الجمهورية الجزائرية الديموقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة20 أوت1955 سكيكدة

UNIVERSITÉ 20 Août 1955 – Skikda

Faculté des sciences

Département d'Informatique

Mémoire de fin d'études en vue d'obtention d'u diplôme de

Master en informatique

Spécialité : Réseaux et Systèmes Distribués

THEME

Une approche de Clustering sans k : Cnok

Réalisé par :

- ATIETALLAH Nouara

Encadré par :

- Dr. ZEGHIDA Djamel

Année universitaire : 2022/2023

Remerciement

Avant tout, Je remercie dieu le tout puissant de m'avoir donné la volante et la patience d'entamer ce travail.

Je tiens à exprimer toute ma reconnaissance à mon encadreur de projet et mémoire, Dr ZEGHIDA Djamel. Je le remercie de m'avoir encadré, orienté, aidé et conseillé.

Mes remerciements vont également au corps professoral et administratif du département informatique de l'université 20 Août 1955 de Skikda pour leur collaboration, tous les efforts déployés et la qualité de l'enseignement dispensé.

Et enfin, j'aimerais adresser une mention particulière à ma famille et mes proches pour leurs soutiens constant et leurs encouragements.

Á la mémoire de mon défunt père…

Résumé

La méthode d'Identification des groupes d'individus au sein d'un jeu de données, et les regrouper en sous-ensembles similaires en fonction de diverses caractéristiques est appelé Clustering. Dans le monde de la science des données, nous pouvons utiliser des méthodes de Clustering pour obtenir des informations précieuses à partir de données en catégorisant les groupes auxquels appartiennent ces points lorsque nous appliquons un algorithme de Clustering. Les méthodes de Clustering souffrent encore de plusieurs insuffisances et nécessitent souvent l'initialisation de plusieurs paramètres comme le nombre de classes ou l'initialisation des groupes.

Dans ce mémoire, et afin d'améliorer les algorithmes de Clustering, nous proposons une approche de classification non supervisée, où l'algorithme découvre automatiquement les classes pour des données numériques sans connaître le nombre de classes a priori, sans partition initiale et sans paramétrage délicat.

L'algorithme proposé a été implémenté dans une application bureau Java qui est utilisée pour faire le Clustering d'un dataset introduit par l'utilisateur.

Mots clés : Clustering de données, classification non-supervisée, Clustering sans k.

ملخص

يُطلق على طريقة تحديد مجموعات الأفراد داخل مجموعة بيانات، وتجميعها في مجموعات فرعية متشابهة بناءً على خصائص مختلفة، التجميع. في عالم علم البيانات، يمكننا استخدام طرق التجميع للحصول على رؤى قيمة من البيانات عن طريق تصنيف المجموعات التي تنتمي إليها هذه النقاط عند تطبيق خوارزمية التجميع. لا تزال طرق التجميع تعاني من العديد من أوجه القصور وغالبًا ما تتطلب تهيئة العديد من المعلمات مثل عدد الفئات أو تهيئة المجموعات

في هذه الأطروحة ومن أجل تحسين خوارزميات التجميع نقترح نهج تصنيف غير خاضع للإشراف، حيت تكتشف الخوارزمية تلقائيا الفئات في البيانات الرقمية دون الحاجة لمعرفة عدد الفئات مسبقا، بدون تقسيم أولي وبدون معلمات حساسة.

تم تنفيذ الخوارزمية المقترحة على شكل تطبيق جافا لسطح المكتب والذي يتم استخدامه لتجميع مجموعة بيانات يقدمها المستخدم.

الكلمات المفتاحية: تجميع البيانات، التصنيف غير الخاضع للرقابة، التصنيف دون K.

Abstract

The method of identifying groups of individuals within a dataset, and grouping them into similar subsets based on various characteristics is called Clustering. In the world of data science, we can use Clustering methods to get valuable insights from data by categorizing the groups that these points belong to when we apply a Clustering algorithm. Clustering methods still suffer from several shortcomings and often require the initialization of several parameters such as the number of classes or the initialization of groups.

In this thesis, and in order to improve the Clustering algorithms, we propose an unsupervised classification approach, where the algorithm automatically discovers the classes in numerical data without knowing the number of classes a priori, without initial partition and without delicate parameterization..

The proposed algorithm has been implemented in a Java desktop application which is used to cluster a dataset introduced by the user.

Keywords: data Clustering, unsupervised classification, Clustering no k.

Table des matières

Chapitre II Conception

Chapitre III Implémentation

Chapitre VI Résultats expérimentaux et discussions

Liste des tables

Liste des figures

Introduction générale

Depuis l'apparition de l'informatique, nous sommes confrontés à une croissance effrénée de la quantité de données stockées dans le monde entier. Ces données se retrouvent sous formes diverses et variées et constituent un gigantesque vivier où l'homme vient puiser des informations et des connaissances pour en tirer le meilleur profit. Une analyse manuelle relève dès lors de l'impossible et au vu de ce constat, l'homme crée des techniques de recherche, d'analyses de données de plus en plus performantes.

Une idée prédomine désormais : regrouper des données et en soustraire des connaissances ; tout pense à croire que cette idée est en relation avec l'instinct primaire de l'homme qui par son comportement obéit à la logique aristotélicienne, celle qui "aime" tout catégoriser.

Dans la dernière centurie, et à partir de la deuxième moitié, l'homme commença à utiliser l'outil informatique pour automatiser des tâches de son quotidien y compris la classification de données. L'automatisation de cette tâche a donné naissance à la classification non-supervisée qui est un regroupement automatique de données issues elles-mêmes d'un ensemble de données, en groupes homogènes inconnus initialement, en fonction de leur similarité.

L'utilisateur introduit des données et reçoit comme résultats des données classifiées dans des groupes, ce processus est appelé Clustering. Cependant cette classification non-supervisée de données devient de plus en plus difficile, avec l'explosion de masse de données, les méthodes exactes prennent pour la résolution d'un problème de Clustering un temps qui prohibe leur utilisation.

Tandis que les méthodes exactes explorent tout l'espace de recherche afin de trouver la solution du problème, des méthodes dites heuristiques ont été proposées pour résoudre ce problème ces méthodes heuristiques explorent juste des parties de l'espace de solutions pour trouver une bonne solution (proche de la meilleure) ce qui réduit le temps écoulé pour résoudre le problème.

En vue d'apporter une contribution aux techniques de Clustering, nous nous sommes intéressés à un autre type de Clustering qui se caractérise par l'absence total de paramètre connus de Clustering à initialiser au préalable.

Outre l'introduction générale et la conclusion générale, ce mémoire sera organisé en quatre chapitres comme suite :

Le premier chapitre sera consacré à une présentation de l'état de l'art du Clustering de données, ses méthodes et les mesures de validité utilisées pour tester la qualité de résultats.

Le deuxième chapitre est dédié à la présentation de la conception de notre contribution afin d'implémenter l'approche CNo-K pour le Clustering. .

Le troisième chapitre sera consacré à l'implémentation de notre approche Clustering CNo-K dans une application bureau avec des captures détaillées de l'application.

Le quatrième chapitre donnera une comparaison entre les résultats de notre approche avec d'autres travaux similaires de Clustering, mettant en valeur la qualité de notre contribution.

Chapitre I

Clustering de données

1. Introduction

« Le seul moyen de faire une méthode instructive et naturelle, est de mettre ensemble les choses qui se ressemblent et de séparer celles qui différent les unes des autres. »

M. Georges Buffon, Histoire naturelle, 1749.

Depuis l'aube des temps, l'homme pratique la classification dans sa vie quotidienne, quand il essaie de répondre aux problèmes et questions sur la catégorie des objets, c'est-à-dire d'affectation d'objets à leur classe (en observant leurs formats, couleurs, tailles .. etc.), un exemple très simple , le fait qu'il avait distingué entre les plantes, en expérimentant empiriquement les propriétés thérapeutiques.

Au cours des siècles, ces classifications s'affinèrent pour nous livrer sous des formes précises et parfois très complexes les différents Herbiers que nous connaissons aujourd'hui (cette plante sert de nourriture, celle-ci pourrait aider efficacement à combattre la maladie et la douleur, ce qui a pris le nom de phytothérapie, . . . etc.).

Nous sommes contraints, au quotidien, à traiter, sans répit, des quantités d'information aussi abondantes que diverses. Ces traitements aident à la prise de décisions et la résolution de problèmes et varient de la simple représentation de données à leur analyse avancée. Pour cela, nous utilisons des méthodes et des techniques pour améliorer et rendre possible ces traitements, dont la classification. Cette dernière peut être supervisée, et préserve, tout court, sa nomination commune d'origine, ou non supervisée appelée dans ce cas, pour plus de précision le Clustering.

Nous nous intéressons dans ce chapitre au Clustering ; nous présenterons ses concepts de base et ses approches et les mesures de leur évaluation et validation.

2. Définition du clustring:

Le Clustering est une méthode d'apprentissage automatique qui consiste à regrouper des points de données par similarité ou distance. Il s'agit d'une méthode d'apprentissage non supervisée (c.-à-d. pas de classes prédéfinies), et a pour but de classifier des objets dans des clusters (groupes) en se basant sur leurs propriétés ou caractéristiques, les objets du même groupe sont les plus similaires, par contre les objets des groupes différents sont les plus dispersés.

Le Clustering est souvent utilisé dans le data mining, deep learning, l'analyse des images, et dans d'autres filières de sciences.

Le Clustering est utilisé, notamment, lorsqu'il est coûteux d'étiqueter les données. C'est néanmoins un problème mal défini mathématiquement : différentes métriques et/ou différentes représentations des données aboutiront à différents regroupements sans qu'aucun ne soit nécessairement meilleur qu'un autre. Ainsi la méthode de Clustering doit être choisie avec soin en fonction du résultat attendu et de l'utilisation prévue des données.

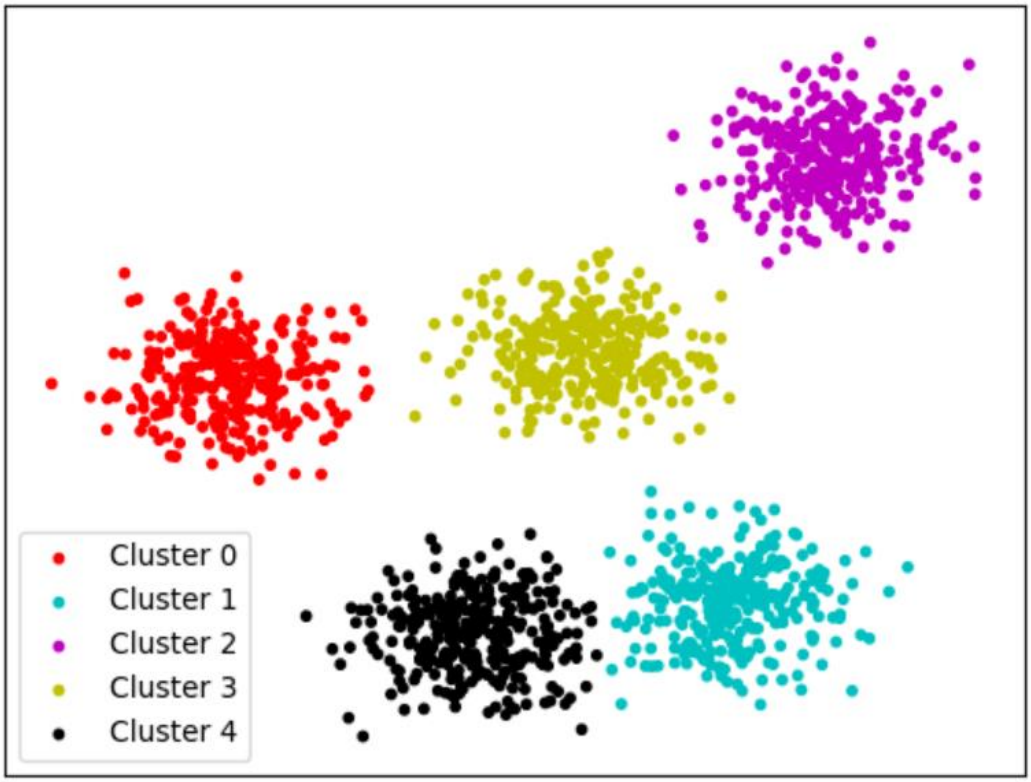

Figure 1 – Clustering.

2.1. Objectifs du Clustering

Mirkin [2] a identifié une liste d'objectifs de Clustering :

- Structuration : représentation des données comme un ensemble de groupes d'objets similaires (généralement, la structuration est l'objectif principal du Clustering).
- Description : des clusters en fonctions des caractéristiques.
- Association : la découverte des interrelations entre différents aspects du phénomène.
- Généralisation : un relevé de données sur les propriétés du phénomène que les données ont des relations avec elles, cet objectif nécessite une analyse multi-étage des objectifs précédents.
- Visualisation : représentation des structures des clusters comme une image visuelle.

2.2. Les étapes principales du Clustering

Le processus de Clustering se divise en trois étapes majeures [30, 31] :

1. La préparation des données : Les objets sont décrits par des variables qui sont de différentes natures (quantitatives, qualitatives, variables structurées...)
2. L'algorithme de Clustering : un algorithme de Clustering est choisi selon la nature des variables en entrée.
3. L'exploitation des résultats de l'algorithme : Cette étape permet de distinguer et d'analyser les classes pertinentes obtenue, afin d'aider à orienter le traitement suivant.

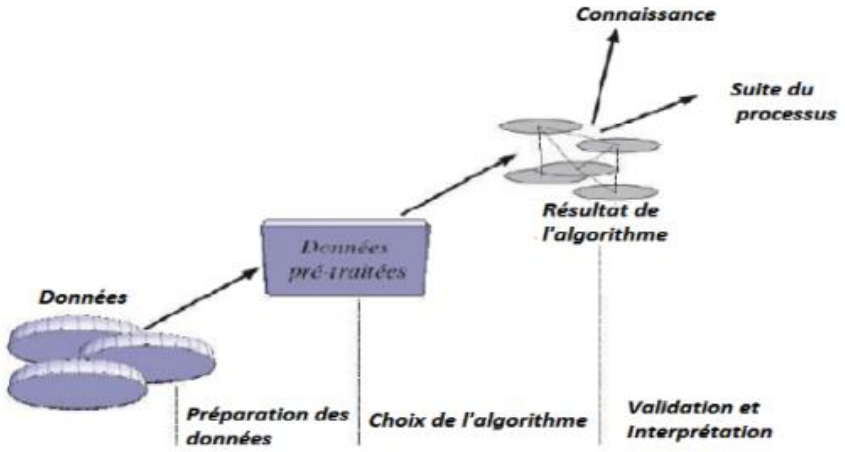

Figure 2– Les étapes principales du Clustering.

3. Représentation des données

Un ensemble d'objets peut être décrit par deux formats standards : la matrice des données, et la matrice de proximité.

3.1. Matrice de données

Dans un ensemble d'objets, chaque objet est décrit par un vecteur d'attributs, l'ensemble lui-même peut être représenté par une matrice de taille $n * d$ (où n représente le nombre d'objets, et d le nombre d'attributs), chaque ligne de cette matrice représente un objet et chaque colonne représente un attribut ou une caractéristique. Par exemple, pour des étudiants de la promo master 2 RSD, l'ensemble de caractéristiques sont : la taille, le poids, l'âge, et la moyenne. Chaque ligne de la matrice de données, (Table 1) [9], représente un individu (objet).

Forme classique

	Attributs			
	Taille	Poids	Age	Moyenne
AMMAR	178	70	23	18.5
DJIHED	187	87	25	15.5
KAIS	179	64	23	16
SALAH	183	83	25	16
ABDOU	178	85	24	17

Forme inversée

	Etudiants				
	AMMAR	DJIHED	KAIS	SALAH	ABDOU
Taille	178	187	179	183	178
Poids	70	87	64	83	85
Age	23	25	23	25	24
Moyenne	18.5	15.5	16	16	17

Forme utilisée dans la méthode de centroïde

	Attributs			
	Taille	Poids	Age	Moyenne
AMMAR	178	70	23	18.5
DJIHED	187	87	25	15.5
KAIS	179	64	23	16
{SALAH, ABDOU}	180.5	84	24.5	16.5

Table 1 - Matrices de données brutes.

D'habitude, les caractéristiques sont vues comme des axes orthogonaux, et les objets comme des points dans cet espace de d-dimensions. La représentation de cet espace est très limitée, car le nombre des axes orthogonaux qui peut être visualisé est trois axes au maximum (Figure 2), le bénéfice du Clustering est sa réussite d'organiser les données multidimensionnelles tandis que la perception visuelle échoue [4].

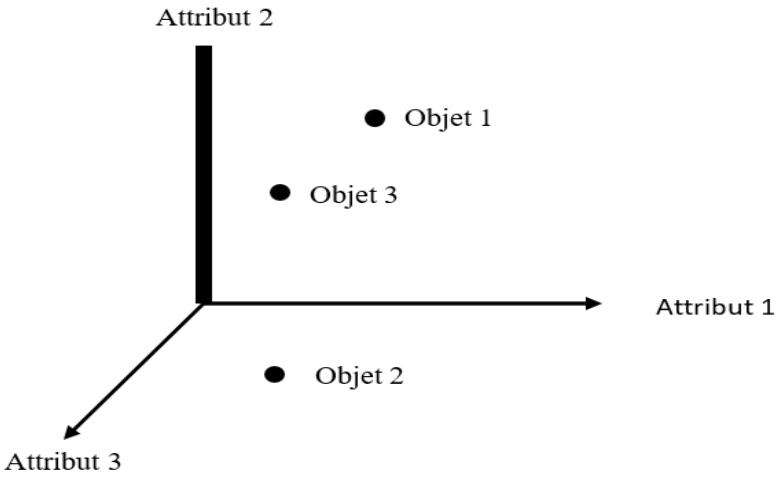

Figure 3- Objets en fonction d'attributs.

3.2. Matrice de proximité

Les techniques de Clustering ont généralement besoin d'un indice de proximité (de similarité ou de dissimilarité) établi entre chaque paire d'objets. Cet indice peut être calculé de la matrice de données ou de l'ensemble d'objets (sans nécessité la création de la matrice de données), la valeur de proximité de chaque paire d'objets est stockée dans une matrice Pi, appelée la matrice de proximité. Dans cette matrice la diagonale est ignorée car la similarité d'un objet par rapport à lui-même est évidente la valeur maximale, ou minimale en cas de distance (dissimilarité). Généralement, d (oi, oj) = 0 si et seulement si $i = j$.

On suppose que toutes les matrices de proximité soient symétriques, c'est-à-dire une paire d'objets a la même valeur de proximité indépendamment de l'ordre dans la fonction P (oi, oj) = P (oj, oi), alors une des moitiés de la matrice (le dessus de la diagonale ou le dessous) est ignorée aussi, voir (Table 2) [4, 5,9].

Matrice de dissimilarité (distance Euclidienne).

	Etudiants				
Etudiants	AMMAR	DJIHED	KAIS	SALAH	ABDOU
AMMAR	0				
DJIHED	19.57	0			
KAIS	6.58	24.44	0		
SALAH	14.29	5.68	19.52	0	
ABDOU	15.11	9.39	21.07	5.57	0

Matrice de dissimilarité (Euclidienne carrée).

Etudiants

Etudiants	AMMAR	DJIHED	KAIS	SALAH	ABDOU
AMMAR	0				
DJIHED	383	0			
KAIS	43.25	597.25	0		
SALAH	204.25	32.25	381	0	
ABDOU	228.25	88.25	444	31	0

Table 2 - Matrices de Proximité.

Hubert [6] et Gower [7], considèrent que la matrice de proximité est une matrice non symétrique [4]. Un exemple simple pour clarifier leur opinion pourrait être le temps nécessaire pour se déplacer d'un endroit vers un autre ; s'ils sont à des différentes altitudes, ce temps sera moindre pour la descente que pour la montée, un autre exemple des personnes qui vivent dans la ville i et travaillent dans la ville j, et ainsi de suite [8].

3.3. Types de données

Les variables d'un objet peuvent être continues, discrètes, ou binaires [10] :

- Les variables continues : ont une plage infinie et innombrable de valeurs (les variables réelles), par exemple la surface d'un trapèze ou d'un triangle.
- Les variables discrètes : ont généralement une plage finie de valeurs, ou dénombrable infinie (les variables entières) par exemple le nombre de mots dans un article.
- Les variables binaires : sont des variables discrètes qui peuvent prendre seulement deux valeurs, par exemple l'existence d'une chose, elle existe ou non.

Clifford et Stephenson [11] ont développé les trois catégories précédentes en six autres catégories plus détaillées.

3.4. Echelles de données

Deux grandes échelles existent : les échelles qualitatives (nominale et ordinale) et les échelles quantitatives (intervalle et ratio) [12], [4] :

- **L'échelle nominale** est une échelle faible (dénuée de sens), par exemple dans les réponses (oui ou non) ou l'égalité de deux objets ($x = y$, $x \neq y$), elles peuvent être codées par (0 et 1), les différentes couleurs des roses sont aussi un autre exemple de cette échelle (rouge=0, bleu=1, blanc=2, et ainsi de suite).

- **L'échelle ordinale** cette échelle est aussi faible, les nombres n'ont pas de sens sauf s'ils sont en relation avec un autre, par exemple les niveaux de difficulté (facile=1, moyen=2, et difficile=3), la différence entre cette échelle et l'échelle nominale est que les valeurs de cette échelle reflètent un ordre.

- **L'échelle d'intervalle** cette échelle donne un sens à la différence entre les objets. Une unité de mesure existe et l'interprétation des nombres dépend de cette unité, par exemple la température mesurée en Celsius (40° est plus grande que 25° par 15°).

- **L'échelle de rapport** (ratio) est l'échelle la plus forte dans laquelle les nombres ont un sens absolu, la différence entre cette échelle et la précédente, est que cette échelle possède un zéro absolu intrinsèquement immanent de la réalité, tandis que le zéro de l'autre échelle est défini arbitrairement. Le poids est un exemple de cette échelle, le 0 kg est la limite minimale.

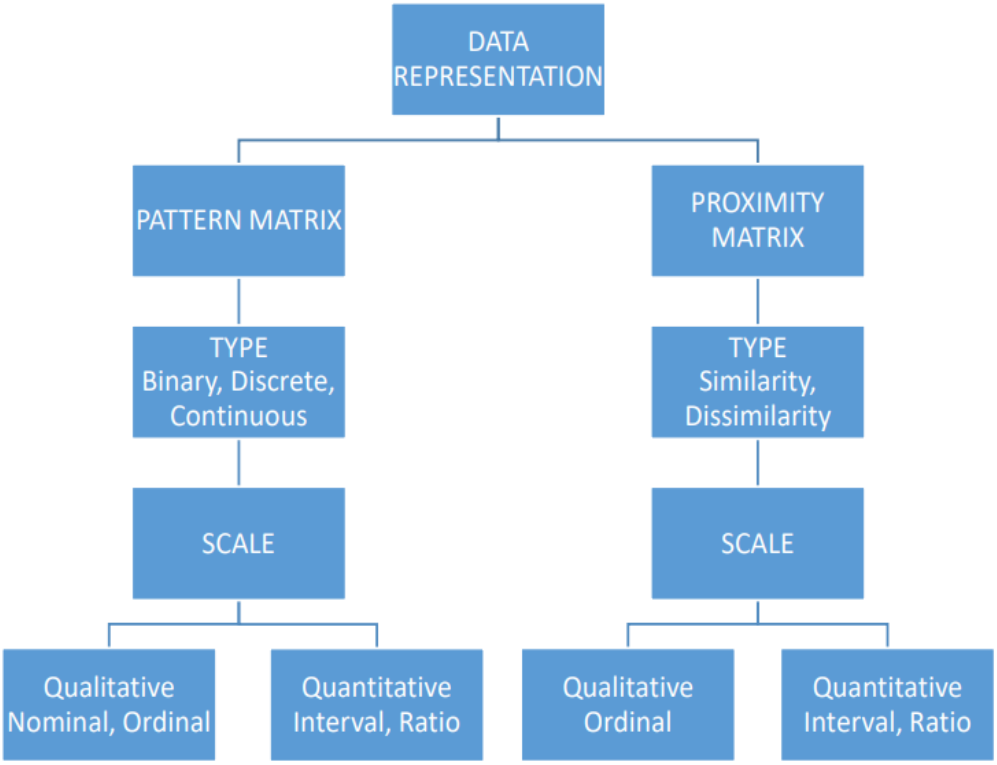

Figure 4 Format, types, et échelle de données.

3.5. Mesures de proximité

La construction de la matrice de proximité consiste au calcul de l'indice de proximité (de similarité ou de dissimilarité) pour chaque paire d'objets qui peuvent contenir des attributs de même type ou de types différents.

3.5.1. Mesures pour les données numériques

Distance de Minkowski : pour deux objets x et y la distance entre eux est définie comme suit :

$$d_{Mink}(x,y) = \left(\sum_{j=1}^{d} |x_j - y_j|^r\right)^{\frac{1}{r}} \qquad \#(01)$$

Où d est le nombre d'attributs

- Si r = 1, on obtient la distance de Manhattan : $d_{Man}(x,y) = \sum_{j=1}^{d} |x_j - y_j|$
- Si r = 2, on obtient la distance Euclidienne : $d_{Euc}(x,y) = (\sum_{j=1}^{d} |x_j - y_j|^2)^{\frac{1}{2}}$
- Si r = ∞, on obtient la distance maximale : $d_{Max}(x,y) = \max_{1\leq j\leq d} |xj - yj|$

3.5.2. Mesures pour les données catégoriques

L'appariement simple (Simple Matching) : pour définir cette fonction de distance il faut d'abord définir la fonction δ :

$$\delta(x,y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases} \qquad \#(02)$$

La distance d'appariement simple est définie comme la somme des fonctions δ appliquées aux attributs des deux objets x et y :

$$d_{sim}(x,y) = \sum_{j=1}^{d} \delta(x_j, y_j) \qquad \#(03)$$

La similarité de cet indice peut être calculée par inversion de la fonction δ, 1 pour l'égalité, et 0 pour l'inégalité.

3.5.3. Mesures pour les données binaires

La plupart des fonctions de Proximité entre deux vecteurs binaires utilisent les fonctions suivantes A, B, C, D

$$A = x.y = \sum_{j=1}^{d} x_j y_j \qquad \#(04)$$

$$B = \bar{x}.y = \sum_{j=1}^{d} (1 - x_j)\, y_j \qquad \#(05)$$

$$C = x.\bar{y} = \sum_{j=1}^{d} (1 - y_j) \qquad \#(06)$$

$$j=1$$

$$D = \bar{x}.\bar{y} = \sum_{j=1}^{d}(1-x_j)(1-y_j) \qquad \#(07)$$

- Jaccard : fonction de similarité $s(x,y) = \frac{A}{A+B+C}$

  de dissimilarité $d(x,y) = \frac{B+C}{A+B+C}$

- Dice : fonction de similarité $s(x,y) = \frac{A}{2A+B+C}$

  de dissimilarité $d(x,y) = \frac{B+C}{2A+B+C}$

- Yule : fonction de similarité $s(x,y) = \frac{AD-BC}{AD+BC}$

  de dissimilarité $d(x,y) = \frac{BC}{AD+BC}$

3.5.4. Mesures pour les données mixtes

Généralement un objet contient des attributs de plus d'un type, alors que les mesures de proximité précédentes nécessitent que tous les attributs d'un objet soient de même type, dans ce cas elles ne sont pas valides, Gower [13] a proposé la méthode de coefficient de similarité générale appliquée aux objets avec attributs de différents types [13, 14] :

$$s_{Gower}(x,y) = \frac{1}{\sum_{j=1}^{d} w(x_j,y_j)} \sum_{j=1}^{d} w(x_j,y_j)s(x_j,y_j) \qquad \#(08)$$

Tel que : S (x_j, y_j) : est un composant de similarité de l'attribut j défini :

- Pour les attributs quantitatifs comme :

$$S(x_j,y_j) = 1 - \frac{|x_j - y_j|}{R_j} \qquad \#(09)$$

W (x_j, y_j) : prend la valeur 0 si la comparaison des deux attributs n'est pas valide, par exemple la valeur d'un des deux attributs est manquante, sinon elle prend la valeur 1, et R_j est la rangée (ou la portée) de l'attribut j.

- Pour les attributs binaires comme :

$$S(x_j,y_j) = \begin{cases} 1, & si\ x_j et\ y_j\ sont\ présents \\ 0, & sinon \end{cases} \qquad \#(10)$$

W (x_j, y_j) : prend la valeur 0 si la valeur des deux attributs est manquante, sinon elle prend la valeur 1.

- Pour les attributs nominaux ou catégoriaux :

$$S(x_j, y_j) = \begin{cases} 1, & x_j = y_j \\ 0, & sinon \end{cases} \qquad \#(11)$$

W (x_j, y_j) : prend la valeur 0 si la comparaison des deux attributs n'est pas valide, par exemple la valeur d'un des deux attributs est manquante, sinon elle prend la valeur 1.

La fonction de distance de cette méthode est définie comme suit [13, 15] :

$$d_{Gower}(x, y) = \left(\frac{1}{\sum_{j=1}^{d} w(x_j, y_j)} \sum_{j=1}^{d} w(x_j, y_j) d^2(x_j, y_j) \right)^{\frac{1}{2}} \qquad \#(12)$$

- Pour les attributs quantitatifs, $d2(xj, yj)$ est définie comme :

$$d^2(x_j, y_j) = |x_j - y_j|^2 \qquad \#(13)$$

- Pour les attributs ordinaux ou continus, $d2(xj,yj)$ est définie comme :

$$d^2(x_j, y_j) = \left(\frac{|x_j - y_j|}{R_j} \right)^2 \qquad \#(14)$$

- Pour les attributs binaires, $d^2(x_j, y_j) = 0$

  Si les deux attributs sont présents ou absents, sinon $d^2(x_j, y_j) = 1$.

- Pour les attributs nominaux ou catégoriaux :

$$S(x_j, y_j) = \begin{cases} 0, & x_j = y_j \\ 1, & sinon \end{cases} \qquad \#(15)$$

3.5.5. Statistique Chi-Square

Les mesures présentées précédemment calculent la proximité entre les objets, ici nous allons introduire une des mesures d'association entre variables (attributs). Cette mesure [10] est habituellement utilisée pour tester l'hypothèse de l'indépendance entre les variables.

La statistique de Chi-square (ou Chi-carré) [10, 14] représente la distribution conjointe de deux variables catégoriques. Des mesures d'associations peuvent être définies en se basant sur cet indice.

On suppose l'ensemble D de données avec n objets de d attributs catégoriques, et u et v deux attributs de l'ensemble d avec respectivement les domaines $\{u_{1,2},\ldots, u_p\}$, et $\{v_1, v_2, \ldots, v_q\}$. La table de contingence représente la distribution conjointe de variables u et v.

	v_1	v_2	...	v_q	Totales
u_1	n_{11}	n_{12}	...	n_{1q}	r_1
u_2	n_{21}	n_{22}	...	n_{2q}	r_2
⋮	⋮	⋮	...	⋮	⋮
u_p	n_{p1}	n_{p2}	...	n_{pq}	r_p
Totales	c_1	c_2	...	c_q	n

Table 3 - Table de contingence des variables u et v.

n_{ij} est le nombre d'objets de l'ensemble D qui ont la valeur ui de l'attribut u, et la valeur v_j de l'attribut v. r_i est le nombre d'objets possédant la valeur u_i de l'attribut u, c_j est le nombre d'objets possédant la valeur v_j de l'attribut v. Selon la (Table 3),

La statistique chi-square est définie comme suit [10, 14] :

$$\chi^2 = \sum_{i=1}^{p}\sum_{j=1}^{q}\frac{(n_{ij} - e_{ij})^2}{e_{ij}} \qquad \#(16)$$

Tel que : $e_{ij} = \frac{r_i c_j}{n}$.

Il existe d'autres extensions de cette mesures, voir [10]. D'autres fonctions de proximité ont été présentées en [14].

4. Problèmes typiques et caractéristiques désirables

Un bon algorithme de Clustering devrait avoir les propriétés suivantes [16, 17] :

- **Scalabilité :** ou passage à l'échelle est la capacité de l'algorithme à bien fonctionner avec les grands nombres d'objets, et avec des objets à hautes dimensions.
- **Capacité à gérer différents types d'attributs :** numériques, binaies, ordinaux, nominaux, etc.
- **Découverte des clusters de formes arbitraires** : plusieurs méthodes déterminent les clusters en se basant sur la distance Euclidienne ou la distance de Manhattan, l'utilisation de pareilles distances donne des clusters sphériques (convexes), tandis que les clusters doivent être de formes arbitraires.
- **Besoin minimum de connaissances pour déterminer les paramètres :** il est souhaitable qu'une méthode ne nécessite pas des conseils limités de la part de l'utilisateur (par exemple le nombre de clusters), notamment dans les grands ensembles de données et de dimensions, afin d'éviter tout biais de résultat.

- **Capacité à gérer les bruits et les exceptions :** dans les ensembles des données du monde réel, il existe des bruits, des exceptions, des valeurs manquantes ou inconnues, et des données erronées.
- **Indifférent à l'ordre des données en entrée :** l'algorithme doit produire le même résultat du mêmes données présentées en différents ordres.
- **Capacité de fonctionner d'une manière incrémentale :** les mises à jour incrémentales peuvent arriver à tout moment, et il faut les intégrer dans le processus du Clustering existant sans le redémarrer à partir du zéro.
- **Incorporation de contraintes par l'utilisateur :** les applications du monde réel peuvent avoir besoin des contraintes indiquées par l'utilisateur.
- **Interprétabilité et utilisabilité :** les résultats de Clustering doivent être interprétables, compréhensibles, et utilisables. Le Clustering a besoin d'être lié à des interprétations et applications sémantiques.

Il existe d'autres propriétés désirables comme la complexité temporelle et spatiale, voir [18, 19, 20].

5. Méthodes de Clustering

Plusieurs méthodes de Clustering apparaissent selon la définition d'un cluster et selon d'autres critères. Dans cette section, nous allons introduire certaines d'entre elles.

5.1. Méthodes par partitionnement

C'est les méthodes les plus simples et les plus fondamentales de Clustering. Ces méthodes organisent les objets dans des clusters exclusifs. Le nombre de clusters k est connu a priori, celui-ci est le paramètre du démarrage de l'algorithme de partitionnement ($k \leq n \mid n$ est le nombre d'objets), le processus se base sur la distance, plus deux objets sont proches, plus ils ont la possibilité d'être regroupés dans le même cluster [17].

Ces techniques [22, 23] sont basées typiquement sur un représentant (ou prototype), par conséquent le Clustering par partitionnement est également appelé un Clustering basé sur un prototype. Un cluster est un ensemble d'objets dans lequel chaque objet est plus proche (similaire) du prototype de son cluster. Ce prototype est un centroïde s'il est la moyenne de tous les objets du cluster. Si ce centroïde n'est pas significatif (ex., lorsque les objets ont des attributs catégoriques), alors il est appelé un médoïde, ce médoïde est un objet particulier (représentatif) du cluster [24].

Les méthodes basées sur un prototype sont généralement constituées de deux étapes fondamentales, la sélection des k prototypes initiaux (pour les k clusters), et l'itération de l'opération de raffinement des prototypes jusqu'à convergence. L'approche la plus représentative

est d'utiliser une méthode non déterministe dans la première étape suivie d'un excès de recherche dans la deuxième étape. L'algorithme général de ce type est défini en algorithme 1 [22].

Algorithm 1 general prototype-based Clustering

1. Select K initial prototypes.
2. Refine prototypes until convergence.
    a. Find the closest prototypes for n points.
    b. Recompute cluster prototypes.

Il existe plusieurs algorithmes qui implémentent cette technique comme K-means, K-medoids, PAM (Partitioning Around Medoids), et CLARA (Clustering LARge Applications) [1, 11].

5.1.1. K-means

K-means est l'algorithme de Clustering le plus utilisé, il a une histoire riche et diversifiée, il a été découvert indépendamment en différents domaines scientifiques par Thorndike en 1953 [26] dans le terme de minimiser la moyenne des distances intra clusters, Steinhaus en 1955

[27], Lloyld en 1957 [28], Forgey en 1965, Ball et Hall en 1965, Jancey en 1965, et McQueen en 1967, et malgré qu'il a plus d'un demi-siècle, il reste l'un des algorithmes de Clustering les plus utilisés. La facilité de mise en œuvre, la simplicité, l'efficacité et le succès empirique sont les principales raisons de sa popularité [25].

L'algorithme commence par choisir k points représentatifs comme centroïdes initiaux des clusters. Chaque point est ensuite attribué au cluster du centre le plus proche de ce point en utilisant une fonction de proximité. Une fois que les clusters sont formés, l'algorithme met à jour les centres des clusters.

Il répète itérativement les deux dernières étapes jusqu'à ce que les centroïdes ne changent pas ou qu'un autre critère de convergence allégé soit satisfait (par exemple, jusqu'au moins 1% des points changent leurs clusters). La fonction d'objectif utilisée par k-means est l'indice de variance intra cluster SSE (voir section 6.1.5 chapitre 1) [23].

Algorithm 2 K-means Clustering

1. Select K points as initial centroids.
2. Repeat
3. Form K clusters by assigning each point to its closest centroid.
4. Recompute the centroid of each cluster.
5. Until convergence criterion is met.

5.2. Méthodes hiérarchiques

Comme l'indique leur nom, elles consistent à créer une hiérarchie de clusters, la construction de la hiérarchie est faite d'une manière graduelle, plus on monte dans la hiérarchie moins les groupes sont spécifiques. Le Clustering hiérarchique peut être représenté par un dendrogramme.

Selon la création de la hiérarchie, les algorithmes hiérarchiques se catégorisent en deux groupes : les algorithmes ascendants (Agglomerative), et les algorithmes descendants (Divisive).

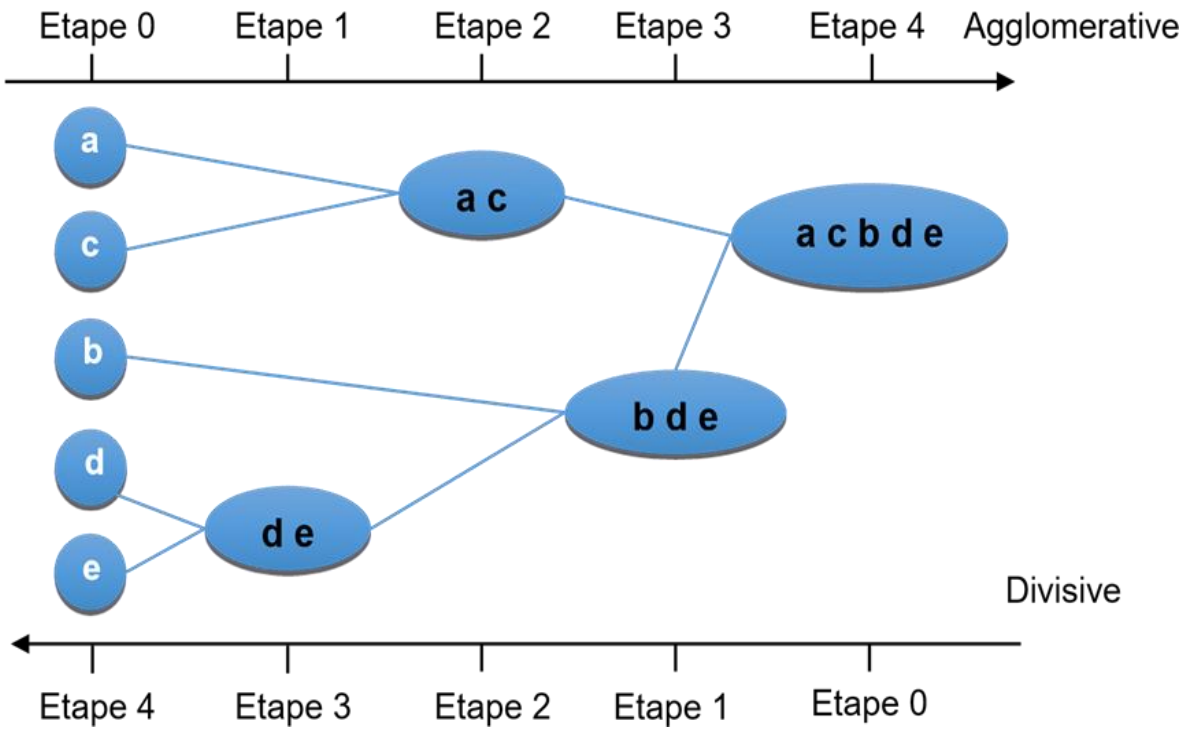

Figure 5 - Clustering hiérarchique.

L'avantage fondamental [37] d'avoir une méthode hiérarchique est qu'elle permet de couper la hiérarchie à n'importe quel niveau et d'obtenir les clusters de ce niveau. Cette fonctionnalité la rend significativement différente des méthodes de partitionnement, le fait qu'elle ne nécessite pas de paramètre k (nombre de clusters).

Les méthodes hiérarchiques [30,32] sont applicables à n'importe quel type d'attribut, elles donnent aussi une facilité pour traiter différentes formes de distance ou de similarité entre objets, elles utilisent la matrice de proximité standard, ou d'autres matrices spécifiques, cependant, elles souffrent du problème du choix du critère d'arrêt.

5.2.1. Les méthodes ascendantes (Agglomerative methods)

L'approche ascendante (ou bien l'approche bottom-up) consiste à créer la hiérarchie des clusters en commençant par n objets. Chaque objet représente un cluster, successivement à chaque étape cette méthode fusionne les objets ou les clusters les plus proches jusqu'à l'obtention d'un seul groupe qui contient tous les objets ou la satisfaction de la condition d'arrêt [17].

L'approche ascendante souffre d'un inconvénient majeur, une fois une partition est faite, un objet ne peut pas être déplacé à un autre groupe [29].

Algorithm 3 Agglomerative Hierarchical Clustering

1. Compute the dissimilarity matrix between all the data points.
2. **Repeat**
3. Merge Clusters as $C_{a\cup b} = C_a \cup C_b$. Set new cluster's cardinality as $N_a \cup_b = N_a + N_b$.
4. Insert a new row and column containing the distances between the new cluster $C_{a\cup b}$ and the remaining clusters.
5. **Until** Only one maximal cluster remains or satisfaction of stop criterion.

La plupart des algorithmes hiérarchiques agglomératifs [3] utilisent l'une des approches suivantes pour calculer la distance entre-clusters.

1. Single-link

Single-link (lien simple), cette approche a été proposée [32, 33, 34] par Florek et al. en 1951, puis par Kruksal en 1956, et Sneath et McQuitty en 1957. Williams, Lambert et Lance (1966) [35] ont généralisé l'algorithme single-link au processus hiérarchique 'nearest-neighbour'. La distance entre deux clusters est la distance minimale des distances entre chaque paire de clusters, pour deux clusters C_i, C_j la distance entre eux est donnée par :

$$\text{dist}(C_i, C_j) = \min_{i \in C,\, j \in C_j} d(i, j) \qquad \#(17)$$

2. Complete-link

Complete-link (lien complet) [33, 37] a été proposée en 1943 par Horn, Sørensen en 1948 [38], et King en 1967.

MacNaughton 1965 [39] a proposé sa version hiérarchique 'furthestneighbour'. Elle calcule la distance entre deux clusters comme étant la distance entre leurs plus loin membres.

$$\text{dist}_{CL}(C_i, C_j) = \max_{i \in C_i, j \in C_j} d(i, j) \qquad \#(18)$$

3. Average-link

Average-link (lien moyen) [33, 40] cette approche est aussi connue comme UPGMA (Unweighted Pair Group Method using Average), elle a été proposée par Sokal et Michener en 1958 [41]. Cette approche [23] a réduit les inconvénients associés aux approches précédentes, elle considère la distance entre tous les points des deux clusters.

$$\mathrm{dist}_{UPGMA}(C_i, C_j) = \frac{1}{n_i * n_j} \sum_{i \in C_i} \sum_{j \in C_j} d(i,j) \qquad \#(19)$$

n_i et n_j sont respectivement les cardinalités des clusters C_i et C_j. Par conséquent cette méthode est très coûteuse, surtout lorsque le nombre d'objets devient important.

4. Average-link pondérée

McQuitty a proposé une autre approche en 1966 [43], elle est connue par WPGMA (Weighted Pair Group Method using Average), cette approche est une version pondérée de la précédente, la distance entre un cluster Cl et un autre cluster constitué de deux petits clusters Ci et Cj est donnée alors par :

$$\mathrm{dist}_{WPGMA}(C_l, C_i \cup C_j) = \frac{n_i}{n_i + n_j} \mathrm{dist}_{WPGMA}(C_l, C_i) + \frac{n_j}{n_i + n_j} \mathrm{dist}_{WPGMA}(C_l, C_j) \qquad \#(20)$$

5. Méthode de centroïde

La méthode de centroïde [14, 40, 37], connue également sous le nom UPGMC (Unweighted Pair Group Method using Centroid), elle a été proposée en 1958 par Sokal et Michener.

Cette méthode calcule la distance entre deux clusters comme la distance entre leurs centroïdes c_i et c_j.

$$\mathrm{dist}\,(C_i, C_j) = d\,(c_i, c_j) \qquad \#(21)$$

Tel que le centroïde $c_i = \frac{1}{n_i} \sum_{i \in C_i} i$ Après la jonction de deux clusters, le nouveau centroïde est donné par la moyenne pondérée [29] :

$$c_{C_i \cup C_j} = \frac{n_i c_i + n_j c_j}{n_i + n_j} \qquad \#(22)$$

6. Méthode de médiane

Gower [33, 14] a proposé une autre méthode basée sur les centroïdes appelé la méthode de médiane en 1967 [44], elle est connue également sous le nom WPGMC (Weighted Pair Group

Method using Centroid), cette méthode a été proposée pour réduire les inconvénients de la méthode UPGMC. Si la taille des deux clusters à fusionner est différente, alors le nouveau centroïde des deux clusters fusionnés sera plus proche du cluster qui a plus d'objets, et il peut rester à l'intérieur de ce groupe. Dans la méthode de médiane le nouveau centroïde est calculé indépendamment de la taille de clusters, la distance entre un cluster C_l et un autre cluster constitué de deux autres clusters C_i et C_j est donnée par :[28]

$$dist_M\left(C_l, C_i \cup C_j\right) = \frac{1}{2} dist_M(C_l, C_i) + \frac{1}{2} dist_M\left(C_l, C_j\right) - \frac{1}{4} dist_M\left(C_i, C_j\right) \qquad \#(25)$$

7. Méthode de ward

La méthode de Ward [45] a été proposée pour calculer la distance entre deux clusters pendant le processus d'un Clustering ascendant (agglomératif), L'objectif de la méthode de Ward est de minimiser l'augmentation de la valeur du critère SSE du Clustering obtenu en fusionnant deux clusters. L'union de toutes les paires possibles de clusters est considérée, et les deux clusters dont leur fusionnement entraîne l'augmentation minimale de la valeur du SSE seront fusionnés. Le critère de Ward peut être interprété comme le carré de la distance Euclidienne entre les centroïdes des deux clusters à fusionner pondéré par un facteur proportionnel aux cardinalités des deux clusters [40]:

$$W\left(C_i, C_j\right) = \frac{n_i * n_j}{n_i + n_j} d^2\left(c_i, c_j\right) \qquad \#(23)$$

8. Formule de Lance et Williams

La formule de Lance & Williams [47] donne la distance entre un cluster C_l et un autre C_{ij} formé par le fusionnement de deux clusters C_i et C_j comme suit :

$$D(C_l,C_{ij}) = \alpha_i D(C_l,C_i) + \alpha_j D(C_l,C_j) + \beta D(C_i,C_j) + \gamma \left| D(C_l,C_i) - D(C_l, C_j) \right| \qquad \#(24)$$

Tels que (C_i, C_j) est la distance entre les clusters C_i et C_j, et α_i, α_j, β, et γ sont des coefficients.

5.2.2. Les méthodes descendantes (Divisive methods)

L'approche descendante (ou bien l'approche top-down ou l'approche incrémentale) [46] commence par un cluster qui contient tous les objets. Habituellement à chaque itération successive le nombre de clusters augmente par un, voir (Figure 4). Successivement à chaque itération un cluster est divisé en deux plus petits clusters, jusqu'à l'état où chaque cluster contient un seul objet ou la satisfaction d'un critère d'arrêt.

Cette méthode a l'avantage d'être plus efficiente que la méthode ascendante, spécialement quand il n'est pas nécessaire de gérer la hiérarchie complète. Elle est considérée comme une

approche globale car elle contient toutes les informations avant de diviser les données. Kauffman et Rousseeuw ont signalé qu'on peut relever la structure principale de données du résultat produit par l'approche [37, 23].

La méthode descendante souffre des inconvénients majeurs suivants : la complexité du scindement d'un cluster en deux sous-clusters, $2n-1-1$ sont les scindements possibles pour diviser un cluster qui contient n objets en deux sous-clusters [61].

Scott et Symons [48] ont proposés un algorithme optimisé qui requit l'examination de juste

$2d-2$ divisions par l'affectation d'objets à l'hyperplan, tel que d est le nombre d'attributs.

Cependant pour les données de basses dimensions cet algorithme consomme beaucoup de temps aussi. En effet le problème de trouver une bipartition optimale pour certains critères de Clustering reste un problème NP-Hard [34, 49, 38, 39].

Cette approche souffre aussi du même inconvénient de l'approche ascendante, une fois une partition est faite, un objet ne peut pas être déplacé à un autre groupe. Un autre problème surgit dès la deuxième itération, c'est le choix du cluster à scinder [14, 29].

Il est possible [12] de construire un algorithme divisif qui n'a pas besoin de considérer tous les scindements possibles comme la stratégie monothétique définie ci-après. Des algorithmes divisifs de complexité polynomiale ont été proposés, mais avec aucune garantie d'optimalité [46, 48].

Il existe deux stratégies de la méthode divisive : monothétique (aussi connue comme analyses d'associations), et polythétique. Un cluster monothétique est un groupe dans lequel tous les objets ont la même valeur d'une variable ou presque la même, la division d'un groupe en deux sous-groupes se base sur un seul attribut (variable), tandis que l'approche polythétique utilise tous les attributs pour scinder un groupe.

Si les données sont binaires l'approche monothétique peut être facilement appliquée, en se basant sur l'absence ou la présence d'un attribut. Généralement la variable qui maximise l'indice de chi-square est choisie pour la division d'un cluster, Lambert et Williams et MacNaughton, Williams, Dale, et Mocket ont décrit la procédure la plus couramment utilisée pour la déterminer [29, 37, 33, 52].

Notons que la spécification des méthodes monothétiques et polythétiques pour juste les méthodes hiérarchiques divisives n'est pas évidente, les méthodes agglomératifs sont des méthodes polythétiques. Les deux méthodes divisive et agglomérative aussi ne sont pas

spécifiques juste pour le Clustering hiérarchique, dans le Clustering non-hiérarchique, aussi, il existe ces deux méthodes. Pour plus de détails voir [55, 53].

Ces méthodes ont été moins courantes que les méthodes ascendantes. Pour un exemple d'application d'une méthode descendante, voir [47], [46], [55].

Algorithm 4 Basic Divisive Hierarchical Clustering

1. Start with the root node consisting by all the data points.
2. **Repeat**
3. Split parent node into two parts C_1 and C_2 using Bisecting K-means to maximize Ward's distance ($C_{1,2}$).
4. Construct the dendogram. Among the current, choose the cluster with the highest squared error.
5. **Until** Singelton leaves are obtained or satisfaction of stop criterion.

5.3. Méthodes basées sur la densité

Dans l'espace de données, il existe des parties dans lesquelles les points sont très denses, séparées par des parties de faible densité. La plupart des méthodes de Clustering regroupent les objets en se basant sur la distance entre objets. Telles méthodes ne peuvent trouver que des clusters sphériques et rencontrent des difficultés à découvrir des clusters de formes arbitraires.

D'autres méthodes de Clustering ont été développées, qui se basent sur la densité, ces méthodes peuvent trouver les régions de forte densité dans l'espace de données, chacune des régions étant prise pour signifier un cluster différent. Leur idée générale est qu'un cluster continu à grandir tant que la densité (nombre d'objets) dans le voisinage ne dépasse un certain seuil.

Un certain nombre de ces techniques ont leurs origines dans des méthodes, de Carmichael et al. [56] et Carmichael et Sneath [57], développées pour surmonter l'inconvénient de liaison, l'un des principaux inconvénients de la méthode single-link. Les clusters sont initialement formés comme la méthode single-link, mais avec des critères pour empêcher l'addition d'objets qui sont beaucoup plus éloignés du dernier objet ajouté au cluster, les objets, rejetés de cette façon, initient de nouveaux clusters [55, 37].

La méthode classique qui repose sur la recherche de régions denses (ou modes) est celle de Wishart 1969 [34]. Cette méthode a été proposée pour le problème d'enlèvement de bruits afin d'être utilisée par l'algorithme single-link.

La recherche de modes est faite en considérant une petite région locale de volume Vn autour de chaque point, et on calcule le nombre d'objets (i) tombant dans cette région pour chaque objet. Les objets sont alors étiquetés comme étant denses ou non selon que leur voisinage Vn contient plus ou moins de points que la valeur du seuil k, ce seuil est mis à une valeur dépendante de la taille de l'ensemble de données n.

Les points non denses sont alors enlevés et les points denses sont groupés par la méthode single-link. Après avoir classifié les points denses, chaque point non dense est ajouté au cluster approprié. L'auteur de cette méthode [46] a proposé sa version hiérarchique afin de réduire les contrôles externes [37, 34].

Deux approches générales non-paramétriques proches de la méthode de Wishart ont été utilisées pour la définition de la densité, l'approche de Parzen, proposée en 1962 [58] et appliquée au problème de classification par Specht en 1967 [78], et l'approche des plus proches voisins introduite par Fix et Hodges [58, 59] est explorée par Patrick [60].

Les paramètres à définir ici sont spécifiés en fonction de n. La première approche définit une valeur fixe de volume Vn autour des objets, et calcule pour chaque point x sa valeur (x) de nombre de points tombants dans son voisinage de taille Vn. La probabilité de la densité estimée au point x est définie comme suit [4, 61]:

$$\hat{\rho}(x) = \frac{k(x)}{n * V_n} \qquad \#(25)$$

Tandis que la deuxième approche donne une valeur fixe au k et calcule pour chaque objet la taille de son voisinage (x) pour qu'elle contienne k objets. La probabilité de la densité estimée au point x est définie comme suit [4, 1, 37, 61]:

$$\hat{\rho}(x) = \frac{k}{n * V_n(x)} \qquad \#(26)$$

Un point est un mode ou dense dans l'algorithme de Wong et Lane [62] s'il possède une probabilité supérieure à un certain seuil $\rho*$ [4, 37, 62]:

$$\{x \,|\, \hat{\rho}(x) \geq \rho*\}$$

La deuxième approche est plus satisfaisante dans les régions clairsemées de la distribution, tandis que la première produira des densités unitaires non-discriminantes pour tous les objets [21].

Les deux approches ont été largement étudiées en [63].

Après cette brève introduction de l'historique de ce type de Clustering, on peut dire que ces méthodes considèrent les clusters comme des régions denses séparées par des régions moins denses, les clusters augmentent vers n'importe quelle direction, la densité continue tant qu'elle ne dépasse pas un seuil.

De ce fonctionnement, les clusters ont des formes arbitraires, et ont la protection contre les bruits. Généralement, la densité se définit comme le nombre d'objets dans le voisinage d'un objet particulier. Les méthodes basées sur la densité ont une bonne scalabilité et ne nécessitent aucun paramètre pour spécifier le nombre de clusters.

Ces propriétés exceptionnelles s'accompagnent de certains inconvénients. Un cluster peut contenir deux sous-clusters de très différentes densités, les deux excèdent le seuil. Le deuxième inconvénient découle des paramètres, une petite différence de valeur d'un paramètre peut produire des clusters différents. Le troisième est le manque d'interprétabilité [18, 16].

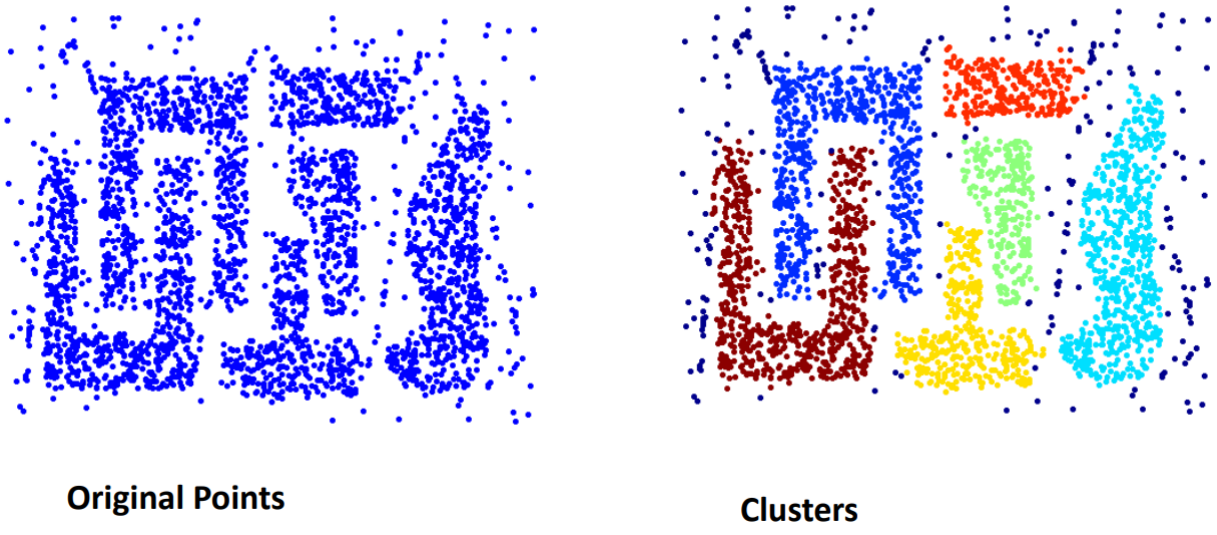

Figure 6 - Clustering par densité.

La (Figure 6) représente le Clustering de données par densité dans un espace bidimensionnel, chaque couleur représente un cluster différent.

Il existe deux types de méthodes basées sur la densité [16, 18]:

- Les méthodes connectives
- Les méthodes basées sur une fonction de densité

5.3.1. Les méthodes connectives

Dans ces techniques, il existe deux cruciaux concepts, la densité et la connectivité, elles sont mesurées en fonction de la distribution locale des plus proches voisins. La connectivité est une relation d'équivalence. DBSCAN et OPTICS sont deux exemples de ces méthodes [18].

DBSCAN

Cet algorithme a été proposé par Ester et al. [64] En 1996 pour découvrir des clusters de formes arbitraires. Afin de définir le fonctionnement du DBSCAN (Density-Based Spatial Clustering of Applications with Noise), il faut d'abord définir certains concepts. Deux paramètres à définir dans cet algorithme EPS et $MinPts$. EPS (Epsilon) ce paramètre représente la distance considérée pour définir le voisinage d'un point (c.-à-d. Les éléments proches d'un élément suivant cette distance sont ses voisins) :

$$\upsilon_{EPS(p)} = \{q \in X \mid dist(p,q) \leq EPS\}$$

Le deuxième paramètre est le nombre de points minimal $MinPts$ dans le voisinage d'un point pour le considérer comme étant noyau (ou core). Le DBSCAN est défini par l'algorithme suivant :

Algorithme5 DBSCAN algorithm

Begin

1. Select an arbitrary point p.
2. If $|\upsilon EPS(p)| \geq MinPts$ then a new cluster is formed with p as core. Otherwise p is border or noise
3. If p is a core object, then collect all density-reachable objects from p to extend the current cluster. If p is classified in another cluster, then merge the two clusters into one cluster.
4. If p is not a core object, then a previously non visited new object is selected.
5. The process terminates when all objects are visited

End

5.3.2. Les méthodes basées sur une fonction de densité

Comme l'indique leur nom, ces méthodes utilisent une fonction de densité afin de calculer la densité. La densité totale est définie comme la somme des fonctions de densité de tous les objets, les clusters sont déterminés par des attracteurs de densité, ces attracteurs sont des maximums locaux de la fonction de densité, DENCLUE [65] est un exemple de ces méthodes [16].

DENCLUE

(DENsity-based Clustering) a été proposé par Hinneburg et keim [3], la fonction de densité est la somme des fonctions d'influence, un exemple de fonction d'influence pourrait être la fonction d'onde carrée, cette fonction est définie comme suit [84, 37]:

$$f_{Square}(x,y) = \begin{cases} 0, & si\ d(x,y) > \sigma \\ 1, & sinon \end{cases} \qquad \#(27)$$

La fonction de densité d'un point x est alors définie comme suit :

$$f(x) = \sum_{y \in X} f_{Square}(x,y) \qquad \#(28)$$

6. Mesures d'évaluation et de performance

Dans la littérature, une grande variété d'algorithmes a été proposée pour différentes applications et tailles d'ensemble de données. Tant que le Clustering de données est un processus non supervisé, alors pas de classes prédéfinies ni d'exemples pouvant montrer la validité des résultats obtenus [3, 67, 68]. Donc il est important d'utiliser des mesures spéciales d'évaluation et de performance pour mesurer la qualité d'un résultat de Clustering.

Les mesures d'évaluation sont majoritairement utilisées pour juger la qualité d'un résultat de Clustering, alors que les mesures de performance sont appliquées pour comparer l'efficacité des algorithmes en utilisant le temps de calcul et/ou le nombre des évaluations des fonctions d'objectif et de contrainte [69].

Il existe trois objectifs [69]: la séparabilité, la connectivité, et la compacité.

La séparabilité des clusters signifie qu'ils sont séparables deux à deux, pour chaque pair de clusters, il existe un hyperplan séparant les deux clusters dans l'espace d-dimensionnel.

La connectivité est le degré auquel les objets adjacents sont placés dans le même cluster, ce degré est défini par un algorithme de voisinage, les plus utilisés sont : KNN (k plus proches voisins), ε-neighborhood, et le NC algorithme.

La compacité d'un cluster est caractérisée par le degré de densité des objets autour des centres des clusters.

6.1. Indices de validité des clusters

La qualité d'un résultat de Clustering peut être évaluée par des indices de validité, il existe trois types d'indices de validité : le premier représente les indices de validité externes, dans ce type, l'évaluation des résultats se base sur une structure pré-spécifiée imposée sur l'ensemble de données X, et reflète notre intuition sur elle.

Dans le deuxième type, l'évaluation de Clustering se base sur le terme de quantités qui entrelace les vecteurs de l'ensemble de données eux-mêmes (par ex. la matrice de proximité), ce type est appelé les validations internes. Dans le dernier type l'évaluation se fait par la comparaison des structures obtenue par le même algorithme appliqué au même ensemble de données mais avec différentes valeurs de paramètres, ou bien par différents algorithmes de Clustering, ce type est appelé les validations relatives [70].

6.1.1. Indice de Davies et Bouldin

Cet indice [71] peut être défini comme une combinaison de mesures d'homogénéité et de séparation :

- **L'homogénéité :**

$$H_i = \frac{1}{n_i}\sum_{x \in C_i} d(x, \mu_i) \qquad \#(29)$$

Tels que : μ_i est le centre du cluster i, et (x, μ_i) est la distance entre l'objet x et le centre μ_i, et n_i la taille du cluster i.

- **La séparation :**

$$Si, = (\mu_i, \mu_j) \qquad \#(30)$$

Tel que : (μ_i, μ_j) est la distance entre le centre du cluster i et le centre du cluster j.

La fonction du Davies et Bouldin [90] alors est définie comme :

$$DB = \frac{1}{k}\sum_{i=1}^{k} DB_i \qquad \#31$$

Tel que : $$DB_i = \max_{1 \le j \le k, i \ne j} \frac{H_i + H_j}{S_{i,j}}$$

6.1.2. Indice de Dunn

L'indice de Dunn [72] est désigné pour trouver les clusters compacts et les biens séparés [73]:

$$D_{n_c} = \min_{i=1,\dots,n_c} \left\{ \min_{j=i+1,\dots,n_c} \left(\frac{d(c_i, c_j)}{\max_{k=1,\dots,n_c} diam(c_k)} \right) \right\} \quad \#32$$

Tel que :

- (c_i, c_j) est la distance entre deux clusters c_i et c_j définie comme :

$$d(c_i, c_j) = \min_{x \in C_i, y \in C_j} d(x, y) \quad \#33$$

- diam (c_k) est le diamètre du cluster c_k défini comme :

$$diam(C) = \max_{x,y \in C} d(x, y) \quad \#34$$

6.1.3. Indice de coefficient de silhouettes

Cet indice peut être aussi défini comme une combinaison de deux mesures, la cohésion et la séparation :

- **La cohésion :**

$$C(x_i) = \frac{1}{n_i - 1} \sum_{y \in C_i, y \neq x_i} d(x_i, y) \quad \#35$$

- **La séparation :**

$$Sp(x_i) = \min_{i \neq j} \frac{1}{n_i} \sum_{y \in C_j} d(x_i, y) \quad \#36$$

Tel que : (x, y) est la distance entre l'objet x et y, et n_i la taille du cluster i.

L'indice de coefficient de silhouette [75, 74] est défini comme :

$$S(k) = \frac{1}{n} \sum_{i=1}^{n} S(x_i) \quad \#37$$

Tel que : n est la taille de l'ensemble de données, et k le nombre des clusters, et (x_i) est définie comme suit :

$$S(x_i) = \frac{Sp(x_i) - C(x_i)}{\max(C(x_i), Sp(x_i))} \quad \#38$$

Cet indice est utilisé pour sélectionner la meilleure valeur du k (nombre des clusters), par le choix du k qui majoré l'indice (k) [74].

Ces trois premiers indices sont internes.

6.1.4. Indice de Fowlkes-Mallows

Cet indice a été proposé par Fowlkes et Mallows [76] comme une mesure pour comparer deux résultats de Clustering hiérarchique. Cependant, il est possible de l'utiliser pour des résultats de Clustering non-hiérarchique. Pour bien définir cet indice, il faut d'abord définir les notions suivantes [77]:

- X l'ensemble d'objets de taille $|X| = n$.
  - $C = \{C_{1,2}, \ldots, C_k\}$ est un ensemble non-vide de sous-groupes disjoints de X tel que l'union de tous les sous-groupes donne à X: $\bigcup_{i=1}^{k} C_i$
- $C' = \{C'_1, C'_2, \ldots, C'_p\}$ est un autre ensemble de partitions d'un autre Clustering du même ensemble X.
- $M = (m_{ij})$ est la matrice de confusion des paires C, C', m_{ij} est le nombre des éléments de l'intersection des deux clusters C_i et C'_j:

$$m_{i,j} = |C_i \cap C'_j|,\ 1 \le i \le k,\ 1 \le j \le p \qquad \#(39)$$

L'indice de Fowlkes et Mallows peut alors être défini comme suit [76, 77] :

$$FM(C, C') = \frac{\sum_{i=1}^{k}\sum_{j=1}^{p} {m_{i,j}}^2 - n}{\sqrt{(\sum_i |C_i|^2 - n)(\sum_j |C'_j|^2 - n)}} \qquad \#(40)$$

Cet indice est un indice relatif.

6.1.5. Indice de variance intra cluster (SSE)

Cet indice est basé sur le concept de la minimisation des distances entre chaque centre de cluster et les autres objets du même cluster (distance intra cluster) [78, 28] :

$$\mathrm{Var}(C) = \sum_{Ci \in C} \sum_{x \in Ci} d\,(x - \mu_i)^2 \qquad \#\ (41)$$

Cet indice est un indice interne.

6.1.6. Indices externes

Pour les indices externes, il nous faut une connaissance des vrais résultats (on a les objets de chaque classe).

- **L'homogénéité :** est l'indice qui vérifie si tous les objets d'un cluster sont de la même classe, il est donné par :

$$Homogeneity = 1 - \frac{H(C|L)}{H(C)} \qquad \#(42)$$

- **La complétude :** cet indice vérifie si tous les objets d'une classe sont assignés au même cluster, cet indice est donné par :

$$Completness = 1 - \frac{H(L|C)}{H(L)} \qquad \#(43)$$

Où $(C|L)$ est l'entropie conditionnelle des classes compte tenu des assignations des clusters, elle est donnée par :

$$H(C|L) = -\sum_{i=1}^{k}\sum_{j=1}^{q}\frac{n_{i,j}}{n}.\log(\frac{n_{i,j}}{n_j})$$

(C) représente l'entropie des classes, il est donné par :

$$H(C) = -\sum_{i=1}^{k}\frac{n_i}{n}.\log(\frac{n_i}{n})$$

Où C représente l'ensemble des classes et L représente l'ensemble des clusters, k est le nombre des clusters, q est le nombre des classes, n est le nombre total des objets, n_i et n_j sont respectivement la taille de la classe i et la taille du cluster j, et n_i, représente le nombre des objets de la classe i assignés au cluster j.

L'entropie conditionnelle des clusters compte tenu des assignations des classes $(L|C)$ et l'entropie des clusters (L) sont définies symétriquement.

- **V Mesure :** cet indice est défini par les deux indices précédents comme suit :

$$VMeasure = 2.\frac{Homogeneity.\ Completness}{Homogeneity + Completness} \qquad \#(44)$$

- **La pureté :** est le degré de points de données qui sont classifiés correctement, elle est donnée par:

$$Purity = \frac{1}{n}\sum_{i=1}^{k} n_{i,j} \qquad \#(45)$$

- **La précision** : cet indice est donné par :

$$Precision(i,j) = \frac{n_{i,j}}{n_j} \qquad \#(46)$$

- **Rappel :** cet indice est donné par :

$$Recall(i,j) = \frac{n_{i,j}}{n_i} \qquad \#(47)$$

- **G Mesure :** cet indice est donné par :

$$(i,j) = \sqrt{Precis\,(i,j) * Reca(i,j)} \qquad \#(48)$$

- **F-Measure :** cet indice est donné par :

$$F - measure = \frac{1}{n}\sum_{j=1}^{k} n_j \max_{t=1\ldots q} \mathrm{F}(L_j, C_t) \qquad \#(49)$$

Tel que : $\mathrm{F}(L_j, C_t) = \frac{2*Precision(i,j)*Recall(i,j)}{Precision(i,j)+Recall(i,j)}$

Une grande valeur de ces mesures est nécessaire pour un bon Clustering.

6.2. Les profils de performance dans l'analyse des clusters

Le Clustering est un problème d'optimisation globale, et la capacité d'un algorithme de trouver la solution globale ou une solution proche du global est très important en tant que telles solutions fournissent la meilleure structure des clusters d'un ensemble de données avec le nombre minimal des clusters.

Par conséquent nous allons introduire brièvement les profils de performance pour la comparaison des algorithmes de Clustering. Les profils sont définis en utilisant trois paramètres : le temps de calcul, la précision, et le nombre des évaluations de la fonction de distance [69].

6.2.1. Le temps de calcul

Supposons l'ensemble de solveurs S appliqués sur un ensemble de problèmes P, on définit :

$t_{p,s}$ = Le temps de calcul nécessaire pour résoudre le problème $p \in P$ par le solveur $s \in S$.

Ici on utilise un ratio $r_{p,s}$ de performance pour comparer la performance du solveur s à la résolution du problème p :

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}} \qquad \#(50)$$

On suppose le paramètre $r_M \geq r_{p,s}$ pour tout p, s est choisi, et $r_{p,s} = r_M$ si et seulement si le solveur s n'a pas résolu le problème p. La performance globale d'un solveur s est définie par [79]:

$$\rho_s(\tau) = \frac{1}{|P|} size\{p \in P : r_{p,s} \leq \tau\} \qquad \#(51)$$

Cette performance est la probabilité d'un solveur $s \in S$ à résoudre l'ensemble P tel que $r_{p,s}$ est inférieure à $\tau \in \mathbb{R}$ [79].

6.2.2. La précision

La précision d'un algorithme de Clustering est déterminée par l'utilisation des meilleures solutions globales connues [88]. Supposons que : $f^*_{s,p}$ est la meilleure solution connue de la fonction objective, et $f_{s,p}$ la solution d'un solveur $s \in S$ d'un problème $p \in P$, un profil de performance est décrit par le ratio de précision :

$$r_{p,s} = \frac{f_{s,p}}{f^*_{s,p}} \qquad \#(52)$$

Comme il est défini dans le temps de calcul, $r_M \geq r_{p,s}$ donc la probabilité d'un solveur $s \in S$ à résoudre l'ensemble P avec la précision $\tau \in [0, r_M]$ est donnée par :

$$\rho_s(\tau) = \frac{1}{|P|} size\{p \in P : r_{p,s} \leq \tau\} \qquad \#(53)$$

6.2.3. Nombre des évaluations de la fonction de distance

On Suppose l'ensemble de données X qui contient n objets de d attributs. Il est attendu que le nombre des évaluations de la fonction de distance d'un algorithme de Clustering dépende linéairement ou presque linéairement du nombre de clusters. Pour chaque itération le nombre des évaluations de la fonction de distance peut être décrit comme O (n_K), où k est le nombre des clusters, donc si l'algorithme de Clustering utilise M itérations pour résoudre un problème $p \in P$, alors il est attendu que le nombre soit comme suit [69] :

$$(n, k) = cMnk \qquad \#(54)$$

Tel que : c est une constante supérieure à zéro. Ici le ratio est défini comme :

$$r_{p,s} = \frac{N_p^s}{N(p,k)} \qquad \#(55)$$

Tel que : N_p^s est le nombre des évaluations de la fonction de distance du solveur s pour résoudre le problème p, et N (p, k) est le nombre des évaluations de la fonction de distance du problème p calculé en utilisant (50).

Ici on définit l'ensemble $\mathcal{Q}_s$ des problèmes résolus par le solveur s, et V l'ensemble des solveurs succédant à résoudre au moins un problème, alors on définit le profil de performance d'un solveur s comme :

$$\rho_s(\tau) = \frac{1}{|P|} size\{p \in \mathcal{Q}_s : r_{p,s} \leq \tau\} \qquad \#(56)$$

Tel que $\tau \in (0, \tau_{max}]$, et $\tau_{max} = \max_{s \in V} \max_{p \in Qs} r_{p,s}$.

Pour tout solveur s n'ayant pas résolu au moins un problème, sa performance $P_s(\tau) = 0$ [69].

7. Conclusion

Dans ce chapitre, nous avons introduit, en détail, le Clustering et les méthodes utilisées dans la littérature pour le résoudre et leurs mesures de validité et de performance. Dans le chapitre suivant, nous allons présenter un nouvel algorithme CNOK, de classification automatique " non supervisé " sans K et sa conception.

Chapitre II

Conception

1. Introduction

La classification automatique appelée aussi classification non supervisée est une tâche se trouvant au carrefour de plusieurs domaines tels que : la classification, la fouille de données, la reconnaissance de formes, le scoring, etc.

Elle vise à découvrir un partitionnement de données qui représente une procédure de regroupement d'un ensemble d'observations dans plusieurs sous-ensembles homogènes et/ou bien séparés, initialement inconnu pour dégager de nouvelles connaissances.

Obtenir des classes claires et bien distinguées demande un travail énorme est n'est guère une tâche facile. Il s'agit d'un problème hautement combinatoire où le nombre de partitionnements possibles de N objets en k classe revient à envisager toutes les partitions possibles de la plus fine où tous les objets sont isolés, à la plus dense, où tous les objets sont regroupés dans une classe unique.

Les méthodes de partitionnement actuelles se trouvent forcées de demander en conséquence quelques paramètres d'initialisation comme le nombre de classes, le partitionnement initial ou le nombre d'itérations maximal, etc. qui permet de gagner un peu sur le nombre de possibilités nécessaires à examiner ; Ce qui rend ces méthodes partiellement supervisé.

Dans notre contribution nous proposons une nouvelle approche pour une méthode de partitionnement de N objets en k classes sans y avoir recourt à des paramètres d'initialisation préalable. Un algorithme ***« non supervisé »*** qui offre la possibilité de trouver automatiquement le nombre de classes dans les données.

2. Le Clustering CNo-K :

Dans cette section, nous présentons un nouvel algorithme de classification non supervisée de Clustering de données, qui permet de regrouper en K cluster distincts des éléments d'un ensemble de données D {e1, ..., en}, sans indication préalable du nombre de cluster K.

Chaque nouvel élément de l'ensemble de données peut former un nouveau cluster (s'il est considéré comme diffèrent) ou bien rejoindre un des clusters existants, selon la mesure de similarité entre ces attributs et les attributs des autres éléments appartenant aux autres clusters.

Nous voulons de nos clusters qu'ils soient :

- Resserrés sur eux-mêmes : deux points qui sont proches doivent appartenir au même cluster ;
- Loin les uns des autres : deux points qui sont éloignés doivent appartenir à des clusters différents.

L'algorithme prend en entrée l'ensemble de données et une mesure de similarité entre ces données, et produit en sortie un ensemble de partitions décrivant la structure générale de l'ensemble de données. Chaque C_i est un cluster qui représente une ou plusieurs caractéristiques de l'ensemble D.

Plus formellement, L'algorithme de Clustering prend (D, s), où D représente l'ensemble de données et S la mesure de similarité, et retourne une partition P = (C_1, C_2, ..., C_k) tel que les C_j (j = 1..k) sont des sous-ensembles de D qui vérifient :

$$\begin{cases} C_1 \cup C_2 \cup C_3 \ldots C_k = D \\ C_i \cap C_j = \Phi \quad si\ i \neq j \end{cases}$$

S est la mesure de similarité qui repose sur le calcul de distance entre deux données (la distance utilisée pour calculer la similarité dans notre algorithme est la distance euclidienne **#(01)**, sachant que chaque donnée est composée d'un ensemble d'attributs numériques. Plus la distance est importante, moins similaires sont les données et vice versa.

Cet algorithme découvre automatiquement les classes dans des données numériques sans connaître le nombre de classes a priori, sans partition initiale et sans paramétrage délicat. L'algorithme commence par l'initialisation, les N objets de D {e_1, ..., e_n} qui sont placés dans un vecteur de données de tailles N.

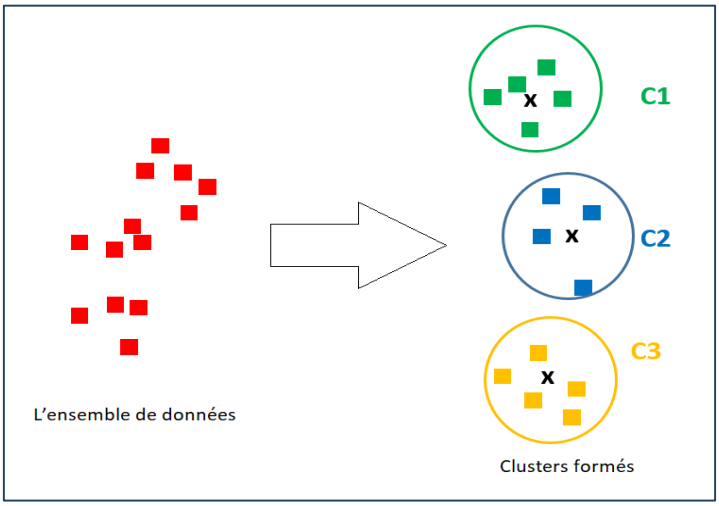

Figure11 Données à clusterer avec l'algorithme CNOK

La première étape peut être décrite comme suit :

- Initialiser le première cluster C_1, qui est créé à partir du premier élément e_1 de l'ensemble de données D, et il est considéré comme le premier élément de ce cluster et son point centroïde dans un premier temps.

- Pour chaque nouvel élément e_i (i>1) de D, l'élément e_i peut rejoindre le Cluster C_1 comme il peut être un élément totalement différents et doit être placer dans un nouveau cluster C_j et aussi considéré cet élément initialement comme le point centroïde de ce nouveau cluster.

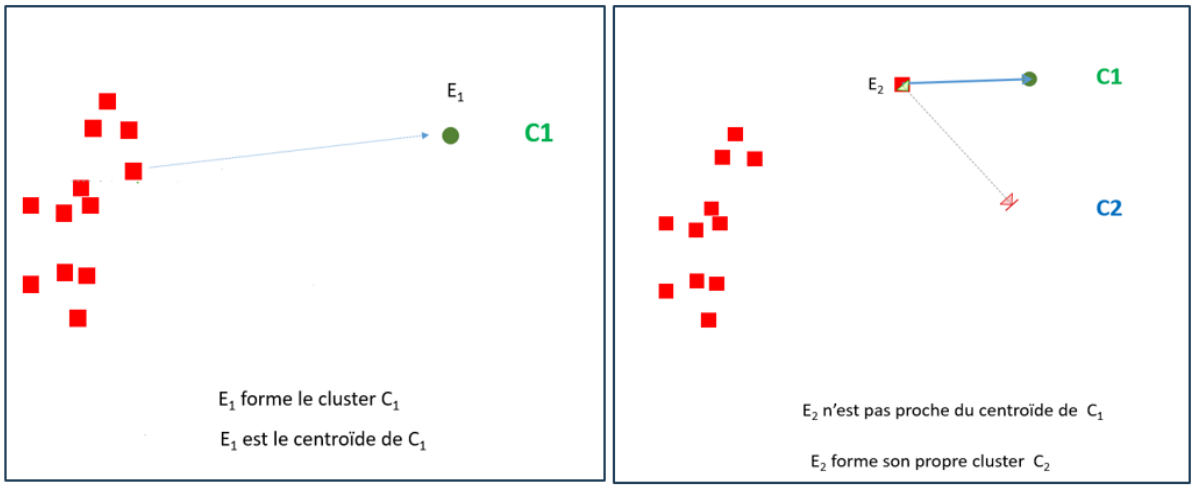

Figure 7- Formation des clusters avec l'algorithme CNOK

- Avant de chercher le cluster le plus proche pour un élément e_i, il faut d'abord que les points centroïdes des clusters créés soient réajustés. Si la distance entre ce nouvel élément e_i et le point centroïde de ce cluster est minime l'élément e_i le rejoint sinon il forme son propre cluster; refaire le même processus pour les N éléments de l'ensemble de données.

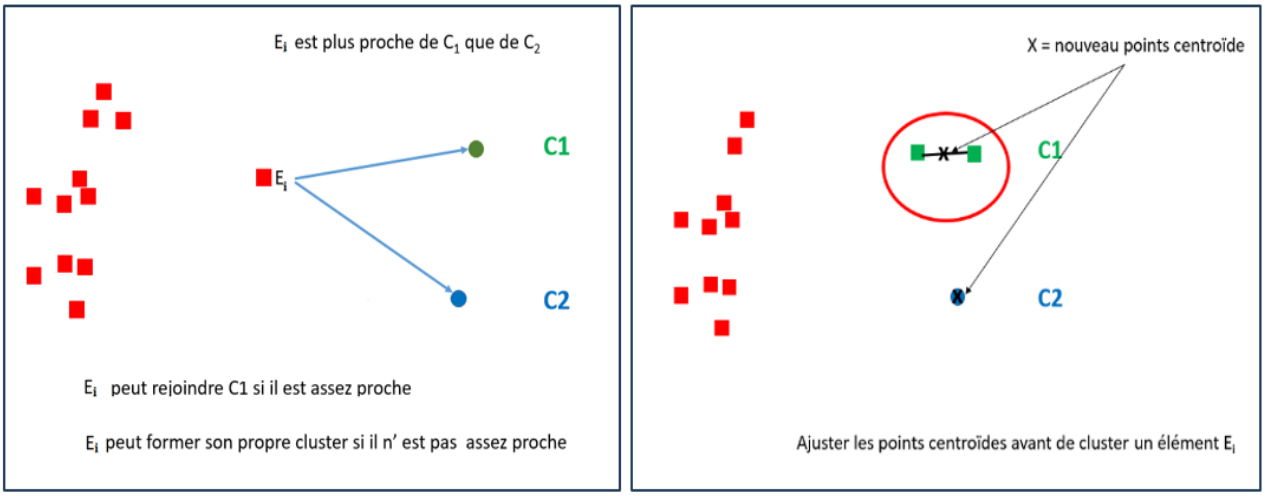

Figure 8- Ajuster les centroïdes avec L'algorithme CNOK

- À la fin de la première étape on obtient k cluster formés automatiquement peuplés avec m éléments différent dans chaque cluster et k points centroïdes pour k cluster créé (figure 7).

Pour la deuxième étape, et pour conclure le Clustering il faut:

- Vérifier que chaque élément e_i appartenant à un cluster C_K est réellement plus proche de son point centroïde que des points centroïdes des autres clusters en réajustent tous les points centroïdes des clusters crées puis recalculer la distance pour chaque élément avec les k point centroïdes, ensuite replacer les éléments dans le cluster le plus proches.

- L'algorithme s'arrête lorsqu'il y a une Stabilisation des centres de clusters. Nous obtenons alors N éléments partitionnés en K clusters homogènes et bien séparés.

Deux autres facteurs ont été aussi introduits dans notre algorithme ;

- ***Le premier facteur*** consiste à définir le critère d'acceptation ou de rejet d'un élément dans un cluster, il est défini par le calcul de la distance moyenne de toutes les distances entre éléments de l'ensemble de données. Ce critère est utilisé pour déterminer si l'élément à clustrer est assez proche d'un cluster et le rejoindre ou pas assez et forme son propre cluster.

- ***Le deuxième facteur*** et d'ajuster le point centroïde de chaque cluster après chaque élément placer dans un cluster (trouver le nouveau point centroïdes pour chaque cluster existant).

3. L'Algorithme CNo-K

L'algorithme n'a pas de paramètres à initialiser, (nombre de clusters, et le nombre maximum d'itérations), il s'arrête lorsqu'il y a une stabilisation des centres de clusters (les points centroïdes ne bougent plus lors des itérations). L'algorithme est défini comme suit :

Algorithme 10 CNo-K

Début

1. Calculer la distance moyenne des N éléments
2. Former C_1 à partir de l'élément e_1
3. **Pour** chaque nouvel élément e_i (i>1)
   - Ajuster les centroïdes des clusters existants
   - Trouver le cluster le plus proche à e_I
   - Si (la distance (e_i, centroïde $C_{j(\text{le plus proche})}$) <= distance moyenne)
     - Rejoindre le cluster le plus proche
   - Sinon l'élément e_i forme son propre cluster.

   Fin
4. **Répéter**
   - Ajuster les centroïdes pour chaque cluster.
   - e_I rejoint le cluster le plus proche

   Jusqu'à stabilisation des centres de clusters

Fin

Entrée : Dataset à clusterer ;

Sortie : Dataset clusterer.

4. UML (Unified Modeling Language)

UML (Unified Modeling Language), le langage de modélisation unifié, est un langage graphique capable d'exprimer les exigences des logiciels ou des systèmes, l'architecture et la conception. UML peut être utilisé pour la communication avec d'autres développeurs, des clients, et de plus en plus avec des outils automatisés qui génèrent des parties de systèmes [80].

L'UML est un standard relativement ouvert, contrôlé par OMG (Object Management Group), un consortium ouvert d'entreprises. L'UML est né de l'unification de nombreux langages de modélisation graphique orientés objet, qui ont été proposés à la fin des années 1980 et au début des années 1990. Depuis son apparition en 1997, l'UML est devenu la norme pour la modélisation graphique de logiciels [81].

L'UML, dans son état actuel, définit une notation et un méta-modèle. La notation est le truc graphique qu'on voie dans les modèles (la syntaxe graphique du langage de modélisation). Par exemple, la notation de diagramme de classe définit comment les éléments et les concepts, tels que la classe, l'association et la multiplicité, sont représentés. Un méta-modèle est un diagramme de diagramme, généralement un diagramme de classe qui définit les concepts du langage [81].

L'UML fournit un corps assez considérable de divers diagrammes qui aident à définir une application [81].

4.1. Diagrammes UML

UML est un langage de modélisation normalisé composé d'un ensemble intégré de diagrammes, la version 2.0 d'UML propose treize types de diagrammes officiels présentés dans (Figure 9). Ces types de diagrammes sont la façon dont beaucoup de gens abordent l'UML.

Ces diagrammes ont été développés pour aider les développeurs de systèmes et de logiciels à accomplir les tâches suivantes [80] :

- La spécification
- La visualisation
- La conception de l'architecture
- La construction
- La simulation et les tests
- La documentation

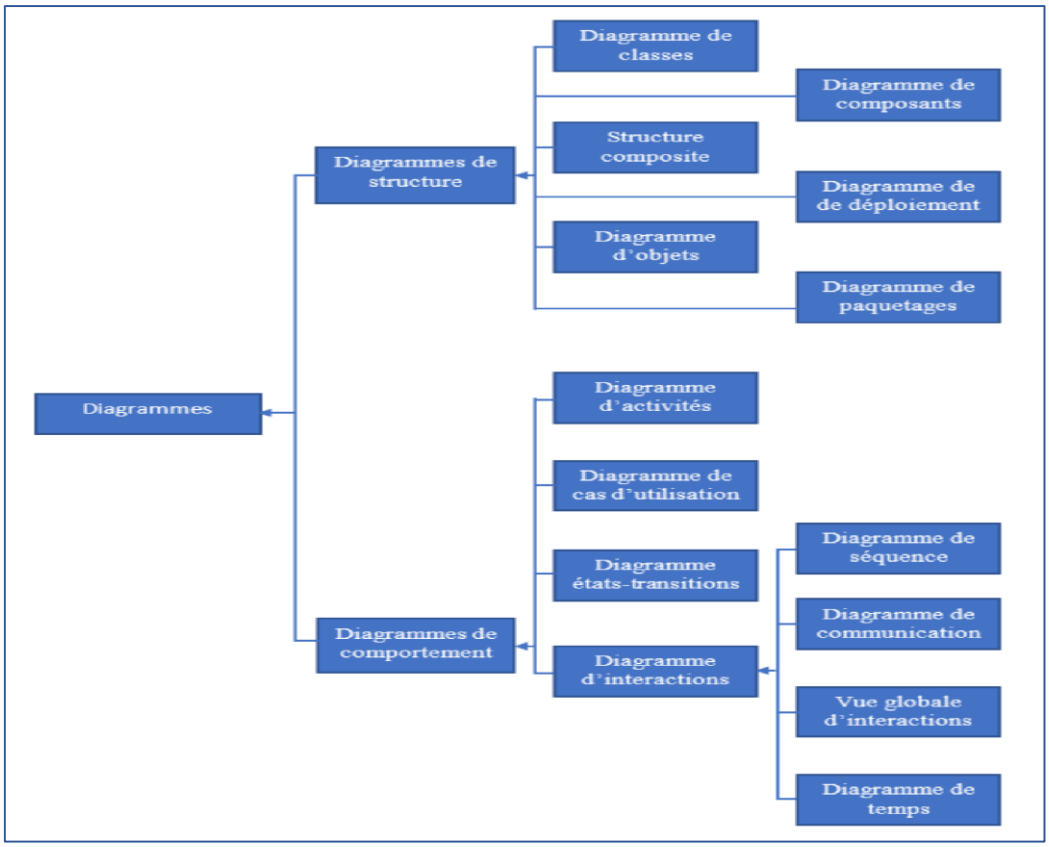

Figure 9 - Diagrammes d'UML.

Les diagrammes UML peuvent être classés en trois classes générales [80]:

- **Diagrammes structuraux** : ils sont utilisés pour montrer les éléments constitutifs du système, des caractéristiques qui ne changent pas avec le temps. Ces diagrammes répondent à la question, Qu'y a-t-il ?

- **Diagrammes comportementaux :** ces diagrammes sont utilisés pour montrer comment le système répond aux demandes ou évolue au fil du temps.

- **Diagrammes d'interactions :** un diagramme d'interaction est en fait un type de diagramme comportemental, ce type est utilisé pour représenter l'échange de messages au sein d'une collaboration pour atteindre un objectif.

Il existe d'autres façons de catégoriser les diagrammes, les trois catégories suivantes sont les plus populaires [80] : figure 10

- ❖ **Diagrammes statiques :** ces diagrammes montrent les caractéristiques statiques du système. Cette catégorie est similaire à celle des diagrammes structuraux. Ils comprennent les diagrammes de classes, de composants, de structure composite, de déploiement, d'objets, et de paquetages.

- ❖ **Diagrammes dynamiques :** ceux-ci montrent comment le système évolue au fil du temps. Les diagrammes de cette catégorie sont : diagramme d'états-transitions et de temps.

- **Diagrammes fonctionnels :** ils montrent les détails des comportements et des algorithmes. Ils répondent à la question, comment le système accomplit les comportements demandés. Cette catégorie comprend les diagrammes de cas d'utilisation, d'interactions, et d'activités.

- **Diagramme de classe :** est utilisé pour montrer des entités du monde réel, des éléments d'analyse et de conception, ou des classes de mise en œuvre et leurs relations, il expresse la structure statique des objets et des classes et le plan statique des programmes.

- **Diagramme de composants :** ce diagramme est utilisé pour montrer l'organisation et les relations structurelles entre les éléments d'un système.

- **Diagramme de structure composite** : utilisé pour montrer comment une chose est faite. Ce diagramme est particulièrement utile dans les structures complexes ou la conception basée sur les composants.

- **Diagramme de déploiement :** ce diagramme permet de montrer l'architecture d'exécution du système, les plateformes matérielles, les artefacts logiciels (éléments livrables ou exécutables) et les environnements logiciels (comme les systèmes d'exploitation et les machines virtuelles).

- **Diagramme d'objets :** utilisé pour montrer un exemple spécifique ou illustratif d'objets et de leurs liens. Souvent utilisé pour indiquer les conditions d'un évènement, comme un test ou un appel d'opération.

- **Diagramme de paquetages :** ce diagramme est utilisé pour organiser les éléments d'un modèle et montrer les dépendances entre eux.

- **Diagramme d'activités :** utilisé pour afficher le flux de données et/ou le flux de contrôle d'un comportement, capture le flux de travail entre les objets coopérants.

- **Diagramme de cas d'utilisation :** ce diagramme est utilisé pour montrer les services que les acteurs peuvent demander à un système.

- **Diagramme d'états-transitions :** permet de montrer le cycle de vie d'un objet particulier, ou les séquences qu'un objet traverse ou qu'une interface doit supporter.

- **Diagramme de séquence :** permet de se concentrer sur l'échange de messages entre un groupe d'objets et l'ordre des messages.

- **Diagramme de communication :** permet de se concentrer sur les messages entre un groupe d'objets et la relation sous-jacente des objets.

- **Diagramme de la vue globale d'interactions** : utilisé pour montrer de nombreux scénarios d'interaction.

- **Diagramme de temps :** utilisé pour montrer les changements et leur relation avec les temps d'horloge.

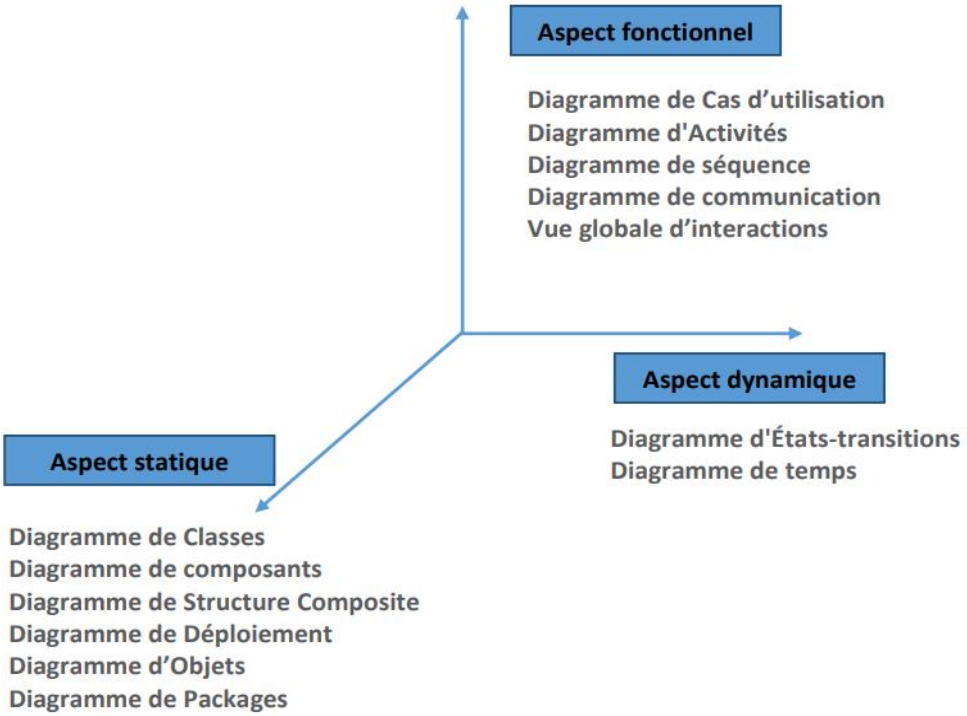

Figure 10 - Les trois aspects (axes) d'une modélisation. [9]

4.2. Diagrammes de conception de notre système

Pour décrire notre système nous avons utilisé trois diagrammes:

- Diagramme de cas d'utilisation
- Diagramme de classes
- Diagramme de séquence

4.2.1. Diagramme de cas d'utilisation

Le diagramme des cas d'utilisation illustre un exemple détaillé du monde réel, il permet de comprendre et de communiquer le but d'un système ou de ses composants. Ce diagramme détermine les utilisateurs du système (acteur), les services du système (cas d'utilisation), et l'objectif de ce système. Un acteur est un rôle qu'un utilisateur joue par rapport au système. Un cas d'utilisation est un but particulier qu'un utilisateur peut réellement utiliser le système pour l'accomplir. Les cas d'utilisation permettent de se concentrer sur les objectifs des utilisateurs et sur la production de systèmes pratiques [80, 81]. (Figure- 11) montres le diagramme de cas d'utilisation de notre système.

4.2.2. Le diagramme de classes

Montre la structure statique des classes dans le système et les différents types de relations qui existent entre ces classes. Il montre également les caractéristiques (les propriétés et les opérations) des classes et les contraintes qui s'appliquent à la façon dont les objets sont connectés [81]. Voir la figure 12.

4.2.3. Diagramme de séquence

Ce diagramme montre l'échange d'évènements entre les objets, il affiche les lignes de vie des objets participants pendant l'échange de messages dans un seul scénario. Un scénario est une séquence d'étapes décrivant une interaction entre un utilisateur et un système.

Une ligne de vie représente la vie en évolution de l'objet participant en montrant les évènements pertinents qui sont importants pour l'objet. Les diagrammes de séquence sont généralement les mieux adaptés pour explorer les scénarios ou les flux d'un cas d'utilisation particulier [80, 81]. Voir la figure 13.

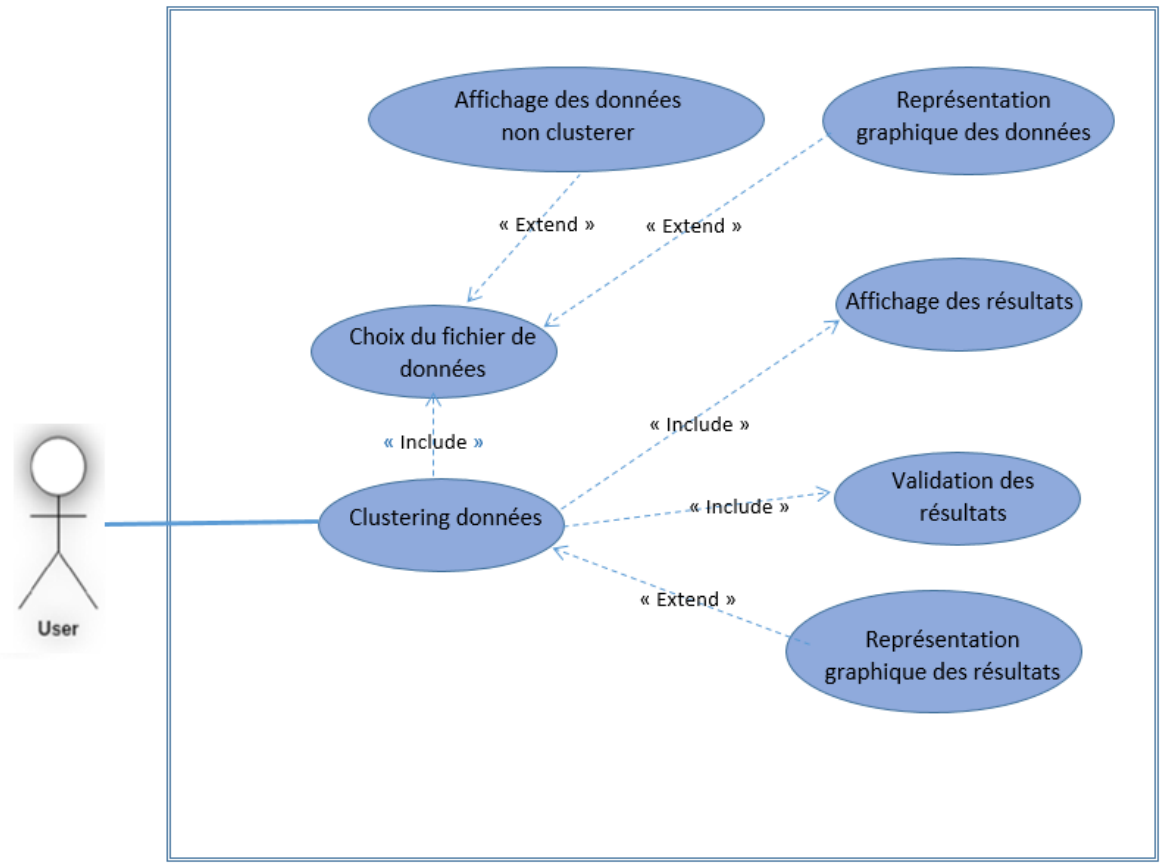

Figure 11- Diagramme cas d'utilisation

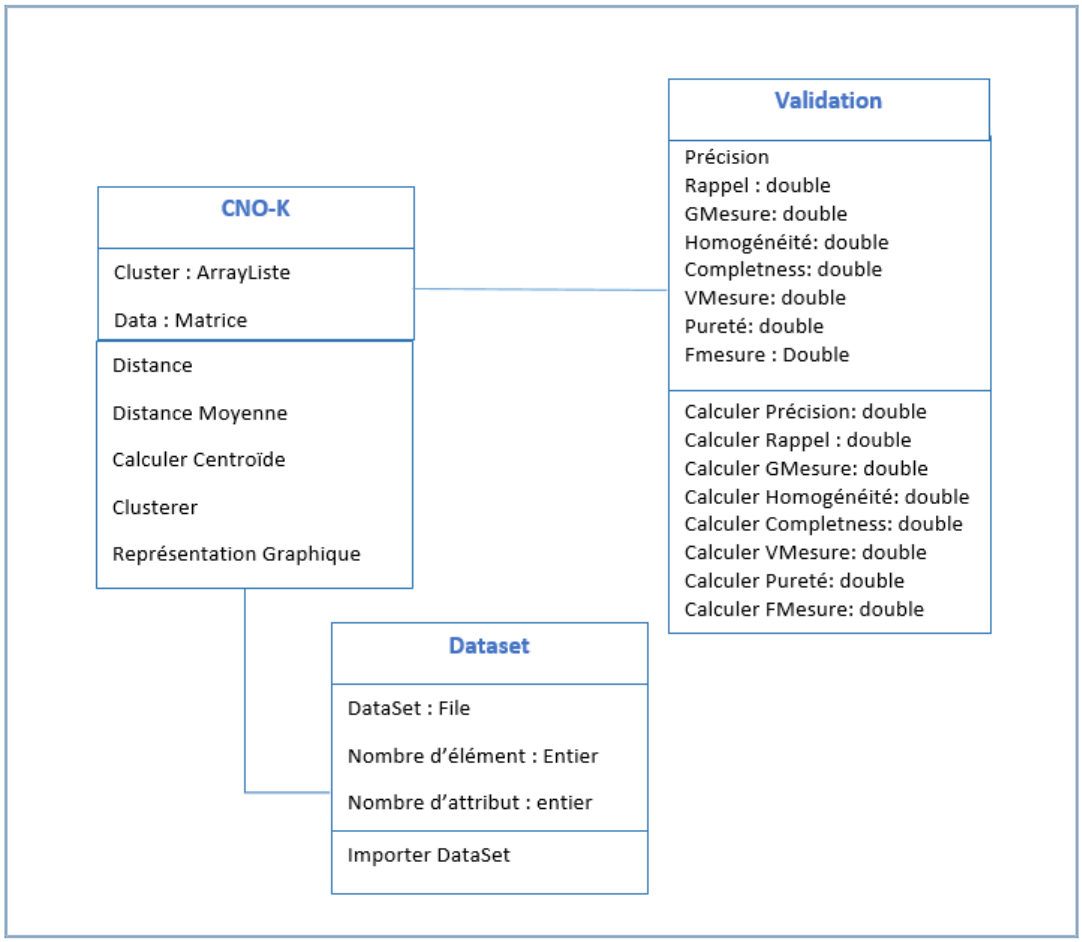

Figure 12 - Diagramme De classes

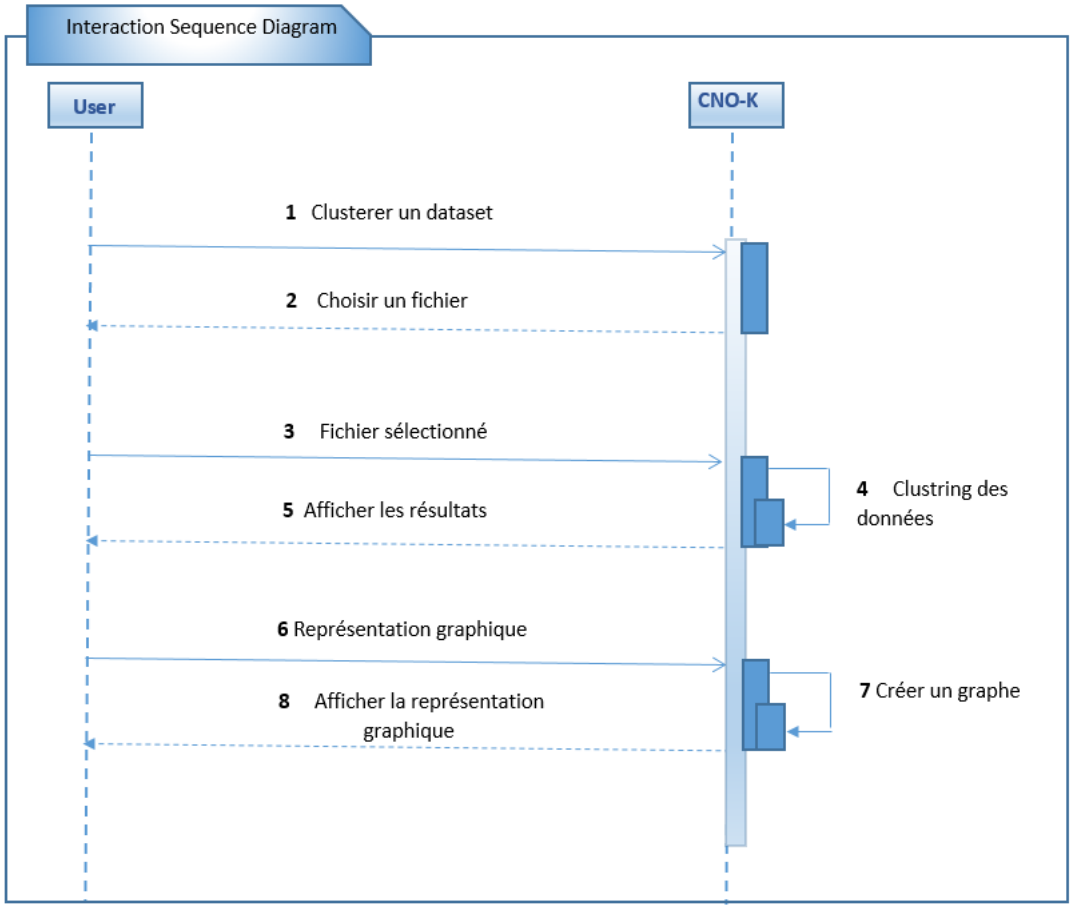

Figure 13- Diagramme De séquence

5. Conclusion

Dans ce chapitre nous avons présenté l'idée de base de notre approche du Clustering en utilisant une nouvelle méthode de Clustering sans précision préalable du nombre de cluster K, appelée CNo-K, ainsi que la modélisation de notre application en utilisant UML. Dans le chapitre suivant, nous allons présenter l'implémentation de notre application en précisant les outils utilisés dans notre travail.

Chapitre III

Implémentation

1. Introduction

Après avoir élaboré la conception de notre application, nous abordons dans ce chapitre la phase de réalisation. Cette phase dévoile la concrétisation de notre approche algorithme de Clustering sans k ; avec la présentation des outils de développement matériels et logiciels, et la présentation quelques captures d'application illustrant son fonctionnement.

2. Environnement de développement

Pour la réalisation de notre application, nous avons eu recours à plusieurs moyens matériels et logiciels.

2.1. Environnement matériel

L'application a été développée sous un environnement matériel caractérisé par :

- **Un processeur :** Intel(R) Celeron(R) CPU N3350 @ 1.10GHz 1.10 GHz.
- **Une RAM :** 4,00 Go (3,85 Go utilisable).

2.2. Environnement logiciel

Pour réaliser notre application, nous avons choisis l'environnement logiciel suivant

2.2.1. Système d'exploitation

L'application a été développée et exécutée sous un système d'exploitation Windows 10 Famille 64 bits.

2.2.2. Langage de développement Java

Java est un langage de programmation créé en 1995 par Patrick Naughton et James Goslin (employés de Sun Microsystems). L'historique de Java remonte à 1991, quand un groupe d'ingénieurs de Sun Microsystems, dirigé par Patrick Naughton et James Goslin, a voulu concevoir un petit langage informatique qui pourrait être utilisé pour les appareils des consommateurs comme les commutateurs de télévision par câble, le projet était appelé « Green ». Ce projet a continué son développement jusqu'à son apparition officielle sous le nom Java. En 2009 Sun Microsystems a été acheté par Oracle qui détient et maintient désormais Java [82].

Java est, largement, utilisé pour le développement d'applications d'entreprises et mobiles. Il prend en charge les tâches de programmations avancées telles que la programmation orienté objet, la programmation réseau, la connectivité des bases de données, et la programmation

parallèle et concurrente [82]. Les caractéristiques de Java telles que portabilité, simplicité, typage fort, gestion de la mémoire, et sûreté et sécurité ont contribuées à son énorme succès [83].

2.2.3. Environnement Eclipse

Est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

3. Présentation de l'application

L'approche proposée a été implémentée dans une application bureau Java qui est utilisée pour faire le Clustering de données.

3.1. Fenêtres de l'application

Toutes les fonctionnalités principales sont représentées par des boutons dans la fenêtre d'accueil, après avoir choisi et importer un dataset à partir du bouton «Importer dataset» on peut utiliser le bouton « Clusterer » pour le clustring comme on peut afficher une représentation graphique des données avant et après Clustering.

Figure 14- fenêtre principale

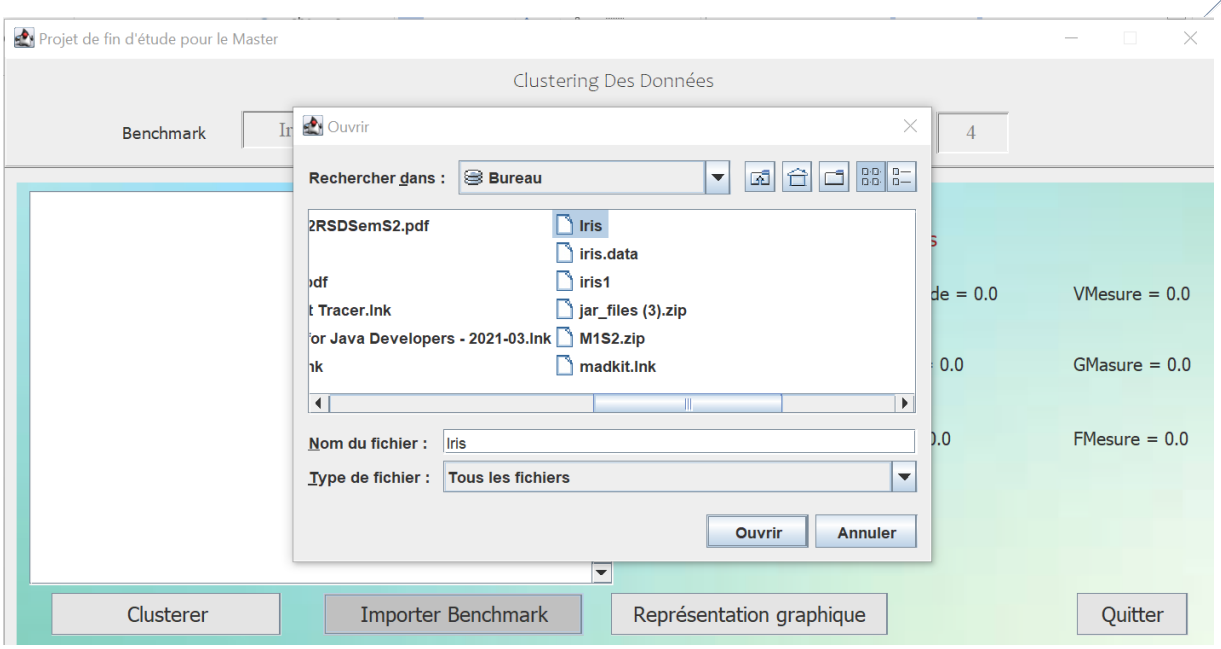

Figure 15- sélection fichier de données

```
7,0 3,2 4,7 1,4 Iris-versicolor
6,4 3,2 4,5 1,5 Iris-versicolor
6,9 3,1 4,9 1,5 Iris-versicolor
5,5 2,3 4,0 1,3 Iris-versicolor
6,3 3,4 5,6 2,4 Iris-virginica
6,4 3,1 5,5 1,8 Iris-virginica
6,5 2,8 4,6 1,5 Iris-versicolor
5,7 2,8 4,5 1,3 Iris-versicolor
5,1 3,5 1,4 0,2 Iris-setosa
4,9 3,0 1,4 0,2 Iris-setosa
4,7 3,2 1,3 0,2 Iris-setosa
4,6 3,1 1,5 0,2 Iris-setosa
5,0 3,6 1,4 0,2 Iris-setosa
5,5 2,4 3,8 1,1 Iris-versicolor
5,5 2,4 3,7 1,0 Iris-versicolor
5,8 2,7 3,9 1,2 Iris-versicolor
6,0 2,7 5,1 1,6 Iris-versicolor
5,4 3,9 1,7 0,4 Iris-setosa
```

```
5,1 3,5 1,4 0,2 Iris-setosa
4,9 3,0 1,4 0,2 Iris-setosa
4,7 3,2 1,3 0,2 Iris-setosa
4,6 3,1 1,5 0,2 Iris-setosa
5,0 3,6 1,4 0,2 Iris-setosa
5,4 3,9 1,7 0,4 Iris-setosa
4,6 3,4 1,4 0,3 Iris-setosa
5,0 3,4 1,5 0,2 Iris-setosa
4,4 2,9 1,4 0,2 Iris-setosa
4,9 3,1 1,5 0,1 Iris-setosa
5,4 3,7 1,5 0,2 Iris-setosa
4,8 3,4 1,6 0,2 Iris-setosa
4,8 3,0 1,4 0,1 Iris-setosa
4,3 3,0 1,1 0,1 Iris-setosa
5,8 4,0 1,2 0,2 Iris-setosa
5,7 4,4 1,5 0,4 Iris-setosa
5,4 3,9 1,3 0,4 Iris-setosa
5,1 3,5 1,4 0,3 Iris-setosa
```

Figure 17- Fichier de données

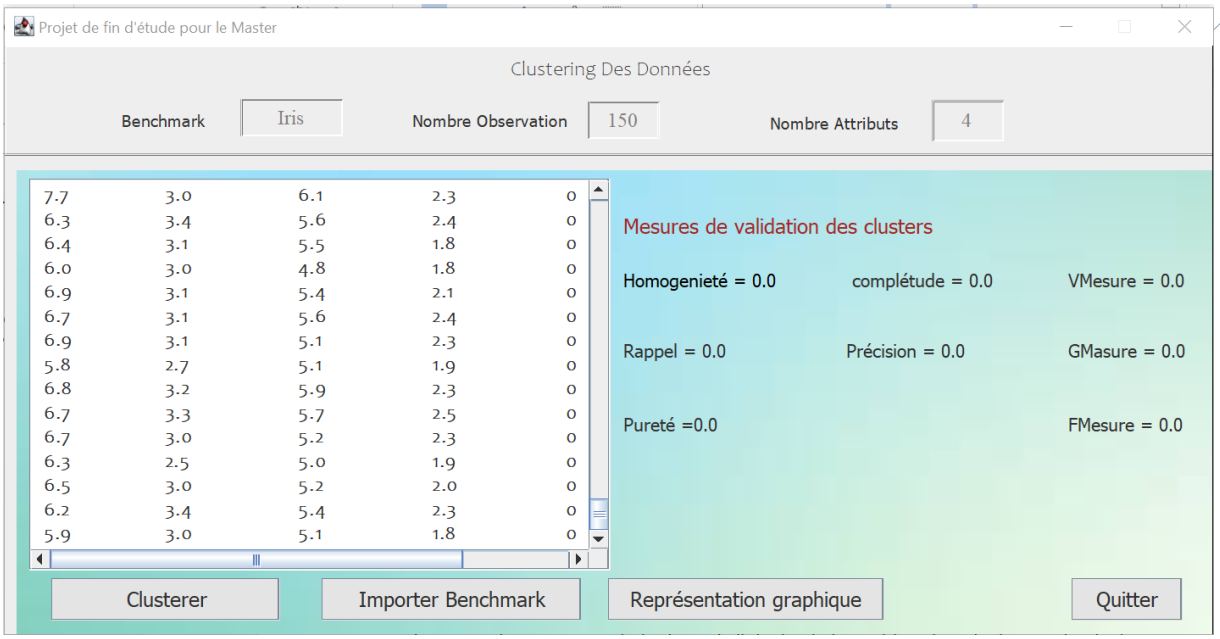

Figure 17- données avant Clustering

Après avoir choisi le fichier qui contient l'ensemble de données. Ils seront affichés sur une table ou chaque ligne représente un objet, et les attributs d'un objet sont séparés par des espaces. La dernière colonne représente la classe de l'objet initialement tous les objets appartient à la même classe.

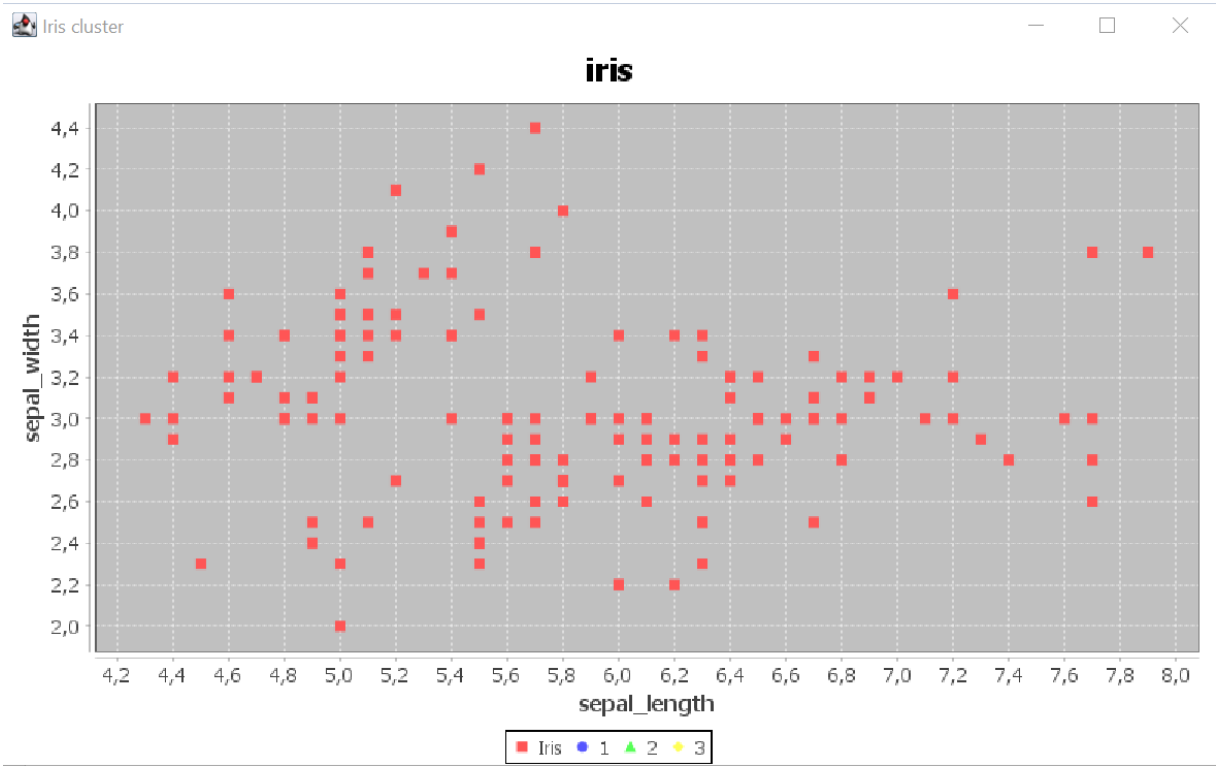

Figure 18- représentation des données avant Clustering

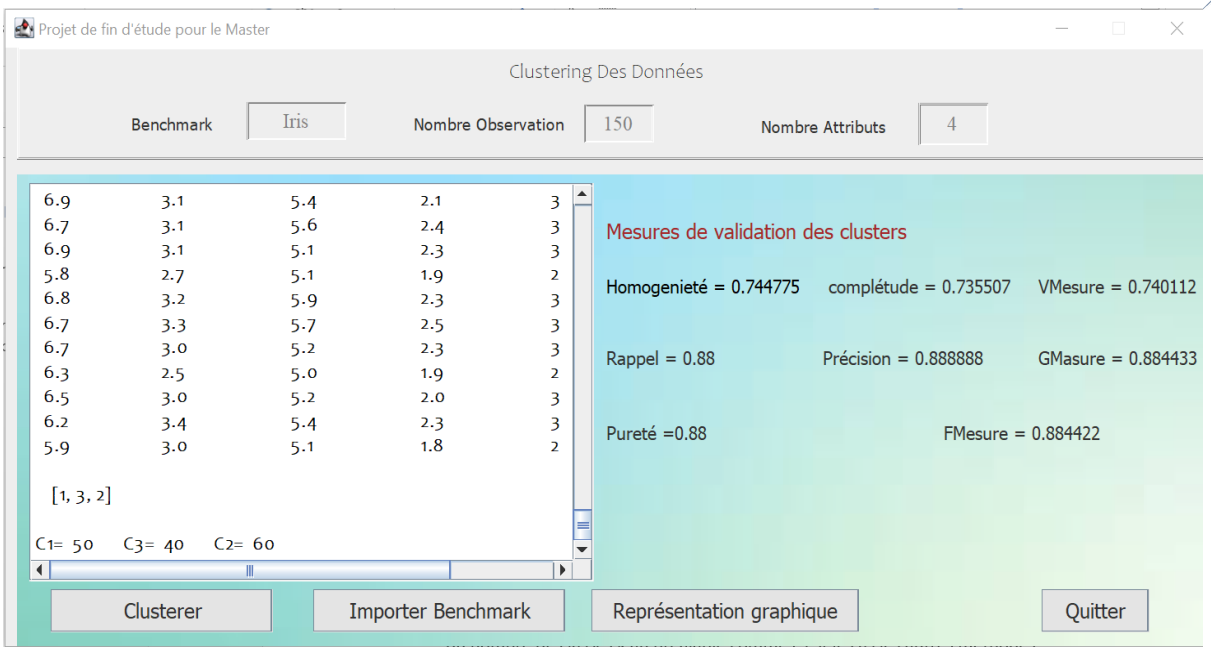

Figure 19- résultats de Clustering par l'algorithme CNO-K

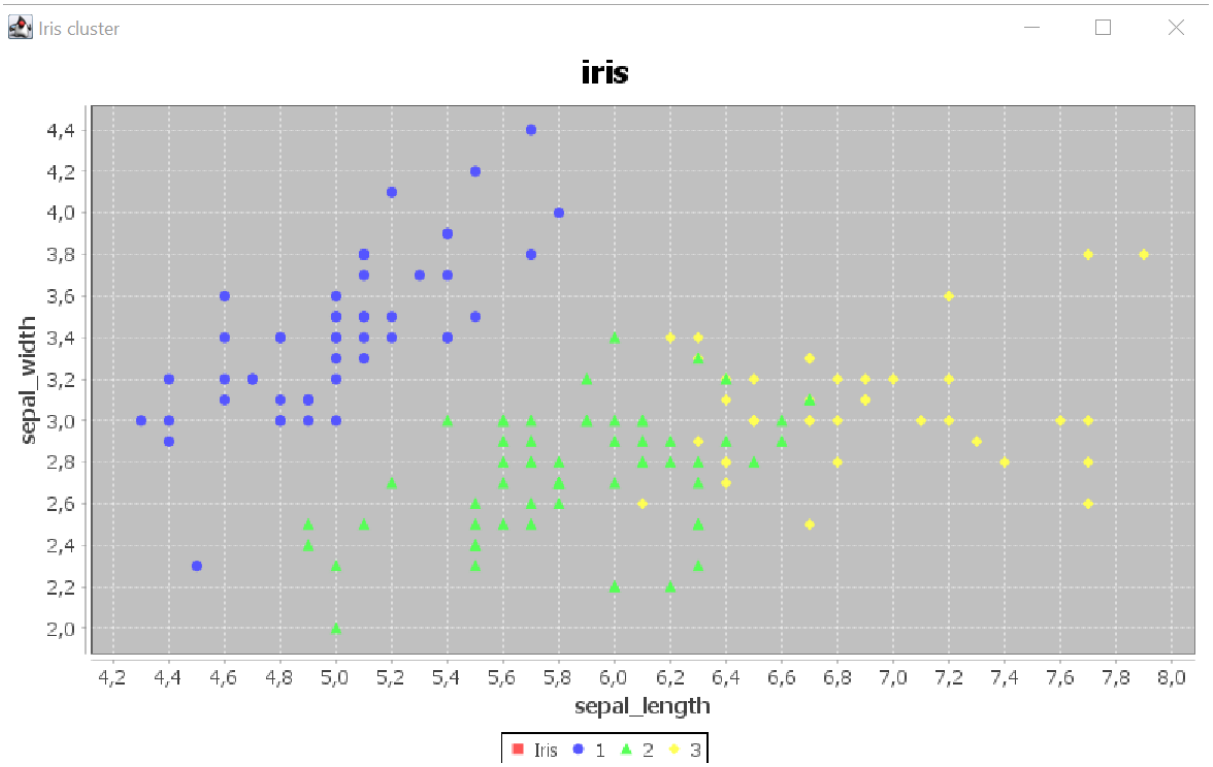

Figure 20- représentation des données après Clustering

Les résultats seront afficher sur deux parties de la fenêtre principale, la premier partie à droite contient la qualité des résultats calculées par huit indices : précision, rappel, G-Mesure, homogénéité, complétude, VMeasure, F-Measure, et la pureté. La deuxième partie gauche contient les résultats affichées sur une table, la dernière colonne de la table représente le numéro de cluster à qui l'objet appartient (avant le Clustering les cases de résultats sont initialisées par des zéros).

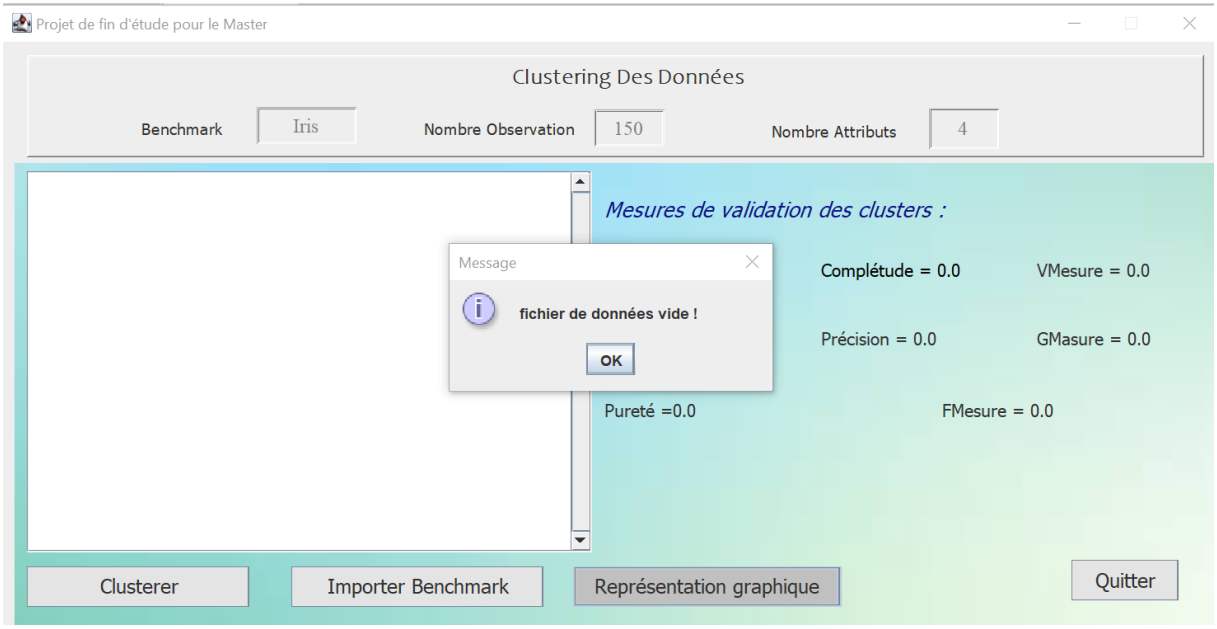

Figure 21- Contrôler et gérer les exceptions

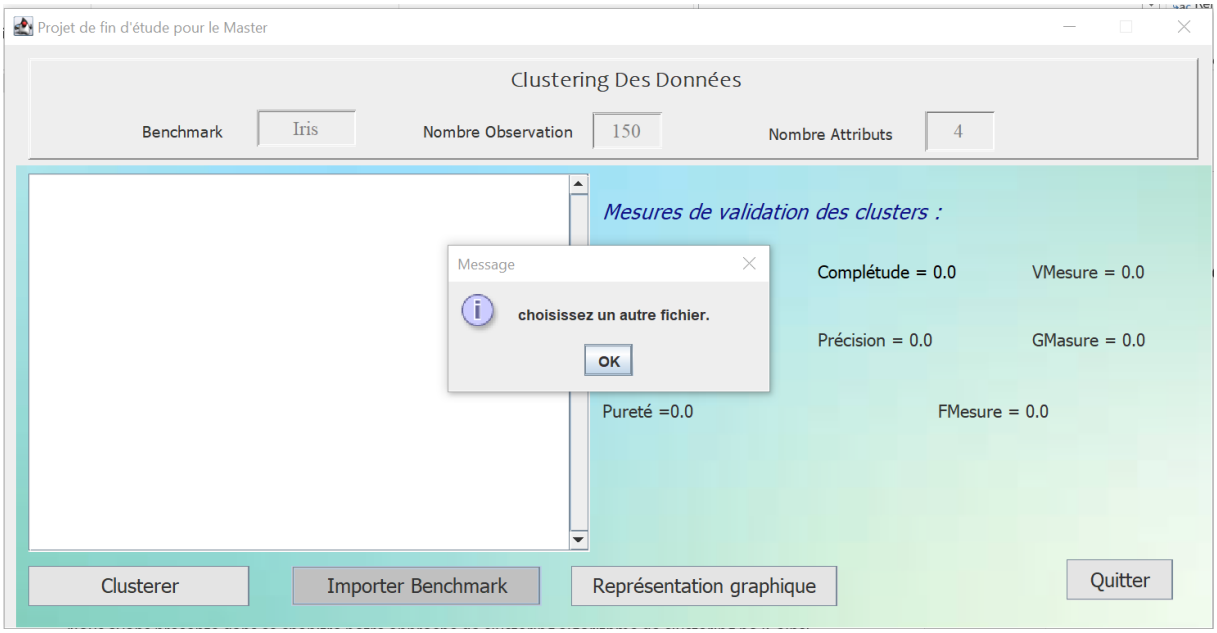

Figure 22- Contrôler le choix de fichier de données

4. Conclusion

Nous avons présenté dans ce chapitre notre approche de Clustering, l'algorithme de Clustering sans K CNo-K ainsi que l'implémentation de l'application pour cette approche, en montrant les outils de développement, avec la présentation détaillée des interfaces.

Dans le chapitre suivant, nous allons mettre en évidence la qualité de nos résultats à travers des comparaisons avec des résultats obtenus par différents métaheuristiques appliquées sur le jeu de données « Iris ».

Chapitre VI

Résultats expérimentaux et discussions

1. Introduction

Après avoir présenté notre approche de Clustering et Afin d'évaluer les performances de notre algorithme de Clustering sans k (CNo-K), appliqué au dataset Iris.

L'ensemble de données Iris a été utilisé dans l'article classique de RA Fisher de 1936, The Use of Multiple Measurements in Taxonomic Problems [84], et peut également être trouvé sur le UCI Machine Learning Repository .

IL contient quatre caractéristiques (longueur et largeur des sépales des fleurs et longueur et largeur de leurs pétales) de 50 échantillons de trois espèces d'Iris (Iris setosa, Iris virginica et Iris versicolor).

L'ensemble de données est souvent utilisé dans des exemples d'exploration de données, de classification et de regroupement et pour tester des algorithmes.

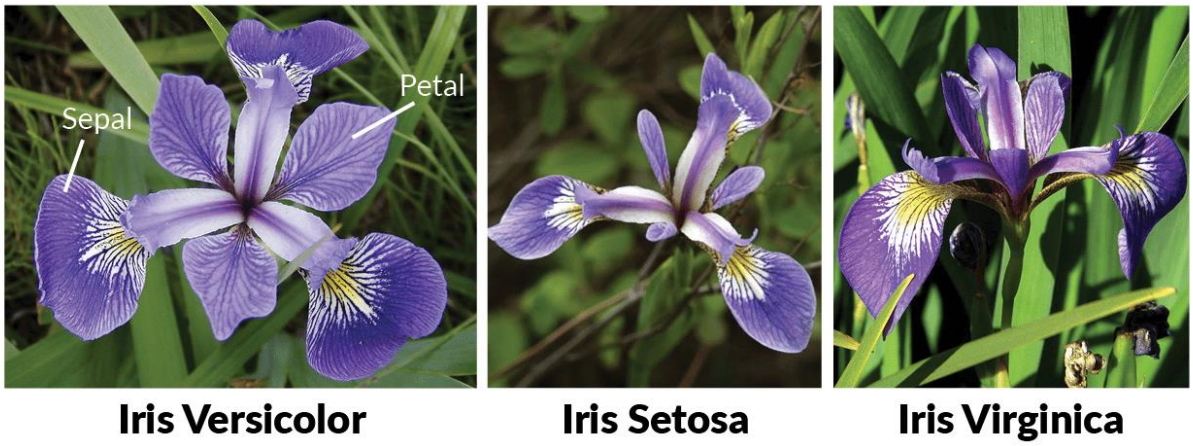

Figure 23 La fleur Iris

Benchmark	Nombre d'instances	Nombre d'attributs	Nombre de classes Iris
Iris	150	4	3

Table 4 – le benchmark utilisés

2. Comparaison de résultats

Notre approche va être comparée avec les méthodes présentées en [85, 86, 87] Deferential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Multi-verse Optimizer (MVO), et les résultats présentés dans [9] Optimisation par loups gris (GW1, GW2, GW3).

Pour les trois approches GWA1, GWA2 et GWA3 les valeurs des paramètres utilisées sont: la taille de la population égale à 60, le nombre maximal d'itérations égale à 50, le nombre des clusters choisi K= 3.

2.1. Comparaison avec MVO

La comparaison ici est faite selon quatre mesures : l'homogénéité, la complétude, VMeasure et la pureté. Ces mesures ont été présentées dans la section des indices externes dans le premier chapitre (section 6.1.6). Pour les trois approches GWA1, GWA2 et GWA3.

Jeu de données Iris	Homogénéité	Complétude	VMeasure	La pureté
DE	0.72778	0.75507	0.74096	0.86733
PSO	0.65750	0.82877	0.72629	0.77133
GA	0.60002	0.69056	0.64046	0.75333
MVO	0.73642	0.74749	0.74191	0.88667
GWA1	0.75961	0.79091	0.77456	0.88467
GWA2	0.74648	0.75420	0.75030	0.89533
GWA3	0.75165	0.76352	0.75754	0.89467
CNO-K	0,744775337	0,735508	0,740112628	0,88

Table 5 - Comparaison des résultats de Clustering avec MVO.

La (Table 5) représente les résultats d'homogénéité, complétude, VMesure de notre approche (CNO-K), Optimisation par loups gris (GW1, GW2, GW3 [9]) Diferential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), et Multi-verse Optimizer (MVO).

Dans cette table on peut constater que les résultats obtenus par notre approche sont considérables, alors que notre algorithme de Clustering sans k (CNo-K) ne nécessite pas la fixation du nombre de classes k au préalable comme c'est le cas des autres méthodes.

2.2. Comparaison avec GWAC

La comparaison ici est faite selon trois mesures : la précision, le rappel, et G-Mesure. Ces mesures ont été présentées dans la section des indices externes dans le premier chapitre (section 6.1.6). Les paramètres des méthodes comparées : K-means (KM), Genetic Algorithm based Clustering (GAC), Harmony search-based Clustering (HSC), Modified Harmony search-based Clustering (MHSC), Particle Swarm Optimization algorithm-based Clustering (PSOC), Flower Pollination algorithm-based Clustering (FPAC), Bat algorithm-based Clustering (BAC), et Grey Wolf algorithm-based Clustering (GWAC), elles sont définies dans le papier [85].

Les résultats sont collectés au cours de plus de 20 exécutions indépendantes.

Jeu de données Iris	précision	rappel	G-Mesure
KM	0.3018	0.3020	0.1144
GAC	0.3534	0.3733	0.2046
HSC	0.2428	0.2393	0.1102
MHSC	0.4586	0.4427	0.4082
PSOC	0.4299	0.4460	0.4364
FPAC	0.4266	0.4385	0.4319
BATC	0.4396	0.4360	0.4359
GWAC	0.5688	0.5813	0.5682
GWA1	0.7895	0.8669	0.8764
GWA2	0.8062	0.8188	0.8650
GWA3	0.8098	0.8364	0.8687
CNO-K	0,88888889	0,88	0,884433277

Table 6- Comparaison des résultats de Clustering avec GWAC.

La table ci-dessus nous montre que notre approche CNO-K a donné des meilleurs résultats pour la précision et le rappel par rapport aux autres algorithmes.

3. Conclusion

Notre nouvelle approche de Clustering «algorithme de Clustering sans précision du k ; CNO-K» ne nécessite pas la fixation de nombre de classe k au préalable comme c'est le cas des autres méthodes de Clustering, nous l'avons testé sur le jeu de données Iris. Et afin d'évaluer ses performances, les résultats obtenus ont été comparés avec ceux d'autres algorithmes.

Les résultats donnés par notre approche sont remarquables mêmes, meilleurs dans le cas des résultats rappel, précision et G-mesure comparés avec ces algorithmes.

Conclusion générale

Toute au long de notre travail nous avons présenté une nouvelle approche de Clustering sans K CNo-K qui ne nécessite pas la fixation du nombre de classes k au préalable ni aucun autre paramètre.

Nous avons commencé par présenter un état d'art sur le Clustering, suffisamment détaillé, y compris les mesures de proximités, les méthodes de Clustering et les indices de validités. Ensuite nous avons présenté notre nouvelle approche de Clustering, ainsi que la phase de conception et de réalisation de ce projet.

Enfin, nous avons comparé les résultats obtenus avec d'autres algorithmes réputés tel que : K-means (KM), Genetic Algorithm based Clustering (GAC), Harmony search-based Clustering (HSC), Modified Harmony search-based Clustering (MHSC), Particle Swarm Optimization algorithm-based Clustering (PSOC), Flower Pollination algorithm-based Clustering (FPAC), Bat algorithm-based Clustering (BAC), et Grey Wolf algorithm-based Clustering (GWAC), définis dans le papier [85].

Notre algorithme a enregistré sur le Benchmark Iris des performances considérables, voir meilleures à celles des autres méthodes, en offrant en plus la possibilité de trouver automatiquement le nombre de classes.

En perspective, une première amélioration notable, serait de proposer un meilleur critère d'acceptation ou de rejet d'un élément dans un cluster, plus précisément de décider si l'élément est assez proche d'un cluster existant, pour le rejoindre ou de créer lui-même, son propre, nouveau cluster, ce qui peut améliorer les performances de l'approche et élargir ses domaines d'application.

Dans notre approche, un élément est considéré plus proche d'un cluster, si la distance entre l'élément et le point centroïde du cluster le plus proche est minime c.-à-d inferieur à une distance moyenne de toutes les distances entre les N éléments à cluster. Pour le Benchmark Iris, cette mesure a donné des résultats remarquables.

Bibliographie

[1] B.Liu. Web Data Mining, Exploring Hyperlinks, Contents and Usage Data. Springer, Berlin, 2011.

[2] B. Mirkin, Clustering for Data Mining: A data Recovery Approach, National Research University Higher School of Economics, 2005.

[3] M. M. a. P. F. A.K. JAIN, Data Clustering: A Review, The Ohio State University, 1999. 93

[4] A. K. J. R C. Dubes, Algorithm for Clustering Data, 1988.

[5] S. C. Johnson, «Hierarchical Clustering schemes,» Psychometrika, p. 241–254, 1967.

[6] Hubert, «in and Max Hierarchical Clustering Using Asymmetric Similarity Measures,» Psychometrika, p. 63–72, 1973.

[7] J. C. Gower, «The Analysis of Asymmetry and Orthogonality,» North-Holland, Amsterdam, 1977.

[8] A. G. C. J. C. Gower, «Graphical Representation of Asymmetric Matrices,» Journal of the Royal Statistical Society, 197

[9] Mémoire de fin d'études en vue de l'obtention d'un Diplôme de Master En Informatique Spécialité : Réseaux et Systèmes Distribuée, Zebiri Ibrahim, 2020.

[10] M. R. Anderberg, Cluster Analysis for Applications, 1973.

[11] C. a. Stephenson, An Introduction to Numerical Classification, 1975.

[12] C. Romesburg, Cluster Analysis for Researchers, 1984.

[13] J. Gower, «A General Coefficient of Similarity and Some of its Properties,» Biometrics, pp. 857-871, 1971.

[14] C. M. J. W. Guojun Gan, Data Clustering Theory, Algorithms, and Applications, 2007.

[15] Wishart, «k-Means Clustering with Outlier Detection, Mixed Variables and Missing Values,» 2003.

[16] S. S. Pradeep Rai, «A survey of Clustering techniques,» International Journal of Computer Applications, 2010.

[17] M. K. J. P. J. Han, «Cluster Analysis: Basic concepts and Methods» chez Data Mining (Third Edition), Morgan Kaufmann, 2012, pp. 443 – 495

[18] P.Berkhin, «Survey of Clustering data mining techniques,» chez Grouping Multidimensional Data, pp. 25-71.

[19] M. A. A. R. B. Parul Agarwal, «Issues, Challeges and Tools of Clustering Algorithms,» International Journal of Computer Science Issues, 2011.

[20] M. V. K. Bomshad, «An Introduction to Cluster Analysis for Data Mining,» 2000.

[21] J. A. Hartigan, Clustering Algorithms, Wiley, 1975.

[22] J. Hämäläinen, «Improvement and Applications of the Elements of Prototype-Based Clustering,» Computer Science, 2018.

[23] C. C. Aggarwal, Data Clustering Algorithms and applications, 2013.

[24] S. K. Tan, «Cluster Analysis: Basic Concepts and Algorithms».

[25] A. K. Jain, «Data Clustering: 50 Years Beyond K-means».

[26] T. R. L., «Who Belongs In The Family,» 1953.

[27] H. STEINHAUS, «Sur la division des corps matériels en parties,» 1956.

[28] P. LLOYD, «Least squares Quantization in PCM,» 1982.

[29] A. C. Rencher, Methods of Multivariate Analysis, 2003.

[30] 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.

[31] A. Yahi ; Clustering des données de puces à ADN ; Thèse de master ; Université M. BOUDIAF ; M'SILA ; 2019.

[32] D. Stahl, Cluster Analysis, 2011.

[33] G. Paul, Cluster Analysis, 2000.

[34] Wishart, «Mode Analysis,» 1969.

[35] W. T. Williams, J. M. Lambert et G. N. Lance, «Multivariate Methods in Plant Ecology,» 1966.

[36] Sneath, «A Method for Curve Seeking from Scattered points,» 1966.

[37] B. S. Duran, Cluster analysis, 1970.

[38] T. SØRENSEN, «A Method of Establishing Groups of equal amplitude in plant sociology based on similarity of species Content and its application to analyses of the vegetation ON DANISH COMMONS,» 1948.

[39] MacNaughton, «Some Statistical and Other Numerical techniques for Classifying individuals,» 1965.

[40] D. C. WUNSCH et R. XU, Clustering, 2009.

[41] C. D. Michener et R. R. Sokal, A Statistical Method for Evaluating Systematic Relationships, 1958.

[42] W. T. W. G. N. Lance, A Generalized Sorting Strategy for Computer Classifications, 1966.

[43] McQuitty, Similarity Analysis by Reciprocal Pairs for Discrete and Continuous Data, 1966.

[44] Gower, A Comparison of Some Methods of Cluster Analysis, 1967.

[45] ward, Cluster analysis, 1963.

[46] A. D. Gordon, «Classification,» 1999.

[47] E. a. Cavalli-Sforza, «A Method for Cluster Analysis,» 1965.

[48] S. a. Symons, «On the Edwards and Cavalli-Sforza Method of Cluster Analysis,» 1971.

[49] Gordon, «Hierarchical Classification,» 1996. 95

[50] Brucker, «On The Complexity of Clustering Problems,» 1978.

[51] Welch, «Algorithmic Complexity: Three NP-hard Problems,» Computational Statistics, 1982.

[52] M. a. Roger, «Cluster Analysis,» 1984.

[53] McGarigal, «Multivariate Statistics for Wildlife and Ecology Research,» 2000.

[54] K. K. Steinbach, «A Comparison of Document Clustering Techniques,» 2000.

[55] K. D. Bailey, Cluster Analysis, Wiley, 1975.

[54] K. K. Steinbach, «A Comparison of Document Clustering Techniques,» 2000.

[56] G. a. J. Carmichael, «Finding Natural Clusters,» 1968.

[57] C. a. Sneath, «Taxometric Maps,» 1969.

[58] Parzen, «On Estimation of a Probability Density Function and Mode,» 1962.

[59] D. F. Specht, «Generation of Polynomial Discriminant Functions for Pattern Recognition,» 1967.

[60] P. a. Fischer, «A generalized k-Nearest Neighbor Rule,» 1970.

[61] H. S. Duda, «Pattern Classification, 2nd Edition,» 1973.

[62] W. a. Lane, «A kth Nearest Neighbor Clustering Procedure,» 1983.

[63] K. Fukunaga, «Introduction to Statistical Pattern Recognition, Second Edition,» 1972.

[64] M. Ester, P. Kriegel, J. Sander et X. XU, «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,» 1996.

[66] A. Hinneburg, «An Efficient Approach to Clustering in Large Multimedia Databases with Noise,» 1998.

[67] J. K. M. Han, Data Mining: Concepts and techniques. Morgan Kaufmann Publishers, 2001.

[68] M. Halkidi, Y. Batistakis et M. Vazirgiannis, Cluster Validity Methods: Part 1, 2002. 96

[69] A. M. Bagirov, S. Taheri et N. Karmitsa, Partitional Clustering via Nonsmooth Optimization, 2020.

[70] S. T. a. K. Koutroumbas, Pattern Recognition, 1999.

[71] B. Davies, A Cluster Separation Measure, 1979.

[72] Dunn, Well Separated Clusters and Optimal Fuzzy Partitions, 1974.

[73] Halkidi, Clustering Validity Checking Methods: Part 2, 2002.

[74] K. a. Rousseeuw, «Finding Groups in Data: An Introduction to Cluster Analysis,» 1990.

[75] P. J. ROUSSEEUW, «Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,» 1986.

[76] F. a. Mallows, A Method for Comparing Two Hierarchical Clusterings, 1983.

[77] S. W. a. Dorothea, Comparing Clusterings – An Overview, 2007.

[78] C. Ramdane, «Le Clustering de données : une nouvelle approche évolutionnaire quantique,» 2006.

[79] E. D. D. ·. J. J. Moré, «Benchmarking optimization software with performance profiles,» 2001.

[80] M. J. Chonoles, «UML 2 for dummies,» 2003.

[81] M. Fowler, «UML Distilled,» 1997.

[82] C. S. Horstmann, Core Java Volume I--Fundamentals, 2015.

[83] J. M. DOUDOUX, Développons en Java, 2005.

[84] BLAKE C., MERZ C., « UCI Repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences », 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[85] I. A. M. A. H. F. Mirjalili, «Multi-verse Optimizer: Theory, Literature Review, and Application in Data Clustering,» 2020.

[86] V. Kumar, J. K. Chhabra et D. Kumar, «Grey Wolf Algorithm-based Clustering Thechnique,» 2017.

[87] M. M. A. A. H. H. F. S. M. Ibrahim Aljarah, «Clustering analysis using a novel locality-informed grey wolf-inspired Clustering approach,» 2019.