction du modèle à partir des exemples dont la classe est connue, suivie d'une étape de classement des objets dont la classe est inconnue. Cependant, les exemples nécessaires à la construction de ce modèle sont très souvent difficiles à obtenir car ils nécessitent généralement l'intervention d'un expert humain, qui va manuellement affecter une classe à un élément, c'est-à-dire lui affecter une étiquette.[6]

Notre objectif dans ce mémoire, est de proposer une technique de classification non supervisée, pour cela nous allons utiliser les techniques de la théorie des jeux afin de créer un modèle. Pour cela, nous allons regrouper les données identiques dans des classes. Les m grandes classes représentent les clusters (les stratégies), et les n classes restantes représentent les joueurs. La distance entre le centre de la classe joueur et le centre d'un cluster influe sur le gain obtenu. Nous cherchons à trouver la meilleure distribution de données dans les classes, où chaque donnée appartient au cluster le plus similaire. Cette distribution est appelée équilibre de Nash qui consiste à trouver une situation où chaque objet n'a pas d'intérêt à changer de cluster. Trouver cet équilibre est un problème d'optimisation combinatoire NP-difficile. Pour cela, nous allons utiliser une heuristique de manière à trouver un équilibre de Nash dans un temps raisonnable.

Pour cela, nous avons organisé notre mémoire en quatre chapitres. Après une introduction, nous commençons naturellement par présenter dans le premier chapitre les notions de base du domaine de clustering. Puis, dans le deuxiéme chapitre, on donne un aperçu sur la théorie des jeux, ensuite on synthétisé quelques travaux sur la combinaison des deux domaines : clustering et la théorie des jeux. Dans le dernier chapitre, nous présentons notre approche, ainsi que les expérimentations menée dans ce cadre, puis nous comparons les résultats de notre approche avec les résultats de quelques algorithmes classiques afin de pouvoir évaluer la performance de notre algorithme. Enfin, on termine par une conclusion et des perspectives.

Chaque chapitre est abordé par une petite introduction qui offre une lecture en diagonal de ce qui est à présenter dans le chapitre, et se termine par une conclusion qui est un bilan de ce qui a été présenté.

1
Le clustering

Introduction

Le clustering est une étude non supervisée, visant à organiser un ensemble d'objets en groupes ou clusters, de façon à avoir des objets similaires groupés et les objets différents organisés dans des groupes différents. Ce problème a été abordé dans de nombreux contextes et par des chercheurs dans beaucoup de disciplines, ce qui reflète son attrait et son utilité comme l'une des étapes les plus importantes de l'analyse exploratoire des données.

Le clustering est loin d'être trivial à réaliser. En fait, le problème est fondamentalement mal posé, c'est à dire un ensemble donné d'objets peuvent être regroupés d'une manière radicalement différente, sans avoir de critères pour préférer un regroupement plutôt qu'un autre.

En raison de l'ambiguïté intrinsèque concernant les problèmes du clustering, une vaste collection d'algorithmes s'est étendue dans la littérature afin d'améliorer les approches existantes sur des applications spécifiques.

Les techniques traditionnelles sont axées sur la notion de caractéristiques. Selon ce point de vue, chaque objet est décrit en termes d'un vecteur d'attributs numériques et

est donc associé à un point dans un espace vectoriel (géométrique) Euclidien de sorte que les distances entre les points observées reflètent les dissimilitudes/similitudes entre les objets respectifs. Cependant, l'inconvénient avec l'approche géométrique est sa limitation intrinsèque, qui porte sur le pouvoir de représentation de la caractéristique vectorielle, à base de descriptions. En fait, il existe de nombreux domaines d'application où soit il n'est pas possible de trouver les caractéristiques satisfaisantes ou ils sont inefficaces pour des objectifs d'apprentissage[4].

Définition 1.1. [10] Le clustering est un processus qui regroupe un ensemble d'objets (physiques ou abstraits) en clusters similaires de telle sorte que les données du même cluster aient des caractéristiques similaires, et celles appartenant à des clusters distincts soient dissimilaires.

1.1 Mesures de distance :

Étant donné que le clustering est basé sur le groupement d'objets semblables, la définition d'une mesure qui peut déterminer si deux objets sont semblables ou différent est nécessaire.

On distingue deux types principaux de mesures employés pour estimer cette relation[13] :
- Mesures de distance.
- Mesures de similitude.

Plusieurs méthodes de clustering utilisent les mesures de distance pour statuer sur la similitude ou la dissimilitude de n'importe quelle paire d'objets. La mesure de distance entre deux objets x_i et x_j est notée d(x_i, x_j). Une telle mesure devrait être symétrique et atteindre sa valeur minimale (habituellement zéro) en cas de vecteurs identiques.[13]

Une mesure de distance est appelée mesure métrique de distance si elle satisfait également les propriétés suivantes [13] :
- Inégalité triangulaire : $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k) \quad \forall x_i, x_j, x_k \in S.$
- $d(x_i, x_j) = 0 \Rightarrow x_i = x_j \quad \forall x_i, x_j \in S.$

Où S représente l'ensemble des objets.

Dans ce qui suit, nous allons passer en revue les principales mesures de distance utilisées en clustering.

1.1.1 Distance de Minkowski :

La distance de Minkowski est utilisée lorsque les attributs des objets sont de type numérique.

Étant donnés deux éléments de dimension p, x_i=(x_{i1}, ...,x_{ip}) et $x_j = (x_{j1}, ..., x_{jp})$, la distance entre les deux éléments peut être calculée par la métrique de Minkowski[10] :

$$d(x_i, x_j) = |x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + ... + |x_{ip} - x_{jp}|^g \tag{1.1}$$

⋆ Pour g=2, on obtient la distance Euclidienne.
⋆ Pour g=1, la mesure est appelée distance de Manhattan.
⋆ Pour $g = \infty$, c'est la métrique de Tchebychev

L'unité de mesure utilisée peut affecter l'analyse des clusters, par conséquent, afin d'éviter la dépendance du choix de l'unité de mesure, on a recours à la standardisation des données. La mesure standardisée tente de donner un poids égal à tous les attributs. Si un poids est assigné à chaque attribut en fonction de son importance, la mesure pondérée de la variable sera alors[13] :

$$d(x_i, x_j) = w_1|x_{i1} - x_{j1}|^g + w_2|x_{i2} - x_{j2}|^g + ... + w_p|x_{ip} - x_{jp}|^g \tag{1.2}$$

La mesure de distance de Minkowski peut être facilement calculée pour des attributs à valuation continue. Dans le cas d'objets décrits par des attributs catégoriques, binaires, ordinaux ou du type mixte, une mesure de distance spécifique doit être définie.

1.1.2 Mesure de distance pour les attributs binaires :

Dans le cas des attributs binaires, la distance entre les objets peut être calculée sur la base de la table de contingence. Un attribut binaire est symétrique si ses deux états ont le même poids. Dans ce cas, on utilise le coefficient d'appariement simple pour évaluer la dissimilitude entre deux objets [10] :

$$d(x_i, x_j) = \frac{r + s}{q + r + s + t} \tag{1.3}$$

Où q est le nombre d'attributs qui sont égaux à 1 pour les deux objets, t est le nombre d'attributs qui sont égaux à 0 pour les deux objets, s et r est le nombre d'attributs qui sont inégaux pour les deux objets.

Un attribut binaire est dit asymétrique si une valeur est moins significative que l'autre, dans ce cas la dissimilarité est calculée en utilisant le coefficient de Jaccard [10] :

$$d(x_i, x_j) = \frac{r + s}{q + r + s} \tag{1.4}$$

1.1.3 Mesure de distance pour des attributs nominaux :

Lorsque les attributs sont nominaux, deux approches peuvent être utilisées :[10]

1. L'appariement simple (Simple Matching) :

$$d(x_i, x_j) = \frac{p - m}{p} \tag{1.5}$$

   Où p est le nombre total d'attributs, et m le nombre de correspondances entre les objets (égalité en valeur des attributs).

2. Créer un attribut binaire pour chaque état de chaque attribut nominal et calculer la dissimilarité comme cité précédemment.

1.1.4 Mesure de distance pour des attributs ordinaux

Lorsque les attributs sont ordinaux, l'ordre des valeurs est significatif. Dans ces cas, les attributs peuvent être traités en tant qu'attributs numériques après leur remplacement par leur rang respectif sur l'intervalle [0.1], ce remplacement est effectué comme suit[10] :

$$z_{i,n} = \frac{r_{i,n} - 1}{M_n - 1} \tag{1.6}$$

Où $z_{i,n}$ est la valeur standardisée de l'attribut a_n de l'objet i. $r_{i,n}$ est cette dernière valeur avant standardisation et M_n la limite supérieure du domaine de l'attribut a_n (la limite inférieure est supposée égale à 1).

1.1.5 Mesure de distance pour des attributs mixtes

Dans ce cas, on peut calculer la distance en combinant les méthodes mentionnées précédemment, et ce en ajoutant le carré de chaque distance trouvée par type d'attribut à

la distance totale, la distance entre les objets x_i et x_j sera alors [10] :

$$d(x_i, x_j) = \frac{\sum_{n=1}^{p} \delta_{ij}^{(n)} d_{ij}^{(n)}}{\sum_{n=1}^{p} \delta_{ij}^{(n)}} \tag{1.7}$$

Où p est le nombre d'attributs, $\delta_{ij}^{(n)} = 0$ si l'une des valeurs est manquante (ie, il n'ya aucune mesure de la variable n pour l'objet i ou l'objet j), ou $x_{in} = x_{jn} = 0$ et la variable n est binaire asymétrique, sinon, $\delta_{ij}^{(n)} = 1$. La contribution de l'attribut n à la distance entre les deux objets est calculée selon le type d'attribut :

- Si l'attribut est binaire ou catégorique, $d_{ij}^{(n)} = 0$ si $x_{in} = x_{jn}$, sinon $d_{ij}^{(n)} = 1$.
- Si l'attribut est continu, $d_{ij}^{(n)} = \frac{|x_{in} - x_{jn}|}{\max_h x_{hn} - \min_h x_{hn}}$, où h parcourt tous les objets non-manquants pour l'attribut n.
- Si l'attribut est ordinal, on calcule d'abord les valeurs standardisées de l'attribut, ces valeurs seront par la suite traitées comme de type continu.

1.2 Foncions de similarité

La fonction de similarité $s(x_i, x_j)$ qui compare deux vecteurs x_i et x_j constitue une alternative aux mesures de distance.[14]

Cette fonction doit être symétrique (ie $s(x_i, x_j) = s(x_j, x_i)$), avoir une valeur élevée lorsque x_i et x_j sont similaires, et atteindre son maximum lorsque les vecteurs sont identiques.

Une fonction de similarité où l'intervalle cible est $[0, 1]$ est appelée fonction dichotomique de similarité, ainsi les méthodes décrites précédemment pour le calcul de la distance dans le cas d'attributs binaires ou nominaux peuvent être considérées comme des fonctions de similarité.

1.2.1 Mesure du cosinus :

Lorsque l'angle entre deux vecteurs est une mesure significative de leur similarité, alors le produit intérieur normalisé peut être une mesure de similarité appropriée [13] :

$$s(x_i, x_j) = \frac{x_i^T x_j}{||x_i||\ ||(x_j||)} \tag{1.8}$$

1.2.2 Mesure de correlation de Pearson :

Elle est définie par [13] :

$$s(x_i, x_j) = \frac{(x_i - \overline{x_i})^T (x_j - \overline{x_j})}{||x_i - \overline{x_i}||||(x_j - \overline{x_j})||} \tag{1.9}$$

Où $\overline{x_i}$ représente la moyenne de x sur toutes les dimensions.

1.2.3 Mesure de Jaccard étendue :

La mesure de Jaccard étendue a été présentée par Strehl et Ghosh en 2000 [13] et elle est définie par :

$$s(x_i, x_j) = \frac{x_i^T x_j}{||x_i||^2||(x_j||^2 - x_i^T x_j} \tag{1.10}$$

1.2.4 Coefficient de mesure matriciel :

Il est similaire à la mesure de Jaccard étendue et est défini par [13] :

$$s(x_i, x_j) = \frac{2x_i^T x_j}{||x_i||^2||(x_j||^2} \tag{1.11}$$

1.3 Approches de clustering

Clustering Dur, Doux et flou :

Le résultat le plus simple et le plus souvent rencontré est le clustering dur (hard clustering). Dans un clustering dur, chaque élément appartient à un et un seul cluster. L'ensemble des données X est divisé en un ensemble de K clusters, $C = \{C_1, ..., C_K\}$ formant une partition de X, c'est à-dire $\bigcup_{i=1}^{K} C_i = X$ et $C_i \bigcap C_j = \emptyset, \forall i \neq j, i, j \in \{1, ..., K\}$

Ce type de résultat est le plus courant et le plus facilement interprétable par l'expert. Cependant il peut être nécessaire de donner plus de flexibilité aux clusters. En effet, il peut arriver que certains objets se distinguent de manière trop significative des autres objets, et leur affecter un cluster peut perturber le processus de clustering. Il arrive que ces objets soient rejetés et qu'aucun cluster ne leur soit affecté dans le résultat final. On parle alors de clustering dur partiel, c'est-à-dire que chaque objet appartient à un ou aucun cluster.

De plus, la frontière entre les clusters peut être difficile à définir, et il arrive que certains objets soient à la frontière de plusieurs clusters. Pour pouvoir refléter ce type d'appartenance, le clustering doux (soft clustering) permet à chaque objet d'appartenir à un ou plusieurs clusters. On peut alors parler de clustering doux partiel si dans le résultat, un élément peut appartenir à aucun, un ou plusieurs clusters.

L'appartenance à plusieurs clusters est cependant difficile à interpréter pour l'expert. En effet, plus les objets vont appartenir à de nombreux clusters, plus le résultat va perdre en précision et va rendre difficile son interprétation. Le clustering flou apporte alors une solution, en permettant à chaque élément d'appartenir à chacun des clusters selon un certain degré d'appartenance. Il est toujours possible de revenir à un clustering dur en sélectionnant pour chaque objet le cluster dont l'appartenance est maximale.[6]

1.4 Méthodes de clustering :

Les méthodes de clustering sont généralement classifiées en quatre catégories majeures [10] :

- Les méthodes hiérarchiques.
- Les méthodes de partitionnement.
- Les méthodes basées sur la densité
- Les méthodes basées sur la grille

1.4.1 Les méthodes hiérarchiques.

Dans un clustering hiérarchique, un cluster peut être divisé en sous clusters, l'ensemble des clusters étant généralement représenté par un arbre. Un objet appartient à une et une seule feuille dans la hiérarchie, mais également à son noeud père, et ainsi de suite jusqu'à la racine. Les méthodes de clustering hiérarchique permettent d'obtenir ce type de résultats. Il existe deux types d'approches de clustering hiérarchique :

• Les approches par agglomération (ou ascendantes).

• Les approches par division (ou descendantes).

Les approches par agglomération (Bottom up approach)

Cette approche commence par des clusters formés d'un seul objet, puis les fusionne successivement jusqu'à ce que le critère d'arrêt soit atteint (Construction de K clusters

par exemple)

Algorithme :

1. Initialement, mettre chaque objet dans son propre cluster ;
2. Parmi tous les clusters courants, sélectionner les deux clusters ayant la plus petite distance ;
3. Remplacer ces deux clusters par un nouveau cluster, formé par la fusion des deux clusters originaux ;
4. Répéter les étapes 2 et 3 jusqu'à atteindre la condition d'arrêt.

Exemple : Chameleon(A Hierarchical Clustering Algorithm Using Dynamic Modelling)[8]

Chameleon est un algorithme hiérarchique de regroupement agglomératif basé sur un modèle dynamique. La caractéristique clé de l'algorithme Chameleon est qu'il calcule à la fois l'inter-connectivité et la proximité afin d'identifier la paire la plus similaire de clusters. Il opère sur un graphe clairsemé dans lequel les noeuds représentent les éléments de données, et des arcs représentant des similarités entre les éléments de données. Ensuite les clusters sont trouvés en deux phases. Au cours de la première phase, Chameleon utilise un algorithme de partitionnement de graphes, et au cours de la deuxième phase, il utilise un algorithme pour trouver les véritables clusters en combinant de manière répétitive les sous clusters.

Les approches par division (Top down appoach)

Cette approche commence par un cluster formé de tous les objets, qui sera ensuite divisé en petits clusters jusqu'à atteindre une condition d'arrêt donnée par l'utilisateur.

Algorithme :

1. Mettre tous les objets dans un seul cluster ;
2. Répéter jusqu'à atteindre la condition d'arrêt :
   (a) Choisir un cluster à diviser ;
   (b) Remplacer le cluster choisi par le sous cluster obtenu.

Avantages du clustering hiérarchique :

⋆ Flexibilité incluse concernant le niveau de la granularité.
⋆ Facilité de manipuler toutes formes de similitude ou de distance.
⋆ Applicabilité à tout type d'attribut.

Inconvénients :

⋆ Imprécision sur les critères d'arrêt.

⋆ La plupart des algorithmes hiérarchique ne revisitent pas les clusters une fois construits en vue de l'amélioration des résultats.

1.4.2 Les méthodes de partitionnement

Les méthodes de partitionnement ont généralement comme résultat un ensemble de M clusters, chaque objet appartenant à un seul cluster. Chaque cluster peut être représenté par un centroïde (représentant du cluster) qui peut être considéré comme une description récapitulative de tous les objets contenus dans le cluster. La forme précise de cette description dépendra du type des objets qui sont groupés.

Au cas où les données à valeurs réelles sont disponibles, la moyenne arithmétique des vecteurs d'attribut pour tous les objets dans un cluster fournit un représentant approprié ; des types alternatifs de centroïdes peuvent être requis dans d'autres cas.

Si le nombre de clusters est élevé, les centroïdes peuvent encore être groupés de manière hiérarchique.

Il existe plusieurs méthodes de clustering par partitionnement, parmi elles on cite :

Méthode du K-Means (Macqueen, 1967) :

L'algorithme du k-means est le plus populaire des algorithmes de clustering, il est utilisé dans des applications aussi bien scientifiques que techniques.

Dans cette méthode, un cluster est représenté par son centroïde qui est une moyenne (habituellement pondérée) des points situés à l'intérieur du cluster, cette approche ne fonctionne convenablement qu'avec les attributs numérique et le résultat final peut être négativement affecté par la présence de bruits.

La somme des écarts entre un point et son centroïde, exprimée avec une mesure appropriée, est utilisée comme fonction objectif. Chaque point est assigné au cluster dont le centroïde est le plus proche.

Algorithme : K-means

1. Sélectionner K points comme centroïdes initiaux ;
2. Former K clusters en assignant chaque point au centroïde le plus proche ;
3. Recalculer le centroïde de chaque cluster nouvellement formé ;
4. Répéter les étapes 2 et 3 jusqu'à ce qu'aucun centroïde ne change.

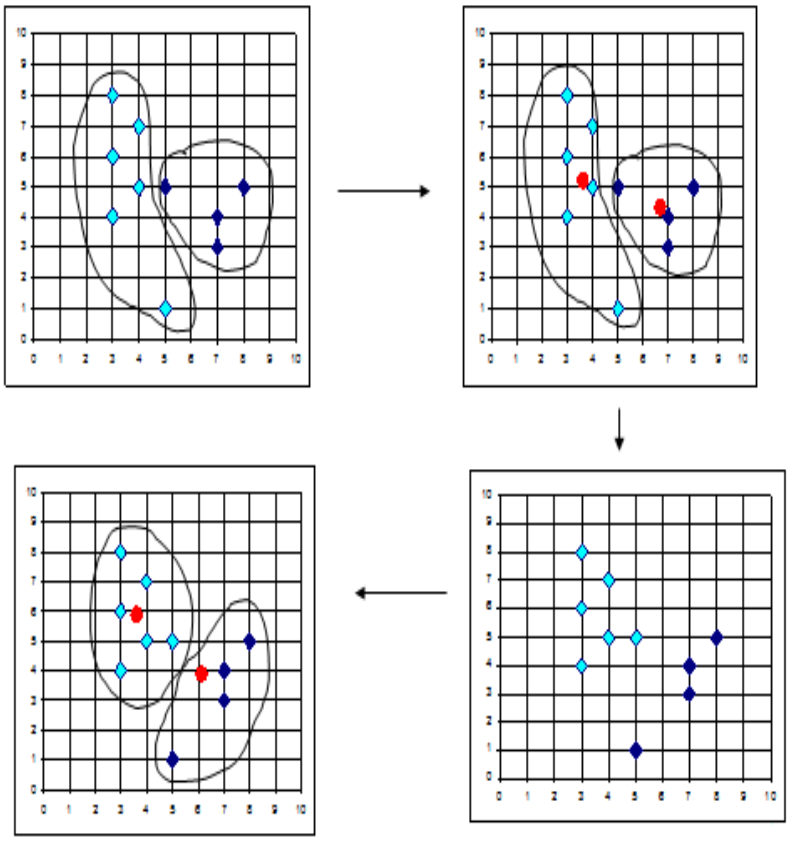

FIGURE 1.1 – Exemple sur l'algorithme K-mean

Avantages :

⋆ Facile à implémenter ;

⋆ Fonctionne avec toutes les mesures standards ;

⋆ Insensible à l'ordre des données.

Inconvénients :

⋆ Il n'est pas applicable en présence d'attributs qui ne sont pas du type numérique
⋆ Le résultat final dépend fortement du choix des centroïdes initiaux
⋆ Ne peut découvrir les groupes non-convexes
⋆ Sensible à la présence de bruits

Méthode des médoïdes :

Le K-médoïde [19] est le point situé à l'intérieur d'un cluster qui peut représenter ce dernier au mieux. La représentation par les K-médoïdes a deux avantages :
• Elle ne dépend pas du type d'attribut.
• Le choix des médoïdes dépend de la concentration des points à l'intérieur d'un cluster, de ce fait, cette méthode est moins sensible à la présence de bruits.

Une fois les médoïdes choisis, les clusters sont définis comme des sous-ensembles de points proches de leurs médoïdes respectifs, et la fonction objectif est définie comme la distance moyenne ou une autre mesure de dissimilarité entre un point et son médoïde.

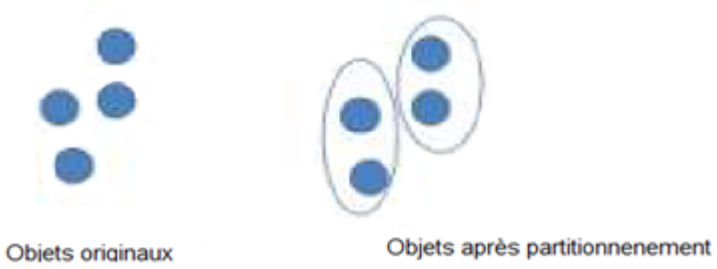

FIGURE 1.2 – Exemple de clustering par partitionnement

1.4.3 Méthodes basée sur la densité :

Les algorithmes basés sur la densité sont capables de découvrir des clusters de formes arbitraires, ce qui assure l'isolement des bruits (outliers) et la prévention contre la formation de clusters non pertinents[19] .

Ces algorithmes groupent des objets selon des fonctions de densité spécifiques. La densité est habituellement définie comme nombre d'objets dans un voisinage particulier des éléments de données. Dans cette approche, un cluster donné continue à augmenter de taille tant que le nombre d'objets dans le voisinage dépasse un certain seuil.

Cette approche se subdivise en deux types :

Méthodes basée sur la densité connective (Density-Based Connectivity Clustering) :

Dans cette technique de clustering, la densité et la connectivité sont mesurées en termes de distribution locale des voisins les plus proches.

La densité-connectivité ainsi définie est une relation symétrique et tous les points accessibles à partir des noyaux des objets peuvent être factorisés dans les composants connectés maximaux servant de clusters. Les points qui ne sont pas connectés à tout point du noyau sont considérés comme des bruits (outliers) et ils ne sont couverts par aucun cluster. Les points non fondamentaux à l'intérieur d'un cluster représentent sa borne. Finalement, les objets du noyau sont les points internes. Le processus est indépendant de l'ordre de données et n'a pas de limitation sur les dimensions ou le type d'attributs.

Méthodes basée sur les fonctions densité (DENSITY FUNCTIONS CLUSTERING) :

Dans cette méthode[19], une fonction de densité est utilisée pour le calcul de la densité. La densité globale est définie comme la somme des fonctions de densité de tous les objets.

Les clusters sont déterminés par les attracteurs de densité qui sont définis comme les maxima locaux de la fonction de densité globale.

Exemple : DBSCAN(Density-Based Spatial Clustering of Applications with Noise)[7]

DBSCAN permet l'identification des clusters de formes arbitraires et le bruit dans une base de donnée spatiale. Cet algorithme requiert seulement deux paramètres d'entrée afin que l'utilisateur puisse spécifier une valeur appropriée. On fixe Eps, le rayon du voisinage à étudier et $MinPts$, le nombre minimum de points qui doivent être contenus dans le voisinage pour considérer la zone comme dense. L'idée clé du clustering basé sur la densité est que pour chaque point d'un cluster, ses voisins pour un rayon donnéEps doit contenir un nombre minimum de points $MinPts$. Ainsi, le cardinal de son voisinage doit dépasser un

certain seuil (considéré comme objet principal). Ensuite, DBSCAN collecte itérativement et de proche en proche les objets atteignables par densité par rapport aux objets principaux, le processus se termine lorsqu'aucun nouveau point ne peut être ajouté à un cluster.

1.4.4 Méthode basée sur les grilles :

Un algorithme de clustering basé sur les grilles [19] utilise des structures de données multi-résolution, où l'espace d'objets est quantifié en un ensemble de cellules, puis identifie l'ensemble de cellules denses connectées pour former des clusters.

1.5 Choix du nombre de cluster

Définir le nombre de clusters[19] est un des problèmes les plus difficiles en clustering. En effet, il est souvent nécessaire de fournir le nombre de clusters souhaité comme paramètre de l'algorithme. Le choix du nombre de clusters a souvent été étudié comme un problème de sélection de modèle. Dans ce cas, l'algorithme est généralement exécuté plusieurs fois indépendamment avec un nombre de clusters différent. Les résultats sont ensuite comparés en se basant sur un critère de sélection qui permet de choisir la meilleure solution. Ce choix est toujours subjectif et fortement dépendant du critère sélectionné pour comparer les résultats.

Deux approches moins subjectives souvent utilisées se basent sur les critères de Minimum Message Length (MML) et Minimum Description Length (MDL). Elles consistent à débuter avec un nombre de clusters relativement élevé, puis à fusionner itérativement deux clusters pour optimiser les critères (MML ou MDL). Les autres critères classiquement utilisés pour la sélection de modèle sont le Bayes Information Criterion (BIC) et le Akiake Information Criterion (AIC). Le Gap statistics est également utilisé pour décider du nombre de clusters. Ces critères reposent généralement sur des bases statistiques fortes et s'appliquent de manière naturelle aux méthodes de clustering probabilistes. Elles peuvent être plus difficiles à mettre en place lors de l'utilisation d'autres types d'approches. De plus, elles sont relativement coûteuses et nécessitent d'effectuer de nombreuses exécutions des algorithmes. L'étude de la validité des clusters découverts peut également être un outil pour effectuer le choix du nombre de clusters. Dans l'idéal, le choix du nombre de clusters reste à l'appréciation de l'expert qui est à même, avec ou sans l'aide d'indices, de choisir le nombre de clusters qui lui paraît adapté.

La section suivante donne un aperçu des techniques de validation du clustering qui permettent, entre autres, d'optimiser le nombre de clusters.

1.6 Techniques de validation du clustering

L'objectif principal de la validation des clusters est d'évaluer les résultats du processus de clustering afin de choisir le meilleur partitionnement des données [12]. Par conséquent, des approches de validation sont utilisées pour évaluer quantitativement le résultat d'un algorithme de clustering. Ces approches possèdent des indices représentatifs, appelés indices de validité.

Deux critères sont largement utilisés dans la mesure de la qualité du clustering des données :

• **Compacité** : Les objets situés dans un cluster doivent être similaires entre eux et différents des objets appartenant aux autres clusters. La variance des objets dans un cluster est un indice de compacité.

• **Séparation** : les clusters doivent être bien séparés entre eux. La distance Euclidienne entre les centroïdes des clusters donne une indication sur le degré de séparation.

Il existe plusieurs indices de validité, parmi eux on cite :

1.6.1 Indice de Dunn

Dunn [5] a proposé un indice permettant d'identifier les clusters compacts et bien séparés. L'objectif principal de l'indice de Dunn est de maximiser les distances inter-clusters (séparation) et de minimiser les distances intra-clusters (augmenter la compacité), il est défini par :

$$D = \min_{k=1,\ldots,K} \left\{ \min_{kk=k+1,\ldots,K;k\neq kk} \left(\frac{dist(C_k, C_{kk})}{\max_{a=1,\ldots,K} diam(C_a)} \right) \right\} \tag{1.12}$$

Où $dist(C_k, C_{kk})$ est une fonction de dissimilarité entre les clusters C_k et C_{kk} définie par :

$$dist(C_k, C_{kk}) = \min_{u\in C_k, w\in C_{kk}} d(u, w) \tag{1.13}$$

$d(u, w)$ étant la distance euclidienne entre u et w.

Une valeur optimale de K (nombre de clusters) est celle qui maximise l'indice de Dunn. Mais cet indice présente deux inconvénients majeurs :
✓ Son calcul est coûteux.
✓ Il est sensible à la présence de bruit.

1.6.2 Indice de Davies et Bouldin

L'objectif de cet indice est de minimiser la similarité moyenne entre chaque cluster et le cluster qui lui est le plus similaire, cet indice est défini par [3] :

$$DB = \frac{1}{K}\sum_{k=1}^{K} \max_{kk=1,...,k\neq kk} \left(\frac{diam(C_k) + diam(C_{kk})}{dist(C_k, C_{kk})}\right) \tag{1.14}$$

Une valeur optimale de K est celle qui minimise DB.

1.6.3 Indice de Turi

Cet indice incorpore une fonction multiplicatrice (qui permet de pénaliser la sélection d'un nombre restreint de clusters) au ratio des distances inter et intra-clusters, il est défini par [12] :

$$V = (c \times N(2,1) + 1) \times \frac{intra}{inter} \tag{1.15}$$

Où c est un paramètre défini par l'utilisateur, $N(2,1)$ est la loi normale de moyenne 2 et de variance 1. Le terme "*intra*" est la moyenne de toutes les distances entre chaque point et le centroïde de son cluster, et il est défini par :

$$intra = \frac{1}{N}\sum_{k=1}^{K}\sum_{\forall u\in C_k} ||u - m_k||^2 \tag{1.16}$$

Ce terme sert à mesurer la compacité des clusters.

Le terme "*inter*" est la distance minimale entre les centroïdes des clusters, il est défini par :

$$inter = \min\{||m_k - m_{kk}||^2\}, \forall k = 1, ..., K-1 \quad et \quad kk = k+1, ..., K. \tag{1.17}$$

Ce terme mesure la séparation des clusters.

La valeur optimale de K est celle qui minimise V.

1.6.4 Indice S_Dbw

Cet indice, introduit par Halkidi et Vazirgiannis [12] mesure la compacité d'un ensemble de données par la variance du cluster, alors que la séparation est mesurée par la densité entre les clusters. Cet indice est défini par [12] :

$$S\_Dbw = scat(K) + Dens\_bw(K) \tag{1.18}$$

Scat(K) est la dispersion moyenne des clusters (mesure de compacité) et elle est donnée par :

$$scat(K) = \frac{1}{K}\sum_{k=1}^{K} ||\delta(C_k)||/||\delta(Z)|| \tag{1.19}$$

Où $\delta(C_k)$ est la variance du cluster C_k et $\delta(Z)$ est la variance de l'ensemble de données Z. $||Z||$ est défini par : $||Z|| = (z^T z)^{1/2}$.

Le terme *Dens_bw(K)* évalue la densité de la zone entre deux clusters par rapport à leur densité, et il est donné par :

$$Dens\_bw(K) = \frac{1}{K(K-1)}\sum_{k=1}^{K}\sum_{kk=1,k\neq kk}^{K}\frac{density(b_{k,kk})}{max\{density(C_k), density(C_{kk})\}} \tag{1.20}$$

Où $b_{k,kk}$ est le point médian du segment défini par m_k et m_{kk}. Le terme *density(b)* est défini par :

$$density(b) = \sum_{ll=1}^{n_{k,kk}} f(y_{ll}, b) \tag{1.21}$$

avec $n_{k,kk}$, le nombre total d'objets dans les clusters C_k et C_{kk}.

La fonction $f(y, b)$ est donnée par :

$$f(y,b) = \begin{cases} 0 & Si \quad d(z,b) > \sigma \\ 1 & sinon \end{cases} \tag{1.22}$$

$$\sigma = \frac{1}{K}\sqrt{\sum_{k=1}^{K} ||\sigma(C_k)||}$$

La valeur optimale de K est celle qui minimise l'indice S_Dbw.

L'inconvénient majeur de cet indice est qu'il ne fonctionne pas pour des clusters de formes arbitraires.

1.7 Domaines d'applications du clustering

Le clustering possède des domaines d'applications extrêmement variés, parmi lesquels :

- **Le Marketing :** segmentation du marché en découvrant des groupes de clients distincts à partir de bases de données d'achats.

- **L'environnement :** identification des zones terrestres similaires dans une base de données contenant des informations (en termes d'utilisation) de la terre.

- **les assurances :** identification de groupes d'assurés distincts associés à un nombre important de déclarations.

- **La planification des villes :** identification de groupes d'habitations suivant leurs type, valeur, localisation géographique.

- **La médecine :** Localisation de tumeurs dans le corps humain.
  Par exemple, dans un nuage de points fournis par le scan du cerveau , on identifie les points définissant une tumeur.

- **La segmentation d'images :** Détection des zones homogènes dans une image.

- **Web log analysis :** Identification de profils d'utilisateurs a travers leur flux de clics (Clickstream).

- **Text mining** Classification des textes selon leur similitude dans des dossiers automatiques.

Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon des principaux concepts et définitions relatifs au problème du clustering. Nous constatons aisément au vu des multiples domaines d'application du clustering et des différents types de données à traiter, qu'aucune méthode ne peut être qualifiée de meilleure par rapport à une autre sur l'ensemble des applications envisageables.

Afin d'aider l'utilisateur dans le choix d'une méthode de résolution, un ensemble de critères a été présenté ainsi qu'une classification des différentes méthodes de clustering existantes.

Le chapitre suivant sera consacré à la présentation de quelques concepts fondamentaux en théorie des jeux qui seront mis en oeuvre ultérieurement dans la résolution du problème de clustering.

2

Rappels sur la théorie des jeux

Introduction

La théorie des jeux cherche à modéliser et analyser des situations d'interaction stratégique ou de concurrence. Elle vise à prédire dans quelle(s) situation(s) doivent se placer un ensemble de partenaires rationnels en interaction, la référence classique qui traite cette discipline est celle de Osbourne et Rubinstein (1994).

Un événement décisif a été la publication du livre intitulé "La Théorie des jeux et comportements économiques " en 1944 par John Von Neumann en collaboration avec Oskar Morgenstern , qui a fixé la terminologie et la présentation des problèmes encore utilisées à ce jour. Vers les années 50, John Nash prouvait que tout jeu possède une situation d'équilibre mixte, dite d'équilibre de Nash, dans lequel aucun joueur n'a intérêt à s'écarter unilatéralement.

La théorie des jeux s'occupe des problèmes d'optimisation interactive. Cette théorie nous aide à analyser et comprendre le comportement des nombreuses parties prenantes qui interagissent dans le processus de prise de décision. C'est un outil d'analyse de situations dans lesquelles les parties s'efforcent à maximiser leurs propres utilités (espérés). Au final, les bénéfices de chaque partie dépendent du vecteur des stratégies choisies par l'ensemble des joueurs.

Les travaux de recherches développés en théorie des jeux peuvent en premier lieu être classés entre jeux coopératifs et jeux non-coopératifs, chaque branche ayant ses propres applications et concepts de solutions.

2.1 Notions de base et définitions

Cette section est dédiée à la présentation des concepts de base et de la terminologie propre à la théorie des jeux.[2]

• **Jeu** : C'est une situation dans laquelle des individus (ou agents) sont en interaction à la recherche d'un gain maximum. Il est donné par le triplet :

$$G = \langle N, (X_i)_{i \in N}, (u_i)_{i \in N} \rangle \tag{2.1}$$

où :
- $N = \{1, \ldots, N\}$ est l'ensemble des joueurs.
- x_i désigne une stratégie du joueur $i \in N$
- $X_i$est l'ensemble des stratégies du joueur $i \in N$
- $u_i(x) \in R$ est la fonction de gain du joueur $i \in N$

• **Stratégie** :Une stratégie est un plan d'actions complet pour chaque joueur spécifiant ce que fera ce dernier à chaque étape du jeu et face à chaque situation pouvant survenir au cours du jeu.dans le cas d'un jeu simultané, les notions de stratégie et d'action coïncident.

• **Jeux finis à N joueurs** : Un jeu est dit fini, si chacun des joueurs a un ensemble fini de stratégies c'est à dire si $|X_i| < \infty, \quad \forall i \in N$

• **Stratégies pures** : Considérons un jeu fini à N joueurs. Une stratégie pure du joueur i est l'action qu'il choisit à chaque fois qu'il est susceptible de jouer, c'est-à-dire, toutes les options possibles qu'a le joueur. On note par X_i, l'ensemble de toutes les stratégies pures du joueurs i avec $i \in N$, et x_i un élément de X_i tel que :$|X_i| = n_i$. On pose $X = \Pi_{i=1}^{n} X_i$ l'ensemble de toutes les issues en stratégies pures du jeu (2.1)

• **Stratégies mixtes** :Les agents peuvent avoir intérêt à agir de manière aléatoire, c'est à dire à choisir une probabilité sur leur ensemble de stratégies pures, appelé stratégie mixte.

Une stratégie mixte pour le joueur $i \in N$ dans un jeu fini à n joueurs est défini par :

$$\Delta_i = \{\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{n_i}) \in R^{n_i}, \sum_{j=1}^{n_i} \alpha_j = 1, \alpha_j \geq 0, j = \overline{1, n_i}\}$$

Avec $n_i = |X_i|$. Dans ce cas, la composante α_i représente la probabilité avec laquelle le joueur i choisira sa stratégie pure $x_i \in X_i$. On note Δ_i l'ensemble des stratégies mixtes pour le joueur i et on pose $\Delta = \Pi_{i=1}^{N} \Delta_i$, l'ensemble de toutes les issues en stratégies mixtes du jeu (2.1) .

2.2 Classification des jeux

Les jeux peuvent apparaître dans plusieurs situations différentes et ont donc plusieurs propriétés différentes à étudier. Pour simplifier l'analyse, on groupe les jeux selon plusieurs critères dont on présente ci-dessous les principales.

2.2.1 Jeux coopératifs /non-coopératifs

La théorie des jeux coopératifs se focalise sur la valeur de la coopération, c'est-à-dire la valeur qu'un ensemble de joueurs peut obtenir en coopérant, sans préciser les actions spécifiques que les joueurs doivent entreprendre afin de créer cette valeur. Les jeux coopératifs modélisent les situations où les joueurs peuvent se grouper en coalitions, les actions des joueurs seront alors menées conjointement de façon à atteindre un objectif commun.

Les jeux non coopératifs modélisent les interactions où les agents sont libres de choisir leurs actions et où un agent rationnel cherche à maximiser son propre bien-être (si un agent se rend compte qu'il a une stratégie admissible b lui permettant d'obtenir une meilleure utilité que celle obtenue avec la stratégie a alors il ne devrait pas jouer a).

2.2.2 Jeux simultanés/séquentiels

Dans un jeu, si les joueurs décident de leurs actions simultanément, alors on parle dans ce cas de jeu simultané. A l'inverse, si les joueurs décident de leur actions l'une après l'autre, alors on est dans le cas d'un jeu séquentiel.

2.2.3 jeux à information complète / incomplète

Un jeu est dit à *information complète* si chacun des joueurs connaît la structure du jeu, c'est-à-dire : l'ensemble des joueurs, les préférences des joueurs, les règles du jeu et le type d'information qu'a chaque moment du jeu chaque joueur possède sur les actions entreprises par les autres joueurs au cours des phases précédentes. Donc, chaque joueur peut se mettre à la place de tous les autres joueurs et du modélisateur. Le jeu du dilemme du prisonnier est à information complète car chacun des prisonniers connaît parfaitement la règle du jeu définie par le policier ainsi que l'utilité de l'autre joueur.

Si au moins un des joueurs ne connaît pas entièrement la structure du jeu, le jeu est dit à *information incomplète.*

2.2.4 Jeux à information parfaite / imparfaite

Un jeu est dit à *information parfaite* si chacun des joueurs, au moment de choisir son action, à une connaissance parfaite de l'ensemble des décisions prises antérieurement par les autres joueurs. Un jeu est à information imparfaite si un des joueurs ne connaît pas, à un moment du déroulement du jeu, ce qu'a joué un autre joueur. Ceci peut arriver dans le cas où on cache l'information aux joueurs ou parce que les joueurs jouent simultanément. Le jeu du dilemme du prisonnier est à *information imparfaite* car les deux joueurs jouent simultanément.

2.2.5 Jeux répétés

Les jeux répétés sont des jeux qui sont joués plus d'une fois. Les joueurs peuvent choisir des actions différentes en considérant l'histoire du jeu étant donné que l'expérience que les joueur ont acquise à travers la répétition est cruciale pour définir leurs actions futures.

2.3 Forme extensive et forme stratégique d'un jeu

2.3.1 Jeu sous forme extensive

Lorsque le jeu est à information complète, chaque joueur connaît toutes les données du problème, pour lui et pour les autres. Toutefois, pour qu'un jeu soit totalement défini, il faut que ses règles précisent l'ordre des coups. Trois types de situations peuvent alors être envisagées :

– soit les joueurs font leurs choix de façon séquentielle, dans un ordre précis fixé à l'avance ;
– soit ils prennent leur décision simultanément ;
– soit ils font face à des situations mixte, avec des coups successifs et des coups simultanés.

Lorsque les règles du jeu stipulent que les joueurs interviennent les uns après les autres, dans un ordre précis et que le nombre d'actions parmi lesquelles leur choix s'exerce est fini, la représentation qui semble la plus appropriée consiste à tracer un arbre (appelé arbre de Kuhn). Une telle représentation est dite sous forme extensive du jeu.

2.3.2 Jeu sous forme stratégique

Lorsque le jeu est à coups simultanés, la représentation par la forme extensive devient particulièrement lourde et compliquée. Pour cela, la modélisation qui apparaît comme la plus appropriée est la forme stratégique, ou normale, qui fait appel à un (ou des) tableau(x) de nombres donnant les gains des joueurs pour chacune des issues possibles, les lignes et les colonnes correspondent aux diverses stratégies. Dans ce contexte, nous supposons que la satisfaction d'un joueur peut être représentée par des nombres réels. Plus le nombre est élevé, plus la satisfaction est importante. Ces préférences sont définies par une fonction d'utilité ou de satisfaction des résultats.

2.4 Concepts de solution

Nous présenterons dans cette section les concepts de solutions les plus utilisés aussi bien dans la branche coopérative que non-coopérative de la théorie des jeux.

2.4.1 Concepts de solution pour les jeux coopératifs

L'unicité d'une solution est toujours une propriété souhaitable et recherchée dans les jeux coopératifs. En général, un jeu coopératif décrit par une fonction caractéristique n'engendre pas une imputation unique, c'est ainsi que différentes procédures sont mises en oeuvre afin d'exclure un certain nombre de solutions et en privilégier d'autres.[16]

Définition 2.1. Formellement,Un jeu coopératif (N, ν) est caractérisé par deux éléments :

- N : Ensemble des joueurs.
- $\nu : 2^N \longrightarrow R$, Fonction caractéristique qui associe à pour chaque coalition $S \subseteqq N$, sa valeur. $S \subseteq N \mapsto \nu(S)$.

Notons le jeu coopératif par :

$$J_c = (N, \nu) \tag{2.2}$$

• **La pré-imputation :**

C'est un vecteur x de R^n tel qu'à chaque joueur $i \in N$ associe un gain vérifiant la condition :

$$\sum_{i \in N} x_i = \nu(N) \tag{2.3}$$

Cette condition exprime la rationalité collective des joueurs, à savoir que si on avait $\sum_{i \in N} x_i < \nu(N)$, ça aurait engendré une perte égale à $(\sum_{i \in N} x_i - \nu(N))$. Inversement, si on avait $(\sum_{i \in N} x_i > \nu(N))$, la solution aurait été non réalisable.

Si de plus, $x_i \geq \nu(\{i\}), \forall i \in N$, on dit que $x = (x_1, \ldots, x_n)$ est une *imputation* pour le jeu (2.2)

Cette condition exprime la rationalité individuelle, c'est à dire le fait qu'aucun joueur n'acceptera de faire partie d'une coalition si son gain en y participant est inférieur au gain qu'il aurait au cas ou il agirait seul.

A ce niveau, nous passons à la présentation des concepts de solution les plus fréquents pour les jeux coopératifs.

•**Le noyau :**

Soit (N, ν) un jeu coopératif à utilité transférable. Soit $x = (x_1, \ldots, x_n)$ le vecteur des paiements des n joueurs. Le noyau du jeu (2.2) est constitué de toutes les allocations $x = (x_1, \ldots, x_n)$ satisfaisant les propriétés suivantes :

1. Rationalité individuelle, ie $x_i \geqslant \nu(i) \quad \forall i \in N$;
2. Rationalité collective : ie $\sum_{i \in N} x_i = \nu(N)$;
3. Rationalité coalitionnelle : ie $\sum_{i \in S} x_i \geqslant \nu(S), \quad \forall S \subseteq N$.

Le nucléole :

Ce concept de solution fût introduit par Schmeidler [18] en 1969. Son but est de minimiser le mécontentement maximal (ou maximiser le gain minimal) des coalitions. Pour définir le nucléole, on doit d'abord définir quelques notions y afférentes :

Définition 2.2. (L'excès) Noté $e(S, y)$ est le montant que gagne une coalition S si elle accepte la répartition des gains y plutôt que de répondre elle même aux besoins de ses membres. c'est à dire :

$$e(S, y) = \nu(S) - y(S)$$

Définition 2.3. (Le vecteur d'excès) noté $\theta(y)$ de dimension $(2^{|N|} - 2)$, est le vecteur dont les composantes sont les excès $e(S, y)$ pour tout $S \subset N$, $S \neq \emptyset$ classés dans l'ordre croissant.On aura donc l'inégalité :

$$\theta_i(y) \leq \theta_j(y), \quad \forall i, j \in N, \quad i < j$$

Définition 2.4. (l'ordre lexicographique) :Soient deux vecteurs de gains $x \in \mathcal{R}^n$ et $y \in \mathcal{R}^n$. Soient $\theta(x)$ et $\theta(y)$ les vecteurs d'excès associés respectivement, ,S'il existe un entier q tel que $\theta_i(y) = \theta_i(x)$ lorsque $(i < q)$ et $\theta_i(y) > \theta_i(x)$ lorsque $(i = q)$ alors on dit que $\theta(y)$ est lexicographiquement supérieur à $\theta(x)$ et on note :

$$\theta(y) >_L \theta(x)$$

Relation entre le nucléole et l'ordre lexicographique :

Le nucléole est défini comme les imputations y qui ont le plus grand vecteur d'excès associé, ie.

$$\theta(y) \geq \theta(x), \forall x \in \{x | x(N) = \nu(N)\}$$

Schmeidler[18] a démontré que le nucléole existe toujours, et qu'il est unique.De plus, si le noyau est non-vide, le nucléole est toujours dans le noyau.

La valeur de Shapley :

Une imputation $\phi = (\phi_1, ..., \phi_n)$ est une valeur de Shapley si elle satisfait les trois axiomes de Shapley [21]qui sont :

1. **Axiome de symétrie :**Pour tout automorphisme π du jeu $< N, \nu >$, $\phi_i(S) = \phi_{\pi_i}(S)$, $\forall i \in N$ où π est une permutation de I dans I.

   Cet axiome signifie que la valeur attribuée au joueur $i \in N$ ne dépend que de sa force stratégique (et non de son libellé).

2. **Axiome d'efficacité :**

   $$\sum_{i \in N} \phi_i(\nu) = \nu(i)$$

   Ce qui signifie que la totalité de la valeur de la coalition est distribuée à l'ensemble de ses joueurs. Cet axiome correspond à l'optimalité au sens de Pareto.

3. **Axiome de linéarité :** si c et d sont deux fonctions caractéristiques en rapport avec le même ensemble I de joueurs, alors :

   $$\phi_i(c + d) = \phi_i(c) + \phi_i(d)$$

Définition 2.5. Pour tout jeu coopératif (N, ν), la valeur de Shapley du joueur $i \in N$ est donnée par :

$$\phi_i(\nu) = \frac{1}{n!} \sum_{i \in S} (|S| - 1)!(n - |S|)![\nu(S) - \nu(S - i)] = \frac{1}{n!} \sum_{\pi \in \Pi} x_i^{\pi} \tag{2.4}$$

Le point de Gately :

La tendance du joueur i à perturber la grande coalition est définie par le ratio suivant [21] :

$$d_i(x) = \sum_{j \neq i} \frac{x_j - \nu(N - i)}{x_i - \nu(i)} \tag{2.5}$$

Si $d_i(x)$ est élevé, le joueur i pourrait perdre en désertant la grande coalition, mais les autres joueurs perdront encore plus.

Le point de Gately d'un jeu est l'imputation qui minimise la tendance maximale à perturber.

La manière générale de minimiser la tendance maximale à perturber est de mettre toutes les tendances à la perturbation à égalité.

Lorsque le jeu est normalisé, ie $\nu(i) = 0, \forall i \in N$, la manière de rendre tous les $d_i(x)$ égaux est de choisir x proportionnellement à $\nu(N) - \nu(N-i)$, ie :

$$G_{\nu_i} = \frac{\nu(N) - \nu(N-i)}{\sum\limits_{j \in N} \nu(N) - \nu(N-j)} \tag{2.6}$$

La τ-valeur :

On définit pour chaque joueur $i \in N$, les quantités suivantes :

$$M_i(\nu) = \nu(N) - \nu(N-i) \quad et \quad m_i(\nu) = \nu(i) \tag{2.7}$$

Alors, la $\tau - valeur$ sélectionne l'allocation maximale réalisable sur la ligne liant $M(\nu) = (M_i(\nu))_{i \in N}$ et $m(\nu) = (m_i(\nu))_{i \in N}$[22].

Pour chaque jeu convexe (N, ν) :

$$\tau(\nu) = \lambda M(\nu) + (1 - \lambda) m(\nu) \tag{2.8}$$

Où $\lambda \in [0, 1]$ est choisi de telle sorte à satisfaire :

$$\sum_{i \in N} [\lambda(\nu(N) - \nu(N-i)) + (1-\lambda)\nu(i)] = \nu(N) \tag{2.9}$$

2.4.2 Jeux non coopératifs

L'équilibre non coopératif, dit aussi équilibre de Nash, est basé sur le principe de rationalité individuelle. Il s'agit d'un état dans lequel aucun joueur ne souhaite modifier sa stratégie si les autres joueurs maintiennent leurs stratégies d'équilibre.

Equilibre de Nash en stratégies pures

Définition 2.6. Un profil de stratégies $x^* = (x_i^*)_{i \in N} \in X$ est un équilibre de Nash du jeu sous forme stratégique (2.1) si et seulement si :

$$\forall x_i \in X_i : u_i(x_i^*, x_{-i}^*) \geq u_i(x_i, x_{-i}^*), \quad \forall i \in N, \tag{2.10}$$

L'équilibre est dit strict en cas d'inégalité stricte.

Interprétation

La relation (2.10) signifie qu'aucun joueur $i \in N$ ne peut bénéficier d'une déviation unilatérale, et ce quelle que soit la stratégie qu'il choisit dans son ensemble X_i.

Conditions d'existence de l'équilibre de Nash

Nous énonçons dans ce qui suit les conditions d'existence d'un équilibre de Nash du le jeu (2.1).

Théorème 2.1. *Si pour tout $i \in N$, les conditions suivantes sont vérifiées*

1. *Les ensembles X_i sont convexes et compacts ;*
2. *Les fonctions $u_i(.) : X \to R$ sont continues ;*
3. *Les fonctions $u_i(., x^{-i}) : X^i \to R$ sont concaves, $\forall x_{-i} \in X_{-i}$,*

alors le jeu (2.1) admet un équilibre de Nash.

Équilibre de Nash en stratégies mixtes

Définition 2.7. Soit un jeu fini à N joueurs. Une situation :

$$\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_N^*) \in \Delta = \Pi_{i=1}^N \Delta_i$$

est un équilibre de Nash en stratégies mixtes, si on a :

$$\forall \beta_i \in \Delta_i, \quad u_i(\alpha_i^*, \alpha_{-i}^*) \geq u_i(\beta_i, \alpha_{-i}^*), \quad \forall i \in N,$$

Proposition 2.1. *Tout équilibre de Nash en stratégies pures est aussi un équilibre de Nash en stratégies mixtes.*

Théorème 2.2. *Tout jeu fini à n joueurs, admet au moins un équilibre de Nash, éventuellement en stratégies mixtes.*

Conclusion

La théorie des jeux fournit le cadre général de modélisation de l'interaction. Son développement a conduit à l'élaboration de nouveaux modèles, augmentant ainsi son champ d'intervention. Le clustering entre dans cette ligne de développement. Dans le chapitre suivant nous allons présenter quelques travaux modélisant le probléme du clustering sous forme d'un jeu.

3

Le clustering par la théorie des jeux

Introduction

L'extraction des connaissances à partir des données pose une difficulté majeure vue l'augmentation exponentielle des données issues de différentes applications. Plus on possède de données, plus il est difficile de les traiter et d'en tirer des conclusions. Donc, le besoin de valoriser les données nécessite le développement de nouvelles approches car les approches traditionnelles ne sont plus adaptées à cause de la taille et de la complexité des données.

Ces dernières années, la théorie des jeux avec ses différentes branches, a constitué une boite à outil servant de support au traitement du problème de clustering sous un angle nouveau, ce qui a fourni des résultats satisfaisants.

Dans ce chapitre, on présentera quelques travaux traitant le problème de clustering du point de vue de la théorie des jeux.

3.1 Clustering par la théorie des jeux coopératifs

La théorie des jeux coopératifs joue un rôle important dans l'approche du problème de clustering. Plusieurs auteurs se sont intéressés à cette approche, mais dans cette section, nous avons choisi de présenter les résultats de (Swapnil Dhamal, Satyanath Bhat, Anoop K R, et Varun R Embar), car non seulement ils ont proposé un nouvel algorithme, mais aussi, ils ont lié les concepts de solutions de la théorie des jeux coopératifs à la solution du problème de clustering.

Un des problèmes majeurs du clustering est la détermination du nombre k de clusters, la théorie des jeux coopératifs, à travers ses différents concepts de solutions, offre une nouvelle approche de ce problème.

Il sera justifié, dans ce qui suit, l'usage des concepts de solutions la théorie des jeux dans la résolution des problèmes de clustering, et un algorithme basé sur la valeur de Shapley sera donné et comparé à certains algorithmes classiques, l'algorithme SHARPC basé lui aussi sur la valeur de Shapley est utilisé pour l'initialisation du K-means.

Or, le K-means souffre de certaines limitations surtout lorsque les classes ont des variances inégales ou lorsqu'elles ne sont pas convexes.

Comme il sera expliqué ci-dessous, la valeur de Shapley est basée sur l'équité, le point de Gately est basé sur la stabilité, la τ-valeur sur l'efficience et le nucléole sur l'équité min-max et la stabilité. Ainsi ces caractéristiques, en particulier l'équité min-max et la stabilité dont jouit le nucléole sont adaptées dans les problèmes de clustering, mais il est démontré aussi que les solutions trouvées par ces différents concepts de solutions coincident pour une certaine fonction caractéristique. Comme le calcul du nucléole est coûteux en temps de calcul, les autres ont fait appel à la valeur de Shapley pour l'élaboration de l'algorithme de clustering.

3.1.1 Modèle et algorithme de clustering basé sur la théorie des jeux coopératifs :

Le jeu coopératif de clustering est définie par le couple (N, ν) où

- N : Ensemble des données à mettre dans des clusters.
- ν : la fonction caractéristique définie par :

$$\nu(S) = \frac{1}{2} \sum_{i,j \in S; i \neq j} f(d(i,j)) \tag{3.1}$$

Où d est la distance euclidienne, et $f : d \longrightarrow [0, 1]$ une fonction de similarité.

Intuitivement, si deux points i et j ont une petite distance euclidienne, alors $f(d(i, j))$ tend vers 1. On utilisera comme fonction de similarité la fonction f suivante :

$$f(d(i, j)) = 1 - \frac{d(i, j)}{d_M} \tag{3.2}$$

Où d_M est la distance maximale entre toutes les paires de point de l'ensemble de données.

En utilisant la fonction caractéristique donnée ci-dessus, il est démontré que la valeur de Shapley peut être calculée en un temps polynomial et est donnée par :

$$\phi_i = \frac{1}{2} \sum_{j \in N; j \neq i} f(d(i, j)) \tag{3.3}$$

A partir de la formule de l'équation caractéristique, on peut déduire que :

$$\nu(S) = \sum_{T \subseteq S; |T|=2} \nu(T)$$

Après avoir démontré l'équivalence entre les différents concepts de solution définis précédemment [15], et vu la simplicité du calcul de la valeur de Shapley, Swapnil et al. se sont basé sur ce concept de solution pour proposer un algorithme qui exploite la similitude entre la valeur de Shapley des éléments d'une coalition (cluster) et la densité de ce cluster (voir les équations 3.2 et 3.3).

Principe de l'algorithme proposé :

L'algorithme proposé prend en entrée une base de donnée, un seuil maximal de similitude noté $\delta \in [0, 1]$, et un seuil de multiplicité pour la valeur de Shapley noté $\gamma \in [0, 1]$.

D'après les équations **(3.2)** et **(3.3)**, la valeur de Shapley représente dans un certain sens la densité. Pour chaque cluster, on démarre avec un point non-alloué ayant la valeur de Shapley maximale et on l'assigne comme le centre du cluster, si ce point a une forte densité autour de lui, on considère alors seulement les points les plus proches, sinon on considérera aussi des points plus éloignés. Cette idée est implémentée à travers le paramètre β. Pour le point ayant la valeur de Shapley maximale, $\beta = \delta$, pour les autres centres de clusters . La valeur de β diminue de manière non-linéaire. Les valeurs de γ et δ doivent être modifiée en conséquence.

En second lieu, si la densité autour d'un point est très faible comparativement à la densité autour du centre du cluster auquel il appartient, ce point ne sera pas intégré

dans l'extension de ce cluster. Ce qui assure la non-concaténation de deux clusters reliés par un fin pont de points. Ce qui assure également que la densité à l'intérieur d'un cluster ne varie pas au-delà d'une certaine limite. Cette idée est implémentée à travers la notion de file d'extension. Les points sont ajoutés à la file d'extension seulement si leurs valeurs de Shapley est au moins γ-multiple de celle du centre du cluster dont ils font partie. La file d'extension permet la croissance du cluster qui s'arrête dès que la file est vide.

L'énoncé de l'algorithme est le suivant :

Algorithme DRAC (Density-restricted Agglomerative Clustering)

Entrée : Ensemble de données, seuil de similarité maximal $\delta \in [0, 1]$,Seuil de multiplicité pour la valeur de Shapley $\lambda \in [0, 1]$

1. Pour chaque point i, calculer la similarité entre chaque paire de points de l'ensemble des données.
2. Pour chaque point i, calculer sa valeur de Shapley avec les équations (3.2) et (3.3)
3. Ordonner les points dans l'ordre non-croissant de leurs valeurs de Shapley. Soit g_M le maximum global des valeurs de Shapley. Construire une nouvelle file qu'on appellera la file d'extension.
4. Construire un nouveau cluster. De tous les points non-alloués, choisir le point ayant la valeur de Shapley maximale comme le nouveau centre du cluster, soit l_M sa valeur de Shapley. Marquer ce point comme étant alloué et l'ajouter à la file d'extension.
5. Poser $\beta = \delta\sqrt{\frac{l_M}{g_M}}$.
6. Pour chaque point non-alloué, si la similarité de ce point avec le premier point de la file d'extension est au moins égale à β, l'ajouter au cluster actuel et le marquer comme alloué. Si la valeur de Shapley de ce point est au moins γ-multiple de l_m, l'ajouter à la file d'extension.
7. Enlever le premier point de la file d'extension
8. Si la file d'extension n'est pas vide, aller à l'étape 6
9. Si le centre du cluster est l'unique point dans son cluster, le marquer comme bruit.
10. Si tous les points sont alloués à un cluster, terminer, sinon aller à l'étape 4.

Résultats expérimentaux

Dans cette section, l'algorithme DRAC est comparé qualitativement avec certains algorithmes classiques, à savoir l'algorithme agglomératif, OPTICS (Ordering Points To Identify the Clustering Structure) et DBSCAN (Density-Based Spatial Clustering of Applications with Noise). SHARPC donne une bonne initialisation au K-means en utilisant un concept de solution de la théorie des jeux, à savoir la valeur de Shapley.

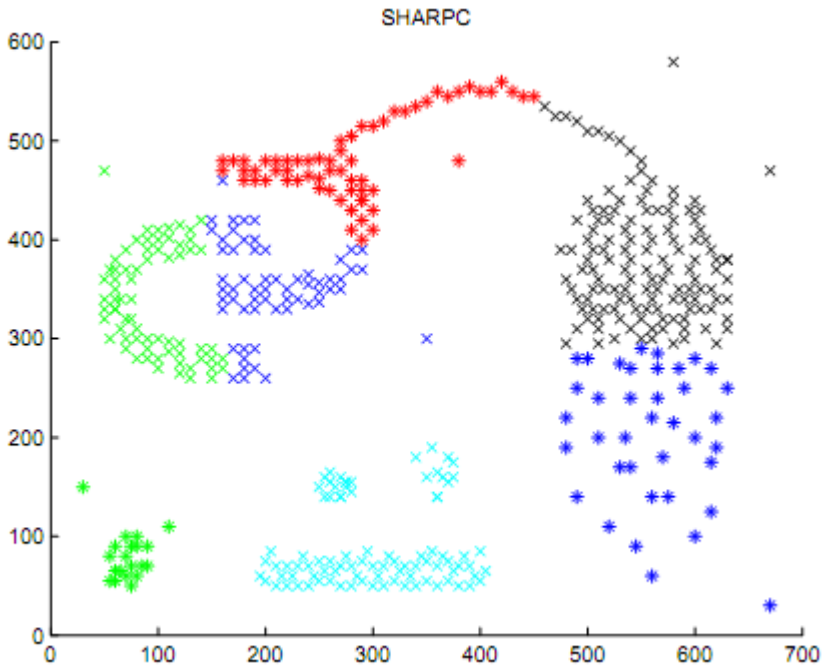

FIGURE 3.1 – Clusters découverts par SHARPC

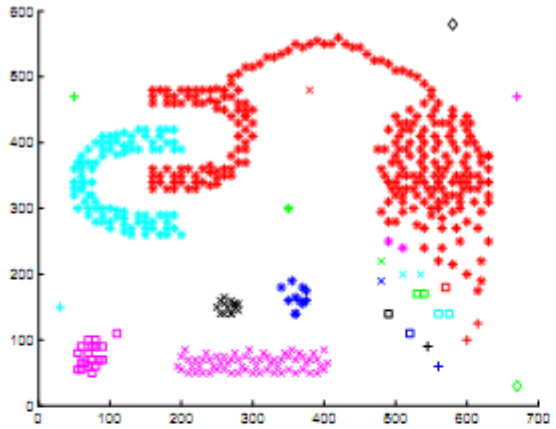

Fig 2. Clusters découverts par le clustering agglomératif

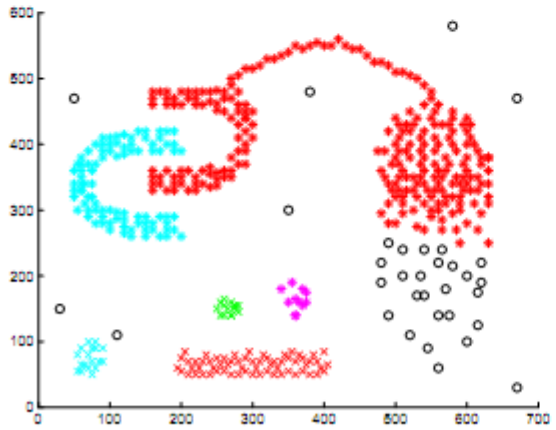

Fig 3. Clusters découverts par DBSCAN

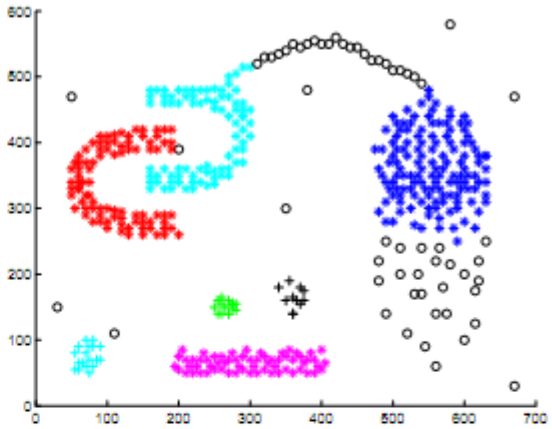

Fig 4. Clusters découverts par OPTICS

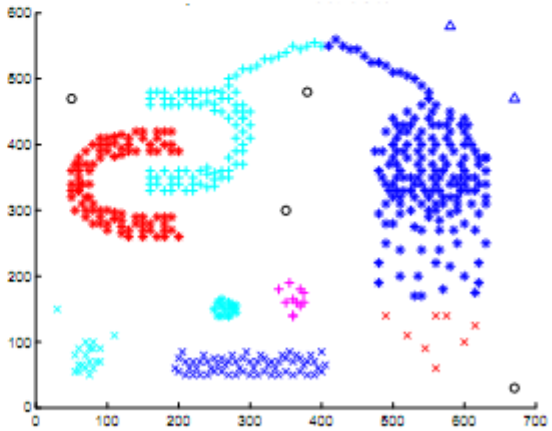

Fig 5. Clusters découverts par DRAC

Analyse des résultats et Commentaires :

⋆ La figure 3.1 montre les clusters découverts par l'algorithme SHARPC . Cet algorithme ne peut détecter les clusters qui ne sont pas convexe. Ainsi, Le cluster (x) est la fusion de trois clusters différents. Si le seuil est augmenté de façon à résoudre le second problème, plus de clusters seront formés et les plus grands clusters se subdivisent en plusieurs petits clusters.

⋆ Le clustering agglomératif(classification ascendante), comme le montre la figure 3.2, peut détecter des clusters de toute forme et taille. Mais vu le seuil constant d'évolution de tous les clusters, il fait face au problème de former plusieurs clusters dans la partie inférieure droite alors qu'ils devraient faire partie d'un seul cluster. Si le seuil est diminué afin de résoudre ce problème, les clusters (*) et (*) seront fusionnées. Un autre problème est que la passerelle connectant les deux classes les fusionne en un seul cluster (*).

⋆ La figure 3.3 montre les résultats de l'algorithme DBSCAN. Il est bien connu que cet algorithme ne peut pas, en général, détecter des clusters de différentes densités. Les points de la partie inférieure en bas à droite sont détectés comme des bruits alors que la région est assez dense pour être classé comme un cluster. Si le seuil est diminué afin de résoudre ce problème, les clusters (*) et (*) seront fusionnées. Un autre problème est que le pont reliant les deux classes les fusionne en un seule cluster(*). La tentative de faire la classification requise mène à des subdivisions inutiles de la classe la plus à droite et entraîne aussi l'augmentation du nombre de bruits détectés.

⋆ Le clustering obtenu grâce à OPTICS est affiché dans la Figure 3.4. Contrairement au DBSCAN, les clusters (*) et (*) sont détectées comme distincts. Toutefois, les points de la partie inférieure droite sont détectés comme des bruits quand ils auraient dû être classés comme un seul cluster.

⋆ La figure 3.5 montre le clustering obtenue par le clustering agglomératif à densité restreinte (DRAC). Comme le cluster (+) est très dense, son centre a une valeur de Shapley très élevée, résultant d'une valeur très élevée du seuil de similitude β. Aucun point du cluster (*) ne dépasse le seuil de similarité requis avec les points du cluster (+) ce qui assure la non-fusion des deux clusters. Les points du centre du pont ont une très faible valeur de Shapley comparativement au centre du cluster (+). Ainsi, ces points n'ont pu dépasser le seuil de la valeur de Shapley en étant au moins γ-multiple de la valeur de

Shapley du centre du cluster. Ceci assure qu'ils ne seront pas ajoutés à la file d'extension et ainsi éviter que le cluster (+) ne soit étendu au cluster (*).

Mais dans l'ensemble, les résultats obtenus par cet algorithme sont plus satisfaisants que les autres algorithmes, car comme on peut le constater sur la figure 5, les clusters séparés par un pont de points ne sont pas fusionnés, ce qui démontre l'efficacité de cette approche.

3.2 Amélioration de l'algorithme du K-means par la théorie des jeux [17]

Dans ce qui suit, on s'intéresse à la tâche de clustering de données numériques en data mining. Une formulation d'un nouvel algorithme de clustering par partitionnement a été introduite. Le processus de clustering est modélisé comme un jeu coopératif sous forme stratégique, de telle sorte que l'on peut trouver efficacement, les motifs qui sont plus proches d'un prototype donné.

L'algorithme a été implémenté et expérimenté sur plusieurs jeux de données artificielles et également sur des ensembles de données issus du monde réel. Les résultats expérimentaux montrent que l'algorithme a de bonnes capacités prédictives. De plus, il est capable de fournir une description intelligible de la solution découverte, du fait que les fonctions mises en oeuvre sont basées sur le calcul des erreurs permettant de suivre à la fois la cohérence interne et l'hétérogénéité externe au niveau des clusters produits.

3.2.1 Le modèle proposé

La définition suivante, nous permet de bien saisir les éléments du modèle proposé.

Définition du jeu sous forme stratégique

Définition 3.1. Un jeu coopératif sous forme stratégique est un triplet :

$$\langle D, \{X_s\}_{\phi \neq S \subseteq D}, \{f_i\}_{i \in D} \rangle$$

Avec les propriétés suivantes :

1. D est un ensemble fini non vide de joueurs, avec $|D| = N$;

2. A toute coalition $\phi \neq S \subseteq D$ on associe un ensemble non vide X_S contenant les stratégies de la coalition S ;
3. Si $S \neq \phi$, $T \subseteq D$, avec $S \cap T = \emptyset$, alors $X_{S \cup T} \supseteq X_S \times X_T$
4. On associe à tout joueur $i \in D$ une fonction de gain définie par :

$$f_i : X = \prod_{l=1}^{N} X_l \to \mathbb{R}$$

Dans le modéle à présenter, ces éléments correspondent à :

- D ; est l'ensemble des n objets de la base. Chaque objet "i" est caractérisé par un ensemble d'attributs $\{t_{i,1}, t_{i,2}, \ldots, t_{i,\delta}\}$;
- X_i ; est l'ensemble des stratégies d'un joueur , elle peut être : 'Liberer()', 'Recruter ()', 'Virer ()', 'Reactiver ()', ou 'Ne rien faire ()' ;
- $\lambda(i, C_j)$: Pour chaque joueur on définit une fonction de gain qui représente le capital du joueur. L'objectif est de la minimiser. Un objet qui fait partie d'une équipe aura au cours du jeu un capital relatif à son équipe, le mieux est d'avoir un capital minimal c'est à dire une moyenne minimale des distances entre cet objet et les membres de son équipe.

3.2.2 Présentation du modéle

Dans un jeu coopératif, les joueurs peuvent former des coalitions, qui sont des parties non vides, dans le but d'améliorer les gains de leurs membres. Pour définir bien le jeu on doit non seulement définir les stratégies dont dispose chaque joueur, mais également le gain apporté pour une coalition donnée.

On considère les objets comme des joueurs. On distinguera deux types de joueurs : les joueurs actifs et les joueurs passifs. Il y'a autant de joueurs actifs que de clusters. Chaque cluster est doté d'un représentant qui permet de le caractériser, ce représentant correspond à un joueur actif. Les autres éléments du cluster correspondent aux joueurs passifs.

Le voisinage d'un objet i peut être défini en se basant sur un seuil, par exemple le nombre maximum de voisins qui sont plus proches de l'objet 'i'. Ce paramètre est fixé de manière subjective pour nous aider dans la sélection des joueurs passifs initiaux. Le voisinage est déterminé à la base des distances qui séparent un objet donné des autres objets. La distance entre deux objets i et j est donnée en fonction de leurs attributs, elle

est notée $d(i;j)$.

• **Description des clusters :** Considérons un nombre K de clusters, qui sont dénotés par $C = C_1, C_2, \ldots, C_K$. Soit $D = \{1, 2, \ldots, n\}$ un ensemble de "n" objets. Chaque joueur i est caractérisé par des attributs $t_{i,1}, t_{i,2}, \ldots, t_{i,\delta}$, où δ est la dimension du vecteur caractéristique. Ces attributs seront utilisés pour définir la mesure de voisinage d'un objet.

Soit $P = \{c_j / j = \overline{1, K}\}$, c_j est le représentant du cluster C_j. Celui-ci est suffisamment représentatif de son équipe. La distance entre deux clusters est exprimée par la distance qui sépare leurs représentants respectifs.

• **Les stratégies :** Chaque joueur doit choisir une des stratégies suivantes :

1. '*Libérer un joueur*' : cela signifie qu'il y a deux équipes dont l'une d'elle accepte de libérer joueur et l'autre de le recruter, le joueur concerné par cet échange est celui ayant un capital minimal ;
2. '*Recruter un joueur*' : cela signifie qu'il y a au moins deux équipes dont l'une d'elle accepte de recruter un joueur et l'autre de le libérer, le joueur concerné par cet échange est celui ayant un capital minimal ;
3. '*Virer un joueur*' : cela signifie que le joueur est viré de sa propre équipe et ne peut pas être admis par aucune des autres équipes, car l'ajout de cet objet (joueur) n'améliore pas leur homogénéité. Ainsi, l'objet est mis dans une équipe additionnelle portant l'étiquette 'Corbeille' ;
4. '*Réactiver un joueur*' : cela signifie qu'une équipe a décidé de prendre l'un des joueurs de l'équipe 'Corbeille', car il améliore son homogénéité ;
5. '*Ne rien faire*' : l'équipe est satisfaite des joueurs dont elle dispose.

• **Le capital d'un objet :**

Le capital d'un objet de données i par rapport à un cluster C_j est donné par :

$$\lambda(i, C_j) = \frac{1}{|C_j|} \sum_{l \in C_j} d(i, l)$$

L'objet ayant le plus petit capital est séléctionné pour etre le représentant du cluster

• **La fonction d'homogénéité :**

Un des problèmes à étudier est celui de l'évaluation du résultat d'un clustering dynamique. C'est-à-dire définir des critères sur le résultat qui permettent de répondre à la question suivante : le résultat du clustering est-il bon ou mauvais par rapport à un jeu de données ?

Les joueurs reçoivent une information incomplète sur la structure du jeu qui ouvre simplement la possibilité aux joueurs d'ajuster leur comportement en fonction d'informations qu'ils peuvent accumuler à partir de l'historique sur les choix des autres joueurs. On définit alors une fonction caractéristique notée par $E : C_j \in C \rightarrow \mathbb{R}^d$, qui, pour chaque sous ensemble de C, associe un vecteur noté $\sigma(C)$. Ce vecteur interprète comment les membres de C sont regroupés et interagissent ensemble.

Le jeu coopératif J sera alors représenté par le quadruplet :[17]

$$J = \langle D, \{X_i\}_{i \in D}, \{\lambda(i, C_j)\}_{i \in D, Cj \in C}, \{E(C_j)\}_{C_j \in C} \rangle$$

Où $E(C_j)$ est la fonction qui calcule l'homogénéité au niveau de chaque équipe ;

L'erreur permet de caractériser la répartition d'un ensemble de données autour de la moyenne, et plus les données sont largement distribuées. Plus l'erreur est élevée. La mesure d'homogénéité d'un cluster peut être calculée par l'erreur locale.

Le vecteur "erreur locale" d'un attribut d'indice i noté par t_i au niveau d'un cluster C_j , ayant c_j comme représentant est noté par $\sigma(C_j)$. Il est calculé comme suit :

$$\sigma_i(C_j) = \frac{1}{|C_j|} \sqrt{\sum_{k=1}^{|C_j|} (t_{ki} - t_{c_j i})^2}$$

l'erreur locale du cluster C_j est alors :

$$E(C_j) = \sigma(C_j) = \{\sigma_1(C_j), \ldots, \sigma_\delta(C_j)\}$$

L'erreur globale est calculée de la manière suivante :

$$E_g = \sum_{i=1}^{\delta} (\max_i - \min_i)$$

avec :

$$\max_i = \max_{j=1,k}(\sigma_i(C_j)); \qquad i = \overline{1,\delta};$$

et :

$$\min_i = \min_{j=1,k}(\sigma_i(C_j)); \qquad i = \overline{1,\delta}$$

.

3.2.3 Algorithme Proposé :[17]

1. **Entrée :** D ; la base de données. K : Le nombre de clusters à former.

   //Le nombre K est choisi par l'utilisateur soit sur la base de la connaissance à priori de son problème, soit d'une manière arbitraire, $K \ll n$.

   MaxIter : Le nombre maximum d'itérations ;
   SeuilMin : Le nombre minimum de points dans un cluster ;
   MaxVois : Le nombre de voisins d'un objet de données.
2. **Sortie :**l'ensemble des K équipes ;
3. Calculer les voisins de taille $MaxVois$ de chaque objet i notée : $Vois(i)$;
4. Choisir aléatoirement de l'ensemble P des K points de la base D à utiliser comme représentants, telque

$$P = \bigcup_{j=1}^{k} c_j, \qquad avec \quad Vois(c_i) \cap Vois(c_j) = \emptyset, \quad \forall i \neq j, \quad i, j \in \{1, \ldots, K\},$$

5. Affecter les objets aux clusters les plus proches en terme de distance ;
6. Calculer le capital de chaque objet dans son cluster
7. Dans chaque cluster, ses membres procéderont à l'élection d'un représentant ayant un capital minimal qui représentera le cluster dans les phases suivantes ;
8. évaluer la configuration actuelle :
   (a) Calculer les erreurs locales $\sigma(C_j)$ avec $j \in \{1, \ldots, K\}$;
   (b) Calculer l'erreur globale E_g
9. **Répéter**
10. Pour chaque équipe C_j, Choisir une des stratégies qui diminue son erreur locale : Liberer , Recruter , Virer , ou Réactiver ;

11. évaluer la configuration courante :
    (a) Calculer les capitaux de chaque objets ;
    (b) Calculer les erreurs locales $\sigma(C_j)$, avec $j \in \{1, \ldots, K\}$;
    (c) Calculer l'erreur globale E_g.
12. On se retrouve devant un autre état du jeu ;
13. Si le nombre d'itérations maximal $MaxIter$ n'est pas atteint, on revient à l'étape 10.

3.2.4 Expérimentation et résultats :

L'algorithme précédent a été implémenté et testé sur un exemple ou il s'agit de regrouper un ensemble de personnes caractérisées par les deux attributs : "âge" et "salaire". Les résultats obtenus sont représentés dans la figure 3.3. Une comparaison est ensuite effectuée par rapport au résultats obtenus avec les algorithmes classiques de clustering.

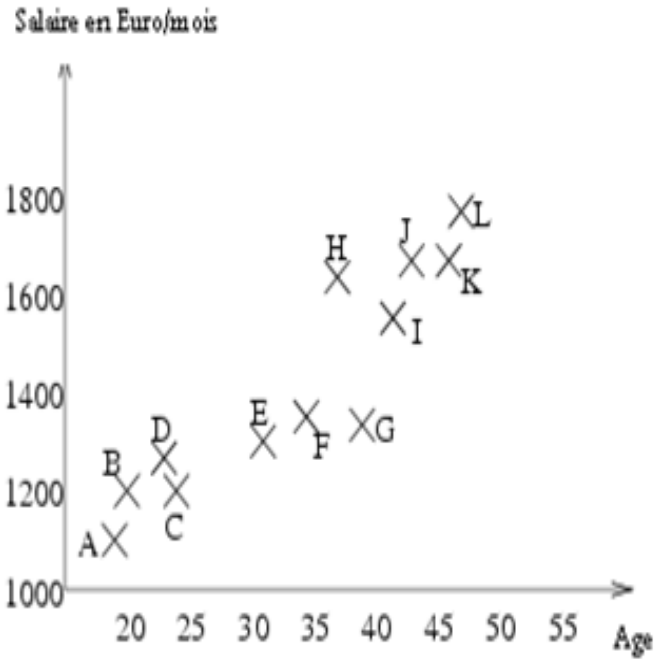

FIGURE 3.2 – Exemple de regroupement d'un ensemble d'individus

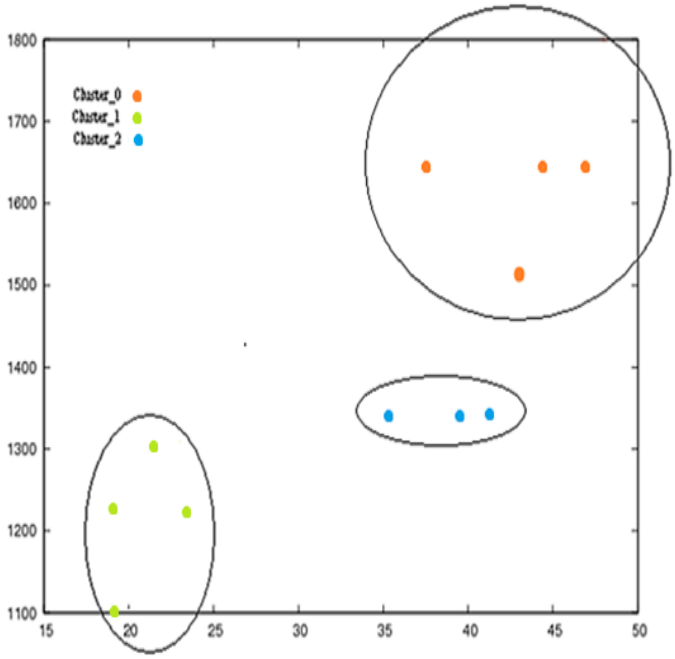

FIGURE 3.3 – Résultat obtenu par la présente approche

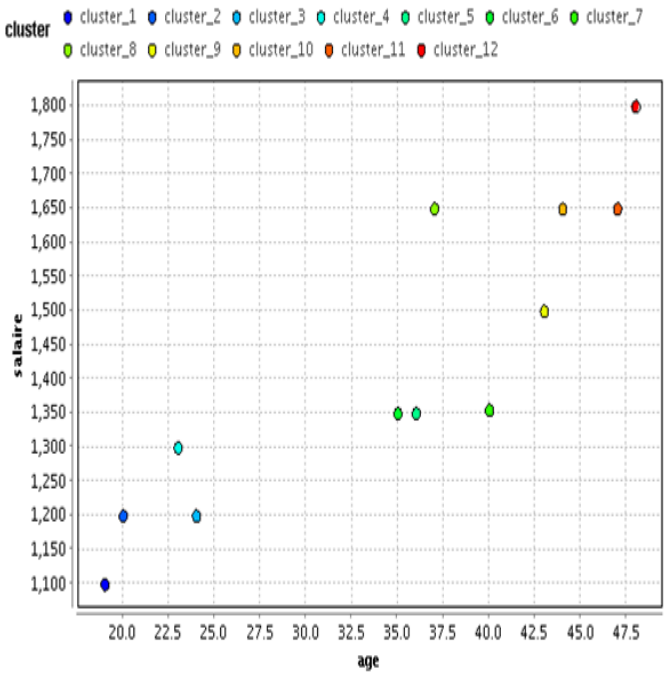

FIGURE 3.4 – Résultat obtenu par DBscan

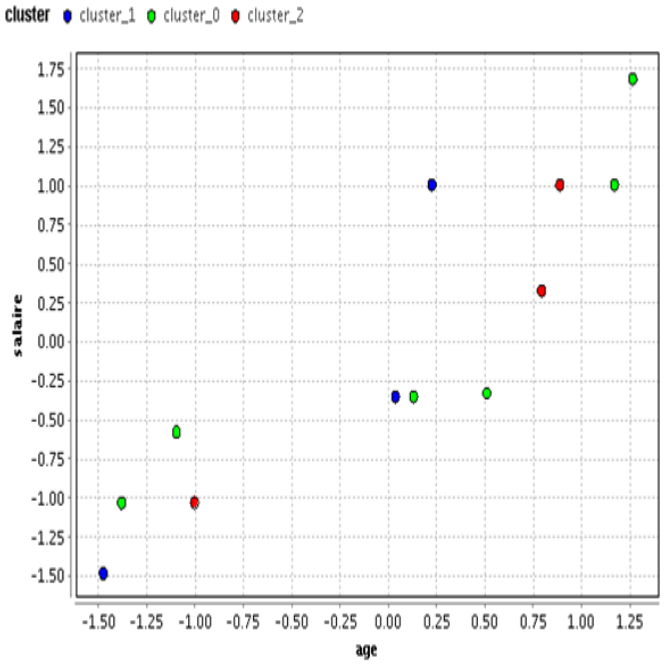

FIGURE 3.5 – Résultat obtenu par l'algorithme aléatoire

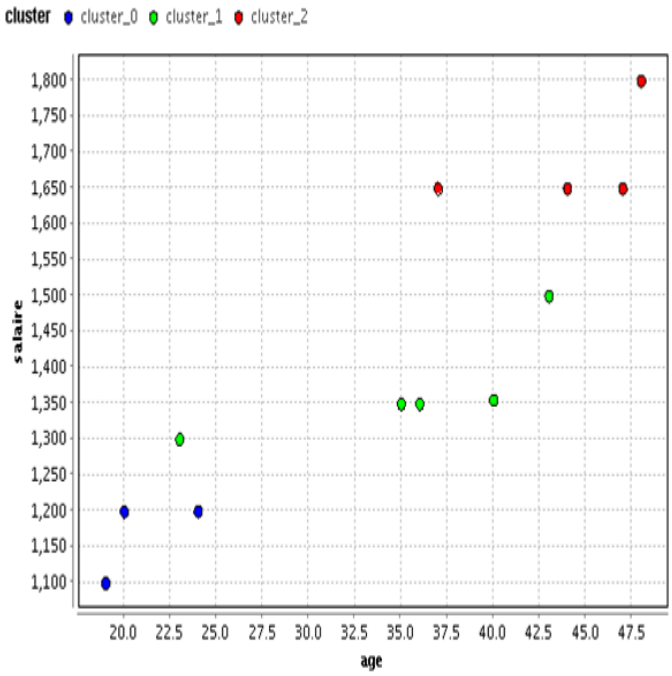

FIGURE 3.6 – Résultat Obtenu par l'algorithme K-means

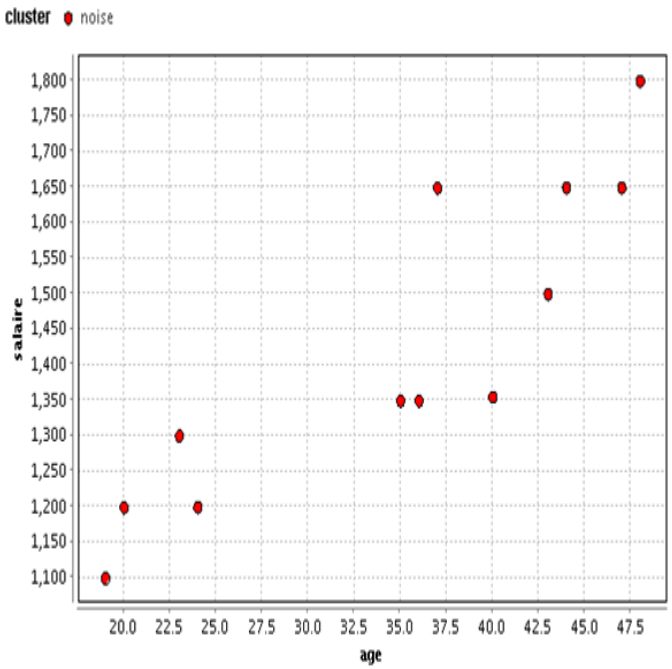

FIGURE 3.7 – Résultat Obtenu par l'algorithme SVM

On remarque clairement d'après les figures ci-dessus que l'algorithme basé sur la théorie des jeux coopératif mis en oeuvre par l'auteur [17] donne des résultats conformes aux attentes, par opposition aux résultats obtenus grâce aux algorithmes classiques qui sont décalés relativement aux résultats attendus.

D'autres expérimentations ont été réalisées par l'auteur sur des données synthétiques puis réelles, et les résultats obtenus par l'approche choisie sont très satisfaisants et conformes à la réalité, ce qui montre l'intérêt et la consistance de l'approche du problème de clustering par la théorie des jeux.

Conclusion :

Dans ce chapitre, nous avons synthétisé deux travaux abordant le problème du clustering par les jeux coopératifs. Notre choix s'est porté sur ces travaux vue qu'ils traitent théoriquement cette approche et les résultats qu'ils énoncent ne s'appliquent pas uniquement à un cas bien spécifique.

Ce survol des travaux de littérature nous a inspiré pour proposer dans le chapitre à suivre, une nouvelle approche, qui pourrait bien fournir des résultats intéressants pour le partitionnement des données.

4

jeux non-coopératifs et clustering

Introduction

Dans cette section, nous présentons une nouvelle approche, qui modélise le problème de clustering dans un contexte de jeu non-coopératif.

Notre approche consiste à trouver une partition de l'espace de départ telle que les données appartenant à un même groupe soient plus similaires entre elles qu'avec les données issues d'un autre groupe. Elle nous permet de construire K partitions (clusters) initiales d'individus similaires et les améliorer afin d'obtenir des clusters correspondant à un équilibre du jeu associé.

4.1 Idée globale de notre proposition

Le modèle que l'on mettra au point s'applique sur des objets qui sont représentés dans un espace vectoriel à deux dimensions. Chaque objet est décrit par un vecteur d'attributs numériques, il est donc associé à un point dans un espace vectoriel (géométrique) euclidien, de sorte que les distances entre les points observées reflètent les dissimilitudes/similitudes entre les objets respectifs.

En désignant chaque objet i par un vecteur $x_i = \{x_{i1}, x_{i2}\} \in \mathcal{R}^2$, et en notant l'ensembles des objets par $D = \{x_i, i = \overline{1,n}\}$, on partitionne l'ensemble D en k clusters de telle sorte qu'un point donné sera dans un et seulement un seul cluster, k étant connu à priori, est un paramètre d'entrée de l'algorithme.

Chaque cluster est représenté par un point unique de R^2 appelé moyenne ou centroïde du cluster. On note l'ensemble des centroïdes par $CR = \{CR_j, j = \overline{1,k}\}$.

Etant donné que les attributs définissant les objets qui seront étudiés sont de type numérique, la mesure de similarité utilisée sera la distance euclidienne, car elle présente l'avantage de la simplicité d'utilisation et d'implementation et la bonne adaptation à la mesure de la similitude des objets décrits par des attributs numériques.

4.2 Pourquoi la théorie non-coopérative ?

Les deux branches de la théorie des jeux "non-coopérative" et "coopérative" diffèrent dans leur façon de formaliser l'interdépendance entre les joueurs. Dans la théorie non-coopérative, un jeu est un modèle détaillé de tous les mouvements disponibles des joueurs. Par contre, la théorie coopérative fait abstraction des détails, et ne décrit que les résultats qui se produisent lorsque les joueurs se rassemblent dans différentes combinaisons.

La littérature traitant le problème du clustering par la théorie des jeux coopératifs est assez abondantes, à l'opposé de l'approche non-coopérative de ce problème qui reste mal explorée. C'est ainsi que notre contribution s'inscrit dans cette dernière approche afin d'évaluer la pertinence et l'efficacité d'une vision "individualiste" du problème du clustering qui fait abstraction de toute stratégie coalitionnelle et qui porte sur l'optimisation individuelle des gains.

4.2.1 Modélisation du problème sous forme d'un jeu non-coopératif

Soit une base de données D contenant n objets. Ces objets sont décrits par r critères (attributs).

L'evaluation de la $i^{éme}$ donnée selon le critère j est notée par $v(i,j)$, $i = \overline{1,n}$, $j = \overline{1,r}$.

Pour une classification des données en k classes (clusters), on procède comme suit :

Etape 1 :

• Regrouper les données séparées par des distances inférieures à un certain seuil, noté *eps* (prédéfini par l'utilisateur), dans m classes ($m < n$). Chaque classe $C_i, i \in \{1 \ldots, m\}$ est caractérisé par un poids $p(i)$ qui représente le cardinal de la classe, et un centroïde noté $CR_i = (CR(i,1), CR(i,2))$ où $CR(i,j) = \frac{1}{p(i)} \sum\limits_{l \in C_i} v(l,j), j = \overline{1,r}$.

Le cas $eps = 0$ correspond à des classes qui contiennent seulement les données identiques sur tous les critères.

Si, pour un certain seuil *eps*, une donnée peut être affectée à plusieurs classes, alors celle-ci sera affectée à la classe la plus proche.

• Ordonner les m classes (clusters) obtenues par ordre décroissant de leur poids.

• Sélectionner les k premières classes ayant les poids les plus élevés.

• les centroïdes des $m - k$ classes restantes seront considérées comme des joueurs.

On note $I = \{k+1, ..., m\}$, l'ensemble de ces joueurs.

• Chacune des k classes sélectionnées est un cluster qui représente une stratégie possible pour chaque joueur $i \in I$.

On note $X_i = \{1, ..., k\}$: l'ensemble des stratégies du joueur i, $\quad i \in I$.

• La fonction de gain $u_i, i \in I$ définie par la distance entre le centroïde de la classe joueur et le centroïde de la classe stratégie est donnée par :

$$u_i(c_j) = \frac{1}{d(ci, cj) + 1}, \qquad u_i(c_j) \in]0, 1]; \ \forall i \in I$$

Où $d(c_i, c_j)$ est la distance Euclidienne entre le centroïde de la classe joueur i et le centroïde du cluster j. On note que le gain $u_i \in]0, 1]$.

Si la distance entre le centre de la classe joueur et le centre du cluster est égale à 0, alors le joueur obtient un gain maximal qui est égale à 1. Plus la distance est grande, plus le gain est petit, ce qui incite les joueurs à se rapprochers du cluster le plus proches d'eux.

A ce niveau, on aura défini tous les éléments du jeu $< I, X_i, U_i >$.

4.2.2 Résolution du modèle :

le jeu de clustering étant bien défini, on passe à la recherche de l'équilibre de Nash correspondant.

Pour cela, nous proposons de dérouler ce jeu en deux étapes :

Etape 1 :

cas 1 : les joueurs interviennent par ordre décroissant de leur poids.

cas 2 : les joueurs interviennent par ordre croissant de leur poids.

Après chaque itération, une mise à jour du centroïdes CR_j et du poids P_j du cluster choisi par le joueur i est effectuée.

Si le joueur i décide de jouer la stratégie j, alors le nouveau centre du cluster C_j sera donné par :

$$CR_i = \frac{(P_i * CR_i + P_j * CR_j)}{(P_i + P_j)} \tag{4.1}$$

Et le nouveau poids P_j du cluster sera :

$$P_j = P_j + P_i \tag{4.2}$$

A ce niveau, nous avons obtenu un partitionnement initial des données en k clusters.

Etape 2 : Amélioration des clusters :

Dans cette étape, on cherche à déterminer l'équilibre de Nash du jeu précédemment défini .

Après l'obtention du clustering initial(représenté par l'issue du jeu à l'étape 1), tous les objets réévalueront leur gains pour déterminer s'ils ont intérêt à garder leur stratégie actuelle ou bien opter pour une autre stratégie qui leur garantira un gain supérieur.

A chaque itération, on passe d'une issue à une autre jusqu'a l'obtention d'un équilibre de Nash ou tous les joueurs ont intérêt à ne pas changer de stratégie unilatéralement.

Vu que ce probléme est NP-difficile [11], il est nécessaire d'utiliser un algorithme d'approximation en introduisant une variable Max_iter qui limite le nombre d'itérations.

Evaluation de la Qualité du clustering

:

Définition 4.1. L'inertie d'un cluster mesure la concentration des points du cluster autour du centroide. Plus cette inertie est faible, plus petite est la dispersion des points autour du centroide. L'inertie d'un cluster C_j est défini comme suit :

$$I_j = \sum_{i \in C_j} d^2(CR_i, CR_j), j = \overline{1,k} \tag{4.3}$$

On désigne par inertie intra-cluster, le terme :

$$I = \sum_{j=1}^{k} \sum_{i \in C_j} d^2(CR_i, CR_j) = \sum_{j=1}^{k} I_j \tag{4.4}$$

Algorithme 1

- **Entrée :** D : la base de données ; K :Le nombre de clusters ($k \ll n$) ;
  $MaxIter$: Le nombre maximum d'itérations ;

Eps : La distance tolérée pour la construction des clusters initiaux ;

- **Sortie :** C , l'ensemble des K clusters ;

1. Regrouper toutes les données séparées par des distances inférieures à eps.
2. Calculer les centroïdes $CR_i, i = \overline{1,m}$ des m classes obtenues (un centroïde CR_i représente le centroide du cluster C_i).
3. Affecter les éléments qui appartiennent à plusieurs clusters au cluster le plus proche
4. Recalculer les centroïdes des m clusters.
5. Calculer les poids des m clusters ($P_i = |C_i|), i = \overline{1,m}$.
6. Classer les clusters par ordre décroissant de leurs poids .
7. Sélectionner les k premiers clusters et les mettre dans un ensemble ES, et mettre les clusters restant dans un ensemble EJ.
8. Tant que —E$_j$| $\neq 0, répéter$

a) Sélectionner le cluster C_t ayant le poids minimum dans l'ensemble EJ.
b) Calculer la distance euclidienne entre le centroïde CR_t et les centroïdes des clusters qui appartiennent à ES. ie $d(CR_t, CR_p)$ pour tout $Cp \in ES$.
c) Ajouter les éléments du cluster C_t au cluster C_{p^*} qui vérifie :

$$dist(CR_t, CR_{p^*}) = \min_{C_p \in ES} d(CR_t, CR_p)$$

d) Mettre à jour les centroïdes CR_{p^*} en utilisant la relation (4.1).
e) Mettre à jour les poids P_{p^*} en utilisant la relation (4.2)
f) $EJ = EJ \setminus C_t$.

9. Initialiser nb_iter à 0 ;
10. Répéter
    - $Change = 0$;
    - Pour chaque cluster $C_i \qquad i = 1 : K$
    - ⋆ Pour chaque élément $j \in C_i$ calculer $d(Cr_j, Cr_l), \qquad l = \overline{1, K}$
    - ⋆ Si $d(CR_j, CR_i) \neq \min dist(j, CR_i)$ alors
    - ∗ ajouter j au cluster C_{l^*} qui vérifie $C_{l^*} = \min\limits_{l} d(CR_j, CR_l)$.
    - ∗ change = change+1 ;
    - $Nbr\_iter = nbr\_iter + 1$;.
11. Jusqu'à avoir $(nbr_{iter} > MaxIter)$ ou $(change = 0)$;
12. Calculer l'inertie intra-clusters.
13. Affichage des clusters $C_i, \qquad i = \overline{1, K}$.

Concernant l'algorithme 2, on reprend les mêmes étapes de l'algorithme 1, en remplaçant l'instruction $8 - a$ par l'instruction :
"Sélectionner le cluster C_t ayant le poids maximal dans l'ensemble EJ."

4.3 Exemple d'application sur une base de données réelle

On applique les deux versions notre algorithme sur un ensemble de données de la base de données classique "Iris", disponible à partir de dépôt de data mining UCI [20].

4.3.1 Présentation de la base de données

La base de données "Iris" contient 150 points de données appartenant à trois classes. Chaque classe représente une espèce différente de la fleur d'iris. Il existe 50 points de chaque classe.

Bien qu'il existe quatre dimensions (largeur des sépales, longueur des sépales, largeur des pétales, et longueur des pétales), seulement deux dimensions (largeur des pétales et la longueur des pétales) sont suffisantes pour distinguer les trois classes.

4.3.2 Résultats expérimentaux

Après application de notre algorithme implémenté sous matlab, pour une valeur de $m = 3$, $eps = 0$, et en utilisant seulement les deux attributs largeur et longueur des pétales, on a obtenu les résultats illustrés par les figures (4.1) pour le premier scénario (ordre croissant)et la figure (4.2) pour le second scénario (ordre décroissant) .

Le premier modèle nous donne une distribution des objets suivant les classes comme suit : 46 objets dans la première classe, 51 objets dans la deuxième classe, et 53 dans la troisième classe, tandis que Le deuxième modèle nous donne la distribution suivante : 48 objets dans la première classe, 50 objets dans la deuxième classe, et 52 objets dans la troisième classe.

4.3.2.1 Résultats de l'algorithme 1 ($1^{ère}$ variante)

Après application de cette variante de notre algorithme, On a obtenu 3 clusters de cardinalités :
$|C1| = 46$, $|C2| = 51$, $|C3| = 53$.

Les centroïdes des clusters ont pour coordonnées :
$C_{R1} = (6.8239, 3.0783)$, $C_{R2} = (5.0039, 3.4000)$, $C_{R3} = (5.80002.7000)$

Les vecteurs représentatifs des clusters sont :

```
C1=( 149   148   146   145   144   142   141   140   138   137   136
     133   132   131   130   129   126   125   123   121   119   118
     117   116   113   112   111   110   109   108   106   105   103
     101   87    78    77    76    75     66    59    57    55    53
     52    51).

C2=( 107    50    49    48    47    46    45    44    43    42    41
```

```
      40    39    38    37    36    35    34    33    32    31    30
      29    28    27    26    25    24    23    22    21    20    19
      18    17    16    15    14    13    12    11    10     9     8
      7     6     5     4     3     2     1)

C3=(  150   147   143   139   135   134   128   127   124   122   120
      115   114   104   102   100   99    98    97    96    95    94
      93    92    91    90    89    88    86    85    84    83    82
      81    80    79    74    73    72    71    70    69    68    67
      65    64    63    62    61    60    58    56    54)
```

l'inertie intra-cluster $I = 37.1412$

Représentation graphique des résultats :

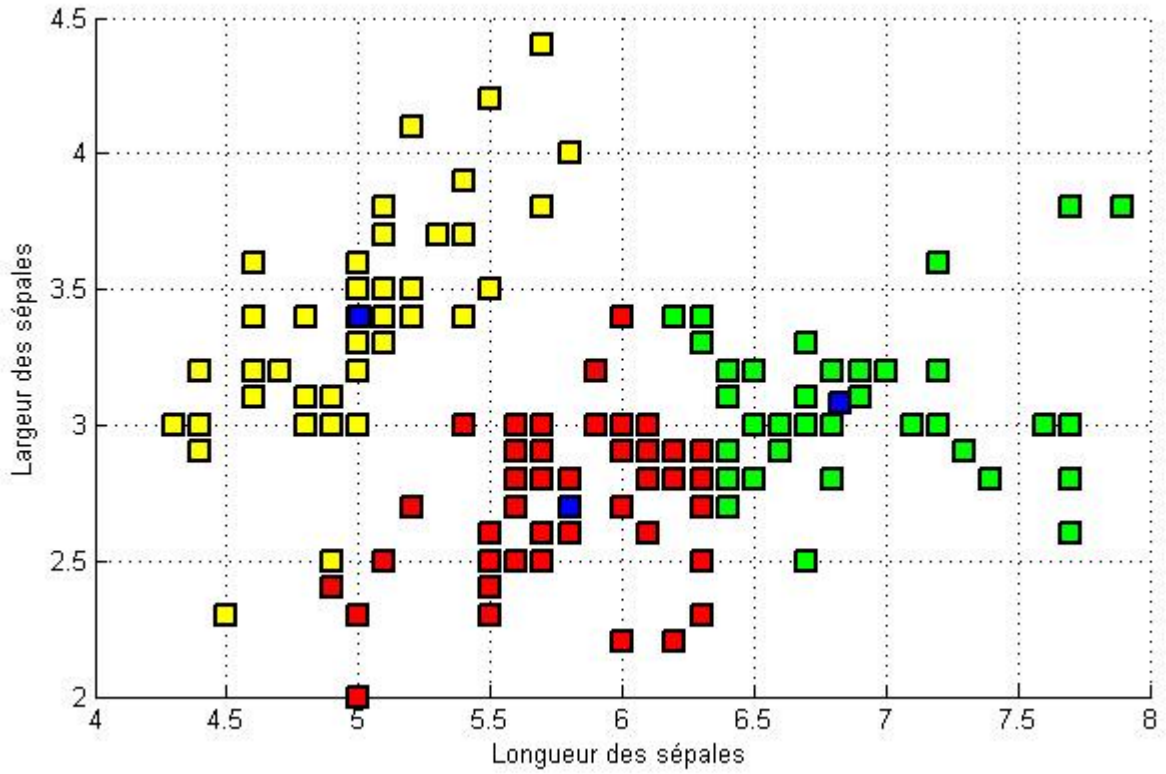

FIGURE 4.1 – Resutats de l'algorithme $1(1^{ère}variante)$.

4.3.2.2 Résultats de l'algorithme 2 ($2^{ème}$ variante)

On aura 3 clusters de cardinalités :
$|C1| = 47, \quad |C2| = 50, \quad |C3| = 53$

Les coordonnées des centroïdes des clusters sont :
$C_{r1} = (6.812766; 3.074468)$, $C_{r2} = (5.0060; 3.4180)$ $C_{r3} = (5.773585; 2.692453)$

Les vecteurs représentatifs des clusters :

```
C1=( 149   148   146   145   144   142   141   140   138   137   136
     133   132   131   130   129   126   125   123   121   119   118
     117   116   113   112   111   110   109   108   106   105   104
     103   101   87    78    77    76    75    66    59    57    55
     53    52    51)

C2=( 50    49    48    47    46    45    44    43    42    41    40
     39    38    37    36    35    34    33    32    31    30    29
     28    27    26    25    24    23    22    21    20    19    18
     17    16    15    14    13    12    11    10     9     8     7
     6     5     4     3     2     1 )

C3=( 150   147   143   139   135   134   128   127   124   122   120
     115   114   107   102   100   99    98    97    96    95    94
     93    92    91    90    89    88    86    85    84    83    82
     81    80    79    74    73    72    71    70    69    68    67
     65    64    63    62    61    60    58    56    54)
```

l'inertie intra-cluster $I = 37.1237$

Nous remarquons que l'inertie intra-cluster fournie par la $2^{ème}$ variante est meilleur que celle fournie par la $1^{ère}$ variante.

Représentation graphique des résultats :

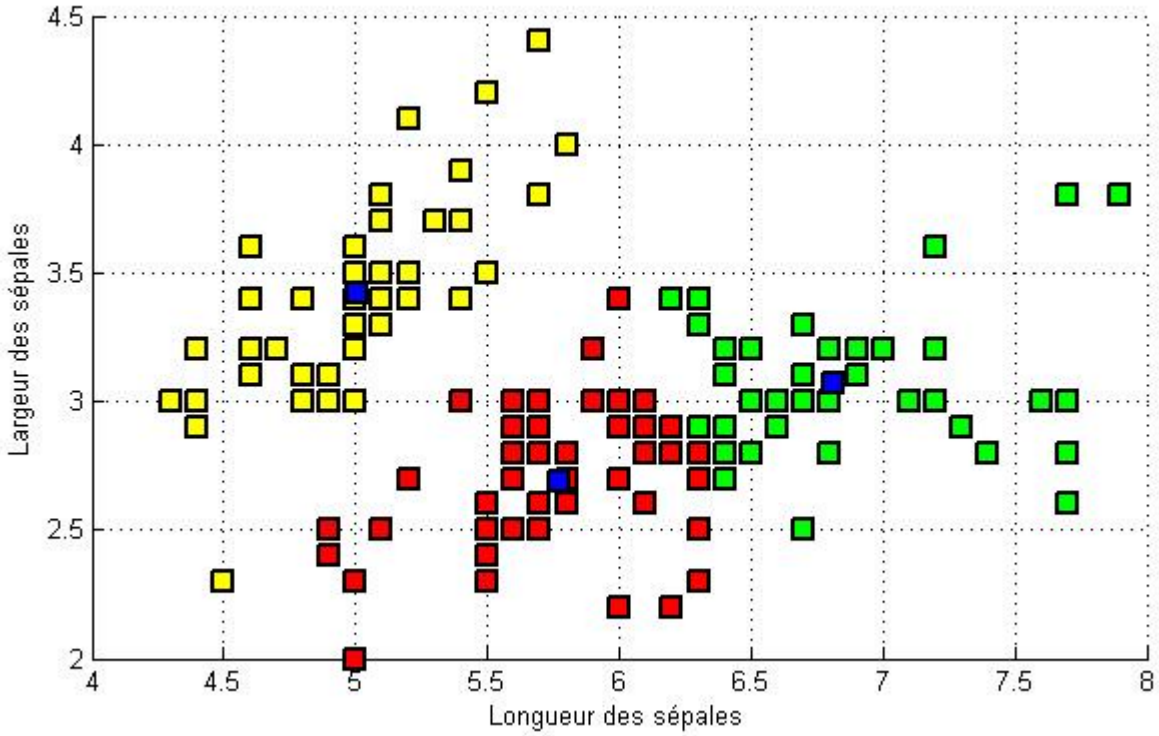

FIGURE 4.2 – Resutats de l'algorithme $2(2^{ème} variante)$.

4.3.3 Résultats de l'algorithme du K-means sous R :

Afin de comparer notre algorithme avec l'algorithme du K-means, on a eu recours au logiciel R qui intègre la méthode du K means et la base de données "Iris" de manière native

> iris.

Cette instruction donne la composition de cette base.

> cl< − kmeans (iris [,1 :2], 3)

Cette instruction nous permet d'exécuter l'algorithme du K-means sur le jeu de données iris en utilisant les deux attributs longueur et largeur des sépales, en utilisant 3

centres.

A l'issue de son exécution, K-means() associe chaque donnée à un groupe, les groupes sont numérotés de 1 à 3.

Les résultats obtenus sont illustrés par la figure suivante :

```
> cl
K-means clustering with 3 clusters of sizes 47, 50, 53

Cluster means:
  Sepal.Length Sepal.Width
1     6.812766    3.074468
2     5.006000    3.428000
3     5.773585    2.692453

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 3 1 3 1 3 1 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3
 [75] 1 1 1 1 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1
[112] 1 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 3 1 1 1 3 1 1 1 3 1 1 1 3 1
[149] 1 3

Within cluster sum of squares by cluster:
[1] 12.6217 13.1290 11.3000
 (between_SS / total_SS =  71.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"
```

FIGURE 4.3 – Résutats du K-means sous R.

Représentation graphique

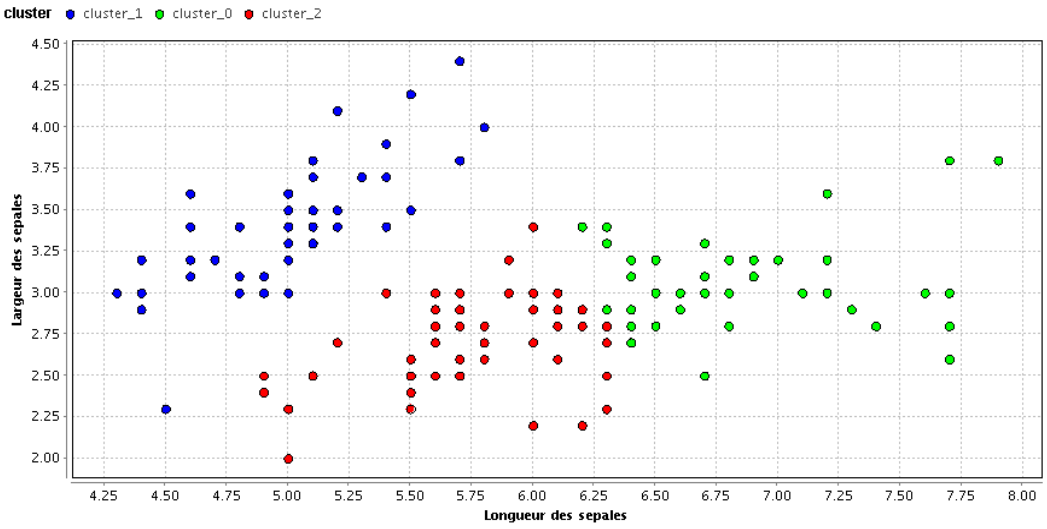

FIGURE 4.4 – Représentation des clusters obtenues par l'algorithme du K-means.

Comparaison des résultats

	Algo1	Algo2	K-means
Card C1	46	47	47
Card C2	51	50	50
Card C3	53	53	53
Centroide C1	(6.8239 ;3.0783)	(6.812766 ;3.074468)	(6.812766 ;3.074468)
Centroide C2	(5.0039 ;3.4)	(5.006 ; 3.428)	(5.006 ; 3.428)
Centroide C3	(5.8 ; 2.7)	(5.773585 ; 2.692453)	(5.773585 ; 2.692453)
Inertie	37.1412	37.1237	37.1237

Commentaires :

• Le partitionnement réel de la base de donnée Iris donne 3 classes composées de 50 élements chacune. Notre approche donne des résultats assez proches des résultats éspérés.

• On remarque que les résultats obtenus par l'algorithme du k-means sont identiques à ceux de notre algorithme (approche ascendante), tandis que l'approche descendante fournit des résultats légerement différents.

4.3.4 Résultats d'autres algorithmes sous RapidMiner :

RapidMiner est un environnement pour la machine d'apprentissage et les processus de data mining. Le concept de l'opérateur modulaire permet la conception de chaînes d'opérateurs emboîtés et complexes pour un très grand nombre de problèmes d'apprentissages. RapidMiner se présente comme une solution open-source pour le problème d'extraction de données, et largement utilisé par les chercheurs et les compagnies [1]

Dans nos expérimentations et pour but de faire une comparaison avec des méthodes existantes, nous avons utilisé le RapidMiner car il implémente certaines de ces méthodes (DBscane, K-medoides,...). Cela nous a aidé en premier lieu à gagner en temps et en performance car le RapidMiner offre beaucoup d'opérateurs qui permettent de modéliser les processus de data mining : classification et clustering, etc.

Voici un exemple de processus qui implémente le clustering de données sous RapidMiner :

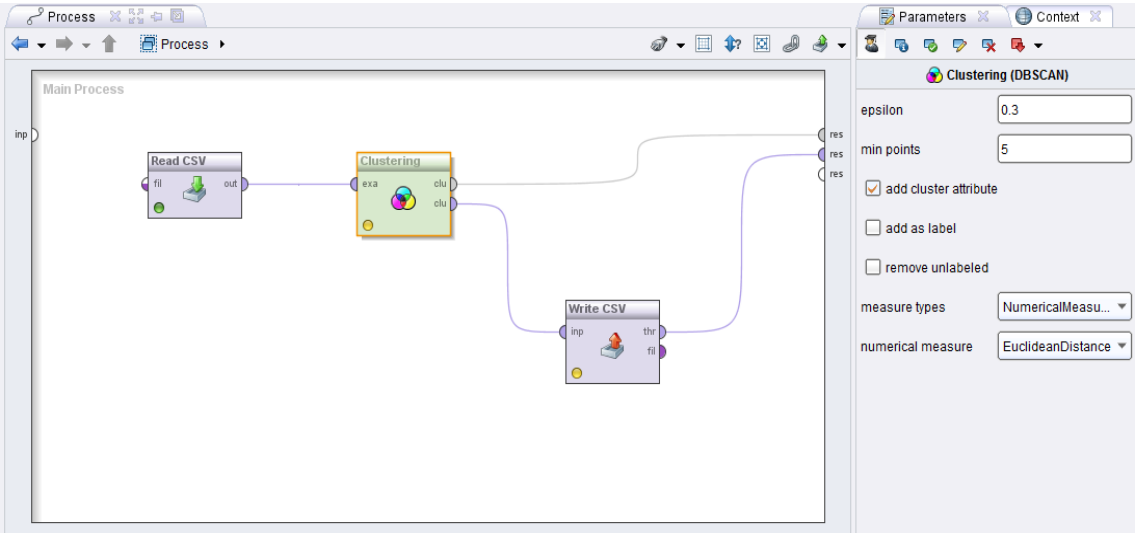

FIGURE 4.5 – Processus de clustering sous RapidMiner.

Le processus de clustering ci-dessus comporte les opérateurs suivants :

✓ ***Read CSV*** Cet opérateur sert a lire un fichier CSV qui contient la base de donnée.

✓ ***Clustering*** Cet opérateur éxecute un des algorithmes classiques de clustering

disponible sous Rapidminer(K-means, K-medoide, DBscan,algorithme EM,...).

✓ ***Write CSV*** Cet opérateur est utilisé pour écrire les données (Résultats de clustering) dans un fichier CSV.

4.3.4.1 DBscane

Cet algorithme intègre une notion de cluster basée sur la densité. Il permet de découvrir des clusters de formes arbitraires et ceci de proche en proche. Il requiert seulement deux paramètres d'entrée (Eps, MinPts).

Aprés l'éxecution de l'algorithme DBscane sous Rapidminer sur la base de données Iris avec les paramétres d'entrée (0.3, 5), on obtient les résultats suivants :

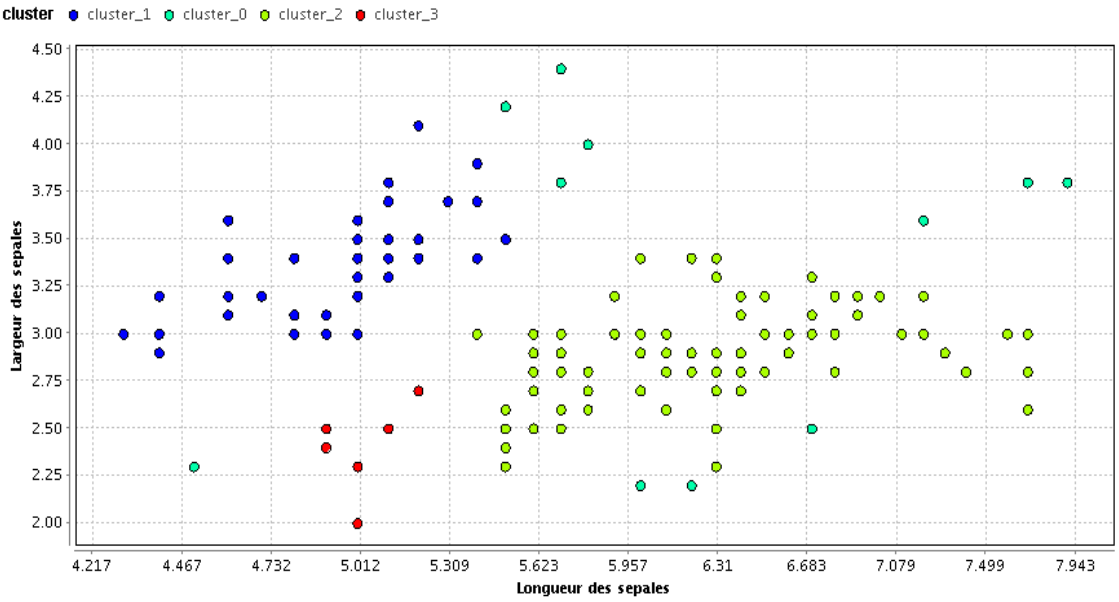

FIGURE 4.6 – Résutats du DBscane sous Rapidminer.

```
Cluster 0(bruits): 12 items
Cluster 1: 45 items
Cluster 2: 87 items
Cluster 3: 6 items
Total number of items: 150
```

4.3.4.2 Algorithme du K-medoide

Aprés l'éxecution de l'algorithme K-medoide, definie precédement, sous Rapidminer avec les paramétres d'entrée (K=3, Max runs = 10), on obtient les résultats suivants :

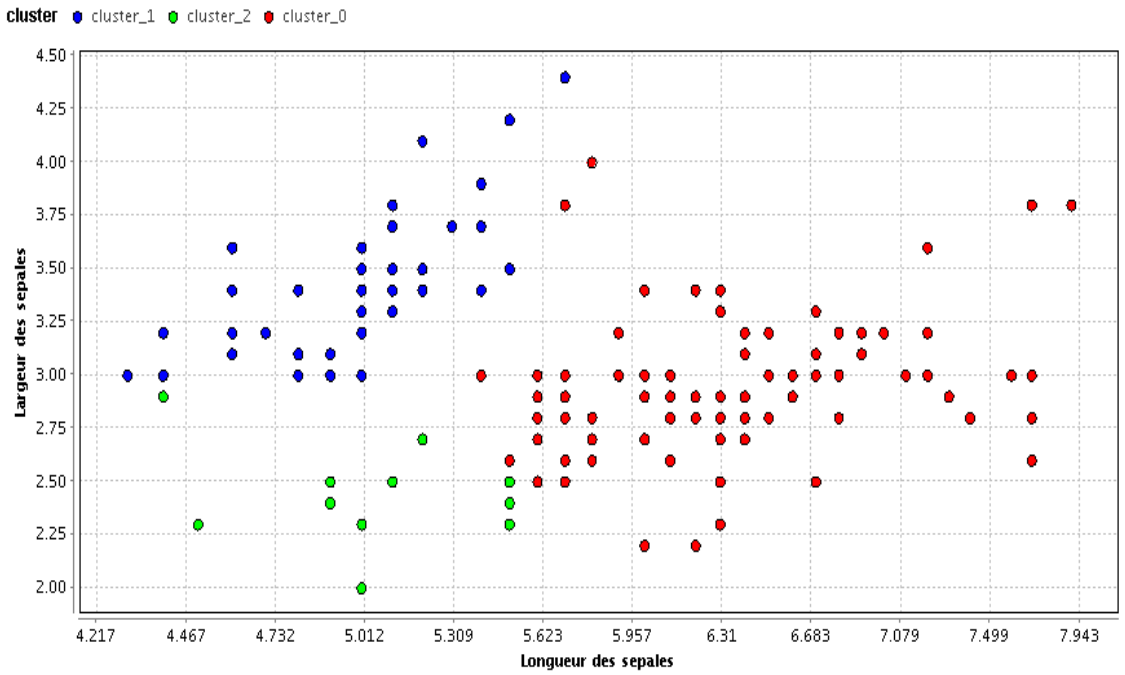

FIGURE 4.7 – Résutats de l'Algorithme du K-medoide sous Rapidminer.

```
Cluster 1: 92 items
Cluster 2: 46 items
Cluster 3: 12 items
Total number of items: 150
```

Comparaison des résultats

L'évaluation de la pertinence des groupes formés dans un algorithme de clustering reste une problématique ouverte. Cela vient du fait que le clustering est subjectif par nature, car il existe souvent différents regroupements possibles pour un même jeu de données.

L'application des differents algorithmes sur la base de données nous a permis de mettre en évidence les points suivants :

• Le partitionnement obtenu grace à l'algorithme DBscan ne correpond pas au partitionnement réel des fleurs de la base de donnée Iris. Ainsi cette algorithme détecte certains objets comme étant des bruits alors que tous les élements de la base de donnée devraient appartenir a une classe.

• Les clusters détéctés par l'algorithme K-médoide sont loin de refléter la réalité, on remarque ainsi que le cluster 1 contient 92 éléments alors que le cluster 3 n'en contient que 12.

	Algo1	Algo2	K-medoide	DBscan
Card C1	46	47	92	45
Card C2	51	50	46	87
Card C3	53	53	12	6
C_{r1}	(6.82 ;3.07)	(6.81 ;3.07)	(5.9 ;3.0)	-
C_{r2}	(5 ;3.4)	(5 ;3.42)	(5 3.3)	-
C_{r3}	(5.8 ;2.7)	(5.77 ; 2.69)	(4.9 ; 2.5)	-

Conclusion

Dans ce chapitre nous avons proposé une nouvelle approche du probléme de clustering. En effet, nous avons modélisé le probléme de clustering sous forme d'un jeu non cooperatif, que nous avons ensuite résolu en utilisant deux versions différentes de l'ordre d'intervention des joueurs. Les résultats expérimentaux obtenus en appliquant nos deux algorithmes implémentées sous MATLAB® sur la base de données Iris étaient conformes aux résultats espérés, contrairement à certains algorithmes classiques tel que DBscan et K-medoide.

Conclusion générale et perspectives

Dans ce mémoire, notre objectif a été d'aborder le problème du clustering du point de vue de la théorie des jeux. Pour cela, nous avons en premier lieu exposé un panorama du clustering, avec ses différentes propriétés et concepts avec un rappel des principales méthodes et algorithmes standards dédiés à sa résolution. Le deuxième chapitre de ce mémoire a servi à présenter les notions de base de la théorie des jeux aussi bien sa branche coopérative que non-coopérative avec leurs différents concepts de solution. Il a ensuiteété question de l'illustration du lien existant entre la problématique du clustering et la théorie des jeux à travers des exemples d'algorithmes mettant en oeuvre la théorie des jeux dans la résolution du problème de clustering. Enfin, dans le dernier chapitre, nous avons expliqué notre contribution dans ce domaine qui consiste en un algorithme de clustering basé sur la théorie des jeux non-coopératifs.

L'évaluation de la qualité d'un algorithme de clustering reste un problème ouvert. Il n'existe aucune approche reconnue comme étant universellement fiable, et aucune méthode ne peut être qualifiée de meilleure par rapport à une autre dans tous les contextes et les résultats obtenus sont à relativiser.

les résultats de l'expérimentation de notre algorithme sur une base de données réelle, dont les résultats sont connus à priori, ont permis de constater une bonne capacité prédictive de notre approche qui donne des résultats assez satisfaisant comparativement à certains algorithmes classiques. Mais les contraintes de temps ne nous ont pas permis de d'étayer cette constatation par d'autres comparaisons avec différentes formes de l'ensemble des données.

En guise de perspectives, nous envisageons d'axer les travaux futurs sur l'amélioration de notre algorithme :

- En incluant le cas de données manquantes qui se présente souvent dans la réalité.
- En étendant notre approche aux autres types d'attributs autres que numériques ;
- En étudiant la possibiilté d'adapter notre algorithme aux données à haute dimension
- Essayer de concevoir une technique pour calculer le nombre optimale de clusters k, pour rendre la classification automatique et sans aucune connaissance apriori.

Bibliographie

[1] *Le site rapidminer. disponible [en ligne] : http ://www.rapidminer.com.*

[2] F. BARACHE, *Sur la théorie des jeux évolutionnaires et ses applications en économie*, Mémoire de Magister en Mathématiques Appliquées, Université A. Mira de Béjaia, Algérie, 2007.

[3] D.L. DAVIES & D.W. BOULDIN, *A cluster separation measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 224-227, 1979.

[4] S.R. BULO, *A game-theoretic framework similarity-based data clustering*, Thèse de Doctorat, Université Ca Foscari de Venise Italie, 2009.

[5] J.C. DUNN, *Well separated clusters and optimal fuzzy-partitions*, Journal of Cybernetics, pages 95-104, 1974.

[6] G. FORESTIER, *Connaissances et clustering collaboratif d'objets complexes multi-sources*, Thèse de Doctorat,Université de strasbourg, France, 2010.

[7] B. DEVEZE & M. FOUQUIN, *Data mining c4.5 -dbscan*, Cours, Ecole d'ingénieurs en informatique EPITA, France, 2004.

[8] E.H. HAN & V. KUMAR G. KARYPIS, *Chameleon : A hierarchical clustering algorithm using dynamic modeling*, IEEE COMPUTER Society, pages 68-75, 1999.

[9] Y. CHUN & T. HOKARI, *On the coincidence of the shapley value and the nucleolus in queueing problems*, Seoul Journal of Economics, 2007.

[10] J. HAN & M. KAMBER, *Data mining : Concepts and techniques*, Management Systems (The Morgan Kaufmann Series in Data Management Systems),MORGAN KAUFFMAN, ISBN 1-55860-901-6, 2006.

[11] J. Laumônier, *Complexité de la théorie des jeux*, 2004.

[12] Y. BATISTAKIS & M.VAZIRGIANNIS M. HALKIDI, *Clustering validity checking methods*, SIGMOD Record, 2001.

[13] L. ROKACH & O. MAIMON, *Clustering methods*, Department of Industrial Engineering, Tel-Aviv University, 2002.

[14] E. HART & D. G. STORK P.DUDA, *Pattern classification*, Wiley, New York,, 2001.

[15] K.R. ANOOP & V.R EMBAR S. DHAMAL, S. BHAT, *Pattern clustering using co-operative game theory*, centenary conference, electrical engineering, Indian institute of science, BANGALORE, 2011.

[16] M. GÖTHE-LUNDGREN & PETER VÄRBRAND S. ENGEVALL, *The traveling salesman game : An application of cost allocation in a gas and oil company*, Department of Mathematics, Linköping Institute of Technology, S-581 83 Linköping, Sweden, 1998.

[17] S. SABRI, *Application de la théorie des jeux pour la définition, développement et implémentation d'un algorithme de clustering*, Mémoire de Magistère, Université de Béjaia, Algérie, 2011.

[18] D. SCHMEIDLER, *The nucleolus of a characteristic function game*, SIAM J. Appl. Math, pages 1163 - 1170, 1969.

[19] P. RAI & S. SINGH, *A survey of clustering techniques*, International Journal of Computer Applications (0975 - 8887) Volume 7- No.12, October 2010.

[20] Le site répertoire de machine d'apprentissage, *http ://archive.ics.uci.edu/ml/machine-learning-databases/iris/.*

[21] P.D STRAFFIN, *Game theory and strategy*, The Mathematical Association of America, pages 202-207, 1993.

[22] S.H TIJS, *An axiomization of the τ-value*, Mathematical Social Sciences, 13(2), pages 177-181, 1987.

Résumé : Ce mémoire s'intéresse à la problématique de la classification de données (clustering). Nous introduisons un nouvel algorithme de clustering basé sur la théorie des jeux non-coopératifs. Le problème du clustering est modélisé comme un jeu séquentiel non-coopératif sous forme stratégique suivant deux scénarios relatifs à l'ordre d'intervention des joueurs. L'algorithme proposé est ensuite implémenté, testé sur une base de données réelle et comparé aux algorithmes classiques préexistants dans ce domaine. La qualité des résultats obtenus montrent la pertinence et l'efficacité de l'approche choisie.

Mots clé : Clustering, jeux coopératifs et non coopératifs, équilibre de Nash.

Abstract : This thesis focuses on the problem of data classification (clustering). We introduce a new clustering algorithm based on the theory of non-cooperative games. The clustering problem is modeled as a sequential non-cooperative game in strategic form following two scenarios for the order response of the players. The proposed algorithm is then implemented, tested on a real database, and compared with existing conventional algorithms in this area. The quality of results obtained show the adequacy and efficiency of the approach.

Keywords : clustering, cooperative and noncooperative game, Nash equilibrium.