F.S.D .....N° D'ordre .....

Université Saad DAHLAB de Blida

Faculté des Sciences

Département d'Informatique

Mémoire présenté par :

M^lle ATTALAH Elbatoul et M^r AISSOU Zakaria

En vue d'obtenir le diplôme de Master

Domaine: Mathématique et Informatique

Filière: Informatique

Spécialité : Ingénierie des logiciels

Sujet :

Visualisation des données multidimensionnelles à partir du clustering hiérarchique ascendant

Soutenue le: , devant le jury composé de :

M^me Bensttiti	Président
M^r Bala	Examinateur
M^lle Regueg	Examinateur
M^lle N.BENBLIDIA	Promotrice
M^lle K.AMEUR	Encadreur

Dédicaces

C'est avec un immense plaisir que je dédie ce travail

A mes très chers parents qui sont toute ma vie et tout ce que j'ai de plus cher au monde, en témoignage de ma reconnaissance infinie pour leur nombreux sacrifices.

Qu'ils trouvent en ce travail la preuve de mon éternel amour et ma reconnaissance envers eux.

Que dieu les garde et leur procure la santé et le bonheur.

Ainsi qu'à mon frère Oussama et mes sœurs Imène et Aya en témoignage de ma grande affection pour eux.

A mon cher ami Oussama et toute sa famille.

A mes chers grands parents, mes tantes, mes oncles et toute ma famille.

Aussi à mes amis Houssem,Ryadh, Meriem,Khadidja, Lyna, Hadjer, Hanane,Soumia et....

Sans oublier mon ami Zakaria qui a été mon acolyte dans cette épreuve et toute sa famille.

Mlle Attalah Elbatoul

Dédicaces

C'est avec un immense plaisir que je dédie ce travail

A mes très chers parents qui sont toute ma vie et tout ce que j'ai de plus cher au monde, en témoignage de ma reconnaissance infinie pour leur nombreux sacrifices.

Qu'ils trouvent en ce travail la preuve de mon éternel amour et ma reconnaissance envers eux.

Que dieu les garde et leur procure la santé et le bonheur.

Ainsi qu'à mes frères mes sœurs en témoignage de ma grande affection pour eux.

A mon cher ami Oussama et toute sa famille.

A mes chers grands parents, mes tantes, mes oncles et toute ma famille.

Aussi à mes amis Houssem,Ryadh

Sans oublier mon amie ElBatoul qui a été mon acolyte dans cette épreuve et toute sa famille.

Mr AISSOU Zakaria

Remerciement

Tout d'abord, nous tenons rendre grâce à dieu tout puissant pour nous avoir donné le courage et la détermination nécessaire pour finaliser ce travail et le mener à terme.

On ne saurait ne pas remercier encore une fois nos parents respectifs qui, par leur amour et leur affection nous ont permis d'arriver là où nous sommes aujourd'hui.

Nous remercions notre promotrice Mlle Benblidia pour son aide précieuse, ses conseils avisés et ses idées riches.

Nous tenons à remercier Mlle Ameur Khadidja qui a endossé son rôle d'encadreur de la meilleure façon qui soit. Nous retiendrons sa patience, sa disponibilité et sa compréhensibilité.

Nous tenons un grand et un spécial remerciement à Oussama pour sa sympathie, son aide et ses encouragements.

Nous remercions les membres de jury pour nous avoir fait l'honneur de juger notre travail.

Nous sommes reconnaissantes à Tous nos enseignants qui nous ont facilité la compréhension et la maitrise.

Nous tenons à qui nous a aidés de près ou de loin.

Merci.

Mlle Attalah Elbatoul et Mr Aissou Zakaria.

Sommaire

Chapitre II : Techniques de visualisation

Chapitre III : Clustering visuel de données

Chapitre IV : Application, tests et résultats

Conclusion générale

Résumé:

L'analyse de grands volumes de données multidimensionnelle utilisant les techniques d'analyse classique devient très importante et difficile .

Dans ce travail nous nous intéressons à la combinaison de la visualisation avec le clustering hiérarchique ascendant afin de visualiser précisément les résultats de clustering Nous enrichissons notre solution par des interactions proposées à l'utilisateur afin d'affiner son analyse, notre but sera de mettre l'utilisateur au centre du système pour qu'il interagisse avec lui de manière intuitive.

Notre travail consiste donc, à la conception et la réalisation d'un outil graphique pour visualiser et analyser les résultats du clustering hiérarchique ascendant de données multidimensionnelles via l'utilisation des technologies de visualisation d'informations.

Mots clés : Clustering hiérarchique ascendant, données multidimensionnelles, mesure de similarité (distance), technique de visualisation, treemap, exploration visuel des données.

Abstract

The analysis of large volumes of multidimensional data using the techniques of classical analysis becomes very important and difficult.

In this work we focus on the combination of visualization with the ascending hierarchical clustering to visualize precisely the results of clustering. We have expanded our solution proposed by the user to refine its analysis interactions; our goal is to put the user at the center of the system to interact with it intuitively.

Our work consists in the design and implementation of a graphical tool to visualize and analyze the results of ascending hierarchical clustering of multidimensional data through the use of information visualization technologies.

Keywords: ascending hierarchical clustering, multidimensional data, similarity measure (distance), visualization technique, treemap, visual data exploration.

ملخص

ان تحليل مجموعة البيانات ذات الإبعاد المتعددة باستعمال الطرق التقليدية اصبح صعبا او شبه مستحيل خصوصا مع تزايد هذه الاخيرة.

في هذا الصدد، نحن مهتمون باستعمال تقنيات العرض البياني متعددة الأبعاد مع الخوارزميات الهرمية المتصاعدة لمساعدة المستخدم أو المحلل في تحليل هذه النتائج واستخراج أكبر قدر من المعلومات عن طريق التفاعل مع العرض البياني وهذا بوضع المستخدم في مركز النظام التفاعل معها.

لذا فالغرض من دراستنا هو تصميم و إنجاز برنامج لتحليل نتائج خوارزميات الهرمية المتصاعدة باستعمال تقنيات العرض متعددة الأبعاد بمزيد من عمليات التفاعل.

كلمات البحث: الخوارزميات الهرمية المتصاعدة ، تقنيات العرض البياني متعددة الابعاد ، الاستكشاف البياني للمعلومات.

Listes des figures

Liste des tableaux

Introduction Générale

Sommaire

1. Contexte Général

L'homme est doté d'une capacité à visualiser l'information très développée qui joue un rôle majeur dans ses processus cognitifs (reconnaissance rapide de motifs, couleurs, formes et textures). Il utilise des méthodes graphiques afin de mieux appréhender des notions abstraites ou pour représenter le monde qui l'entoure.

Le développement rapide des outils informatiques permet au système informatique de stocker de très grandes quantités des données hétérogènes. L'analyse de ces données devient très importante et difficile. Des méthodes de classification et de visualisation efficaces sont des alternatives crées pour représenter ces immenses ensembles de données. Dans ce contexte on s'intéresse à la méthode de clustering hiérarchique ascendant (CHA) et la technique de visualisation « Treemap »pour visualiser les résultats de clustering hiérarchique ascendant. En effet, l'exploration visuelle des données a un fort potentiel d'applications, car elle facilite l'analyse, l'interprétation, la validation et aussi augmente l'aspect cognitif chez les analystes en interagissant sur les résultats.

2. Problématique

La capacité actuelle des ordinateurs permet d'enregistrer de très grandes quantités des données, à chaque milliers de second, l'analyse de ces grandes structures devient très importantes et difficile, le clustering était une des solutions les plus intéressantes, l'idée est de regrouper des enregistrements par des populations homogènes séduit. Mais la vitesse à laquelle l'informatique se développe fait que même les populations, autre fois distinctes, deviennent nombreuses et on revient au point de départ. Des techniques de visualisations de données multidimensionnelles existent dans le but de projeter visuellement les sources de données importantes, avec la possibilité d'interactions pour cibler l'analyse et proposer une meilleure exploration à l'utilisateur. Mais là aussi, les techniques de visualisation bien que révolutionnaires deviennent de plus en plus faibles face aux données encore plus importantes.

3. Objectifs

Donc dans ce contexte, notre objectif consiste à améliorer le processus d'analyses de résultats du Clustering hiérarchique des données multidimensionnelles par la conception et la réalisation d'un outil graphique pour visualiser et analyser ces derniers via l'utilisation d'une technique de visualisation hiérarchique qui s'appelle « Treemap ».

4. Organisation du mémoire

Afin d'atteindre l'objectif cité ci-dessus, notre mémoire s'articulera autour de quatre chapitres :

- Chapitre I : Clustering des données, ce chapitre débutera par les concepts de base, Nous présenterons les différentes mesures de distances existantes entre les différents types de données pris en charge par notre outil graphique. Nous nous orienterons à fur et à mesure vers le clustering (regroupement hiérarchique) nous décrirons l'algorithme choisi et toutes ces itérations, on finira ce chapitre par quelques mesures de validité de clustering.
- Chapitre II : Techniques de visualisation, dans lequel nous introduirons le domaine de la visualisation des données, le processus adopté, ses techniques et ses objectifs.
- Chapitre III : Clustering visuel de données, dans lequel nous définirons notre approche et le processus à réaliser, nous parlerons de processus de visualisation des résultats de clustering(Approche de couplage de clustering et visualisation),ce chapitre contient la démarche de modélisation utilisée pour concevoir notre outil graphique, nous présenterons la conception du système suivant cette démarche, Partant de l'analyse des besoins, ensuite l'analyse du système et la présentation des scenario jusqu'à la conception.
- Chapitre IV : Tests et implémentation, dans lequel nous présenterons l'environnement de développement (langages et outils utilisés), des captures d'écrans présenteront des interfaces de différentes étapes de l'exécution de notre application, ensuite nous évaluerons les expérimentations du notre système et la qualité du résultat qu'il fournit comparée à l'objectif initial à travers des expérimentations et des tests sur plusieurs benchmarks.

En fin, nous conclurons ce mémoire par une conclusion générale contenant quelques prescriptives à notre travail.

CHAPITRE 1

Clustering de données

I.1. Introduction

La classification automatique ou clustering est une technique importante dans le domaine de l'analyse de données. Appliquée dans de nombreux domaines scientifiques tels que l'imagerie, la biologie, le marketing, médecine etc., elle inclut des algorithmes et des méthodes pour regrouper ou classifier des objets, selon un critère de similarité. Elle se distingue de la classification supervisée par le fait qu'elle ne se base pas sur des classes prédéfinies. [1]

Le principe général du clustering, repose sur le regroupement des objets de telle manière que ceux appartiennent à la même classe soient fortement similaires entre eux et fortement dissimilaires avec les objets qui appartiennent aux autres classe. [2]

Dans ce chapitre nous allons essayer de donner une introduction sur le clustering .nous parcourrons quelques techniques du clustering avec ces concepts, A la fin, nous terminons par donner les différents types de mesures de validité de clusters.

I.2. Les concepts de base

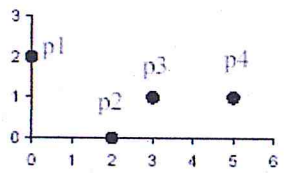
I.2.1.La matrice de données (Jeu de données) [3]

Les objets (échantillons, mesures, modèles, événements) sont habituellement représentés comme des points (vecteurs) dans un espace multidimensionnel, où chaque dimension représente un attribut distinct (variable, mesure) décrivant l'objet. Ainsi, un ensemble d'objets est représenté comme une matrice $m*n$, avec m lignes, une pour chaque objet et n colonnes, une pour chaque attribut. Cette matrice est appelée matrice de données ou jeu de données

I.2.2.La matrice de proximité [3]

Plusieurs algorithmes de clustering utilisent la matrice de données originale et beaucoup d'autres emploient une matrice de similarité, ou une matrice de dissimilarité. Pour la convenance, les deux matrices sont généralement mentionnées comme une matrice de proximité, P. Une matrice de proximité, P, est une matrice $m*m$ contenant toutes les dissimilarités ou les similarités entre les objets considérés. Si pi et pj sont le *ième* et le *jème* objets, respectivement, alors l'entrée à la *iemme* ligne et la *jemme* colonne de la matrice de proximité est la similarité, ou la dissimilarité, entre pi et pj.

La figure I.1 montre, respectivement, l'espace de données utilisé, leur matrice de données et leur matrice proximité correspondante. [4]



(a) Espace de données

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

(b) Matrice de données

	p1	p2	p3	p4
p1	0.000	2.828	3.162	5.099
p2	2.828	0.000	1.414	3.162
p3	3.162	1.414	0.000	2.000
p4	5.099	3.162	2.000	0.000

(c) Matrice de proximité

Figure I.1.L'espace de données, leur matrice de données et leur matrice de proximité.[4]

I.2.3.Définitions d'un Cluster

La définition de ce que constitue un cluster n'est pas bien définie et le terme, " cluster" n'a pas de définition précise [4] [5]. Cependant, plusieurs définitions d'un cluster sont généralement utilisées tel que :

I.2.3.a.Cluster bien séparé [4]

Un cluster est un ensemble de points tel que n'importe quel point dans le cluster est plus proche (ou plus similaire) de chaque autre point dans le cluster que de n'importe quel point qui n'est pas dans le cluster. Parfois un seuil est employé pour spécifier que tous les points dans un cluster doivent être suffisamment proches (ou similaires) l'un de l'autre.

I.2.3.b.Cluster basé sur le centre [5]

Un cluster est un ensemble de points tel qu'un point dans un cluster est plus proche (plus similaire) du " centre" de ce cluster, que du centre de n'importe quel autre cluster. Le centre d'un cluster est souvent un centroïde, la moyenne de tous les points dans le cluster, ou un médoïde, le point le plus représentatif d'un cluster.

I.2.3.c.Cluster contiguë (Le voisin le plus proche ou le clustering transitif) [4]

Un cluster est un ensemble de points tel qu'un point dans un cluster est plus proche (ou plus similaire) d'un ou de plusieurs autres points dans le cluster que de n'importe quel point qui n'est pas dans le cluster.

I.2.3.d.Définition basée sur la densité [5]

Un cluster est une région dense de points, qui est séparée des autres régions de haute densité par des régions de basse densité. Cette définition est souvent utilisée quand les clusters sont irréguliers ou entrelacés et quand les bruits sont présents.

I.2.4.Types et échelles de données [6]

La mesure de proximité et le type de clustering utilisé dépendent des types et échelles des attributs de données. Les trois types d'attributs sont montrés dans le tableau I.1, tandis que les différentes échelles de données sont montrées dans le tableau I.2.

Binaire	deux valeurs, vrai ou faux
Discret	un nombre fini de valeurs ou les entiers
Continu	un nombre infini de valeurs ou les réels

Tableau I.1. Les différents types d'attributs. [6]

Qualitative	Nominal	les valeurs sont juste des noms différents. par exemple : les codes postaux, les couleurs, le sexe.
	Ordinal	les valeurs reflètent un ordre, rien plus. par exemple : bon, meilleur, mieux ou couleurs ordonnées par le spectre.
Quantitative	Intervalle	la différence entre les valeurs est significative par exemple, l'intervalle de température.
	Ratio	rapport entre deux grandeurs. par exemple : les quantités monétaires, comme le salaire et le bénéfice et beaucoup de quantités physiques comme courant électrique, pression, etc.

Tableau I.2. Les différentes échelles de données. [6]

I.2.5. Distance et similarité [7]

Le concept de similarité ou de dissimilarité est le composant essentiel de n'importe quelle forme du clustering qui nous aide à naviguer dans l'espace de données pour former des clusters.

En calculant la similarité, nous pouvons sentir et articuler à quel point deux points sont proches, et sur la base de cette proximité, nous pouvons, les assignent au même cluster. Formellement, la similarité *d(x,y)* entre *x* et *y* est considéré comme une fonction à deux arguments satisfaisant les conditions suivantes :

$d(x, y) \geq 0$

$d(x, x) = 0$

$d(x, y) = d(y, x)$.

La distance est la mesure la plus utilisé parmi les types de mesures de similarité et de dissimilarité, elle exige la satisfaction de l'inégalité triangulaire c'est-à-dire, pour n'importe quel points *x*, *y* et *z*, nous avons : $d(x, y) + d(y, z) \geq d(x, z)$.

- **Distance entre les variables continues**

x_i et y_i sont les différentes valeurs qui représentent les deux points x et y qui disposent de n dimensions.Et (i=1, 2,3……n).

Le tableau suivant présente quelques fonctions de distance : [7]

Fonction de distance	fonction
Distance Euclidienne	$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Distancede Manhattan	$d(x,y) = \sum_{i=1}^{n}\|x_i - y_i\|$
Distance de chebyschev	$d(x, y) = max_{i=1,2.....n}\|x_i - y_i\|$
Distance de Minkowski	$d(x,y) = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p}$,p>0

Tableau I.3 : Fonctions de distance entre deux points *x* et *y*.

- **Distance entre les variables (attributs) binaires**

Comme nous l'avons décrit, un attribut de type binaire est représenté par deux valeurs qui peuvent être vrai/faux, 1/0, oui/non etc.[7]

La distance entre deux objets possédant des attributs binaires est calculée à l'aide de la table suivante :

	1	0
1	a	b
0	c	d

Tableau I.4. Matrice de contingence pour les données binaires. [7]

Exemple :
$O_i = (1, 1, 0, 1, 0)$ et $O_j = (1, 0, 0, 0, 1)$ ici nous avons a = 1, b = 2, c = 1, d = 1.

➢ Variables symétriques : [7]

$$d(i, j) = \frac{b + c}{a + b + c + d} \quad , d(O_i, O_j) = 3/5$$

Exemple : le sexe peut être codé comme suit :
Si deux personnes sont du même sexe on attribut 0 à la distance et 1 si elles sont de sexe différent.

➢ Variable asymétriques : [7]

$$d(i, j) = \frac{b + c}{a + b + c} \quad , d(O_i, O_j) = 3/4$$

Exemple : teste VIH, généralement on code par 1 la modalité la moins fréquente donc deux personnes ayant la valeur 1 pour le test sont plus similaires que deux personnes ayant un 0 pour le test.

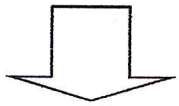
- **Méthode pour le calcul de similarité des valeurs qualitatives**

Les données catégoriques (nominales) où les variables ont plus de deux niveaux –couleur des yeux par exemple- pourraient être traitées de la même manière que les données binaires où chaque niveau de variable est considéré comme une variable binaire unique. [8]

Exemple :

Sujets	sexe	Couleur des yeux	test	permission	Exam 1	Exam 2
1	M	marron	P	N	O	O
2	M	marron	P	N	N	O
3	F	vert	N	O	N	N

Tableau I.5. Matrice de données initiales [8]



sujets	Sexe	marron	vert	test	permission	Exam 1	Exam 2
1	M	1	0	P	N	O	O
2	M	1	0	P	N	N	O
3	F	0	1	N	O	N	N

Tableau I.6. Matrice précédente traitée et préparée pour le calcul de distances [8]

Le sexe et la couleur des yeux sont des critères symétriques.

Le test et les exams 1 et 2 sont asymétriques et on précise que : P et O = 1 et N = 0.

Nous nous intéressons au calcul des distances des valeurs asymétriques.

$$d(1,2) = \frac{0+1}{2+0+1} = \frac{1}{3} = 0.33$$

$$d(1,3) = \frac{1+3}{0+1+3} = \frac{4}{4} = 1$$

$$d(2,3) = \frac{1+2}{0+1+2} = \frac{3}{3} = 1$$

On déduit que les sujets 1 et 2 sont plus proches entre eux que le binôme de sujets 1 et 3 ou 2 et 3. La distance entre les sujets 1 et 3, et 2 et 3 est la même.

I.3. Techniques principales de clustering

Depuis de nombreuses années, beaucoup de différentes techniques de clustering ont été proposées.

L'objectif de ces méthodes est de regrouper les individus en un nombre restreint de classes Homogènes. Dans ce type de méthodes les classes seront obtenues à l'aide des algorithmes Formalisés. [6]

On distingue quatre méthodes de classification non-supervisée (clustering) : [7]

- clustering par partitionnement (non hiérarchiques).
- clustering hiérarchiques.
- Clustering basé sur la densité.

- Clustering basé sur les grilles.

Pour chacune de ces méthodes, il y a beaucoup de sous-types et beaucoup d'algorithmes différents pour trouver des clusters.

Dans notre travail nous nous intéressons au clustering hiérarchique.

I.3.1. Classification hiérarchiques

Le clustering hiérarchique construit une hiérarchie de clusters, ou, en d'autres termes, un arbre de clusters, également connu sous le nom de dendrogramme [8]. Ce dendrogramme décrit l'ordre dans lequel les points sont fusionnés (vue de bas en haut) ou les clusters sont fractionnés (vue de haut en bas). La figure I.2 représente sept points *p1*, *p2*, *p3*, *p4*, *p5*, *p6* et *p7* dans trois clusters et le dendrogramme correspondant. Le dendrogramme peut être coupé à différents niveaux pour donner les différents clusters des données [9] :

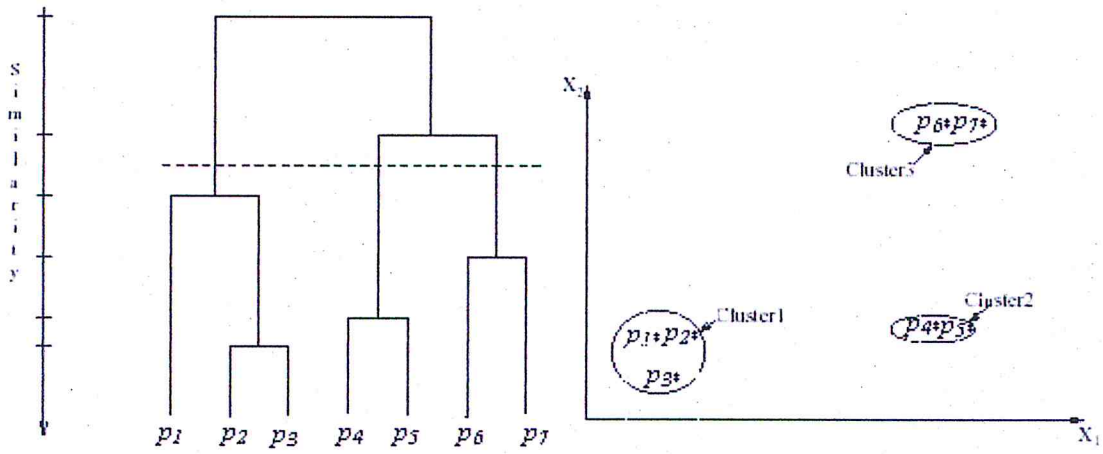


Figure I.2 : Clustering de sept points et le dendrogramme correspondant. [9]

Il existe deux approches de classification hiérarchique : Classification ascendante (agglomérative) et Classification descendante (divisive).illustrer dans la figure.I.3.

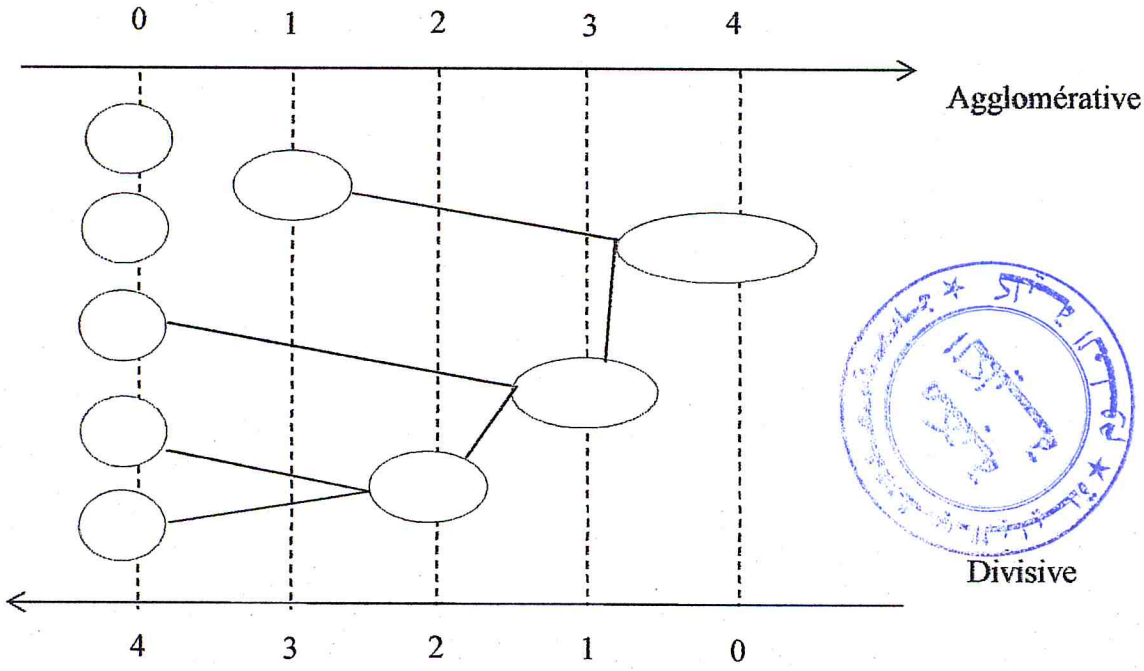


Figure I.3: Dendrogramme [10]

I.3.1.a .Classification hiérarchique descendante (top-down) [11]

Commence par un cluster de tous les points de données et fractionne récursivement le cluster le plus approprié et qui procède par dichotomies successives de l'ensemble des individus tout entier, jusqu'à ce que tous les clusters aient une taille 1.

L'algorithme divisive le plus simple et le plus utilisé est donné par les étapes suivantes [12] :

1. Calculer l'arbre de couverture minimum (MST ou Minimum Spanning Tree) du graphe de proximité.

2. Créer un nouveau cluster en enlevant le lien correspondant à la plus grande distance.

3. Répéter l'étape 2 jusqu'à ce que seulement les clusters de singleton restent.

Dans notre cadre d'étude on s'intéresse par la classification hiérarchique ascendante que nous allons détaillées dans la section suivante.

I.3.1.b. Classification hiérarchique ascendante (CHA)

Commence par des clusters d'un seul point (singleton) et fusionne récursivement deux ou plusieurs clusters les plus appropriées jusqu'à obtenir un seul cluster. [11]

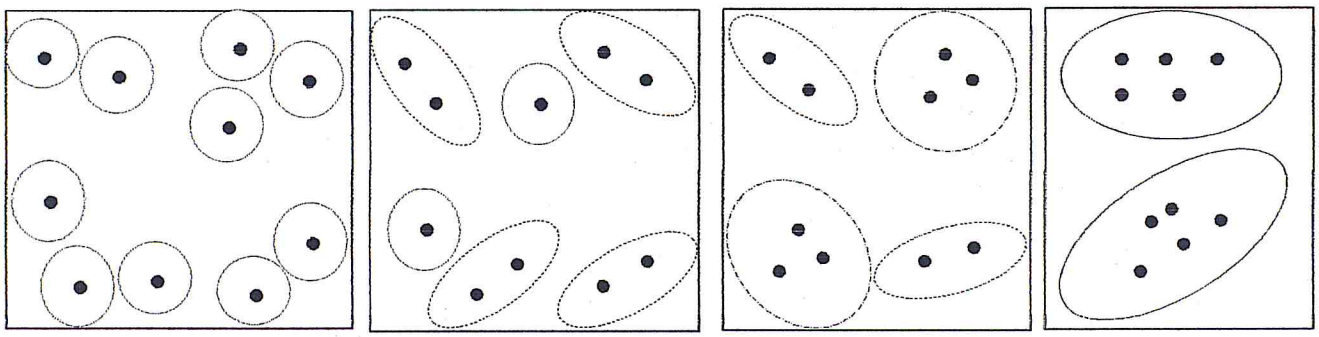


Figure I.4: Groupement agglomeratif [9]

- **Algorithme**

Plusieurs techniques hiérarchiques ascendantes peuvent être exprimées par l'algorithme suivant connu sous le nom de l'algorithme de Lance-Williams [11]:

1. Chaque élément est un cluster.
2. Calculer la matrice de proximité.
3. Fusionner les deux clusters les plus proches (les plus similaires).
4. Mettre à jour la matrice de proximité pour refléter la proximité entre le nouveau cluster et les clusters originaux.
5. Répéter les étapes 3 et 4 jusqu'à ce que seulement un seul cluster reste.

Un algorithme *agglomératif* fonctionne donc en recherchant à chaque étape les classes les plus proches pour les fusionner, et l'étape la plus importante dans l'algorithme réside dans le choix d'une formule pour le recalcule des distances après fusion.

I.3.1.b.1.Stratégies d'agrégation sur dissimilarités [7]

Le problème est de définir la dissimilarité entre la réunion de deux éléments et un troisième. On distingue trois méthodes :

- ✓ **Le saut minimum (« Simple link »)**

Cette méthode (connue sous le nom de « single linkage » en anglais »)

La proximité de deux clusters est définie par la distance minimale entre n'importe quels deux points dans les clusters différents. Cette distance minimale entre les points appartenant aux clusters A et B est calculée avec la formule :

$$d(A, B) = min_{x \in A, y \in B}\ d(x, y)$$

✓ **Le diamètre (« complete link »)**

La proximité de deux clusters est définie par la distance maximale entre n'importe quels deux points dans les clusters différents. Cette distance maximale entre les points appartenant aux clusters *A* et *B* est calculée avec la formule :

$$d(A, B) = max_{x \in A, y \in B}\ d(x, y)$$

✓ **Distance moyenne (« Average link »)**

Pour la version d'Average link du clustering hiérarchique, la distance de deux clusters est définie par la moyenne des distances entre toutes les paires de points dans les clusters différents. C'est une approche intermédiaire entre Single link et Complete link .Ceci est exprimé par l'équation suivante:

$d\ (A\ ,B) = \frac{\sum_{x \in A, y \in B} d(x,y)}{n_A\ n_B}$, *tel que* n_A, n_B *sont les nombre d'element de A,B respectivement*

I.3.1.b.2.Les avantages et les inconvénients [13]

L'avantage du clustering hiérarchique est sa stabilité. Ceci est dû à deux raisons particulières, premièrement, l'initialisation des classes est toujours la même et deuxièmement, pour une itération quelconque, les algorithmes considèrent seulement les classes précédemment obtenues ; de cette manière, un objet appartenant à une classe ne peut pas se retrouver dans une autre classe dans les itérations suivantes. Ceci peut être vu comme un avantage mais aussi comme un inconvénient car la flexibilité de la méthode diminue. Leur principal inconvénient est lié à la taille de l'ensemble de données. A chaque itération, ces méthodes utilisent la matrice de distance inter point ou interclasse. Ceci fait que pour les applications contenant des bases de données très grandes ces méthodes ne sont que rarement utilisées.

I.3.2.Classification par partitionnement (non hiérarchiques)

Il permet de définir une partition composée d'un nombre maximum de classes défini à l'avance. Cette méthode consiste alors à choisir d'abord un initiateur pour chaque groupe. Chaque élément est ensuite rattaché à l'initiateur le plus proche. De ce processus résulte un ensemble de groupes disjoints. Le centre de gravité de chaque groupe est alors calculé. Le processus est réitéré comme initiateurs de groupes les centres de gravité obtenus à l'itération précédente ; cela jusqu'à ce que les centres de gravité obtenus soient stables.

I.3.2.a. Les méthodes de clustering non hiérarchique

Ce sont des méthodes qui produisent directement une partition en un nombre fixé de classes. Parmi ces méthodes, nous retrouvons :

I.3.2.a.1.Méthode de leader [14]

Lorsque le premier objet arrive, on lui attribue la première classe et il devient le leader de celle-ci. Ensuite, chaque fois qu'un nouvel objet se présente, on calcule sa distance par rapport aux leaders de chacune des classes existantes à cet instant, et on compare cette distance à un seuil. Si cette distance est inférieure au seuil fixé, on attribue au nouvel objet la classe du premier leader trouvé (pour lequel la distance calculée est inférieure au seuil), sinon une nouvelle classe est créée et le nouvel objet devient le leader de cette Classe.

I.3.2.a.2.Méthode de *k*-means [9]

Cette méthode est encore appelée algorithme des centres mobiles [9] .Ce type D'algorithme, où la classe est représentée par son centre de gravité.

L'algorithme *k*-means est l'un des algorithmes de clustering les plus connus. Il est basé sur la méthode des centroïdes (ou centres de gravité). Le principe de cette méthode est le suivant :

On se donne pour commencer, *k* centres arbitraires *c1*, *c2*,..., *ck* où chaque *ci* représente le centre D'une classe c_i. Chaque classe c_i est représentée par un ensemble d'individus plus proches de *ci* que de tout autre centre. Après cette initialisation, on effectue une deuxième partition en regroupant les individus autour des m_j qui prennent alors la place des c_j (m_j est le centre de gravité de la classe c_j, calculé en utilisant les nouvelles classes obtenues). Le processus est ainsi réitéré jusqu'à atteindre un état de stabilité où aucune amélioration n'est possible.

Cette méthode est convergente et surtout avantageuse du point de vue calcul mais elle dépend Essentiellement de la partition initiale. Il existe donc un risque d'obtenir une partition qui ne soit pas optimale mais seulement meilleure que la partition initiale.

I.3.2.a.3.Méthode des nuées dynamiques [15]

Cette méthode a été proposée par [15]. Elle peut être considérée comme une Généralisation de la méthode des centres mobiles. Le principe de la méthode est le suivant : on tire au hasard *k* noyaux parmi une famille de noyaux (chaque noyau contient un sous-ensemble d'individus).Puis chaque point de l'ensemble d'apprentissage est affecté au noyau dont il est plus proche. On obtient ainsi une partition en *k* classes dont on calcule les noyaux. On recommence le processus avec les nouveaux noyaux et ainsi de suite jusqu'à

ce que la qualité de la partition ne s'améliore plus. Cette méthode a l'avantage de traiter rapidement de grands ensembles d'individus. Elle fournit une solution dépendant de la configuration initiale et nécessite le choix du nombre de classes. En général le nombre de classes est fixé par l'utilisateur et l'initialisation est faite par un tirage au hasard.

I.3.2.b.Les avantages et les inconvénients [15]

Les méthodes non hiérarchiques permettent de traiter rapidement de grands ensembles d'individus. La complexité de l'algorithme est seulement de n log(n) ou n est le nombre de données, son principal inconvénient est le nombre de classe est fixé au départ. Si le nombre de classes n'est pas connu ou si ce nombre ne correspond pas à la configuration véritable de l'ensemble d'individus (d'où le risque d'obtenir des partitions de valeurs douteuses), il faut presque toujours tester diverses valeurs de *k*, ce qui augmente le temps de calcul. C'est pourquoi, lorsque le nombre des individus n'est pas trop élevé, on préfère utiliser les méthodes hiérarchiques.

I.3.3.Clustering basé sur la densité

Les algorithmes basés sur la densité considèrent les clusters comme des régions de points denses dans l'espace de données qui sont séparées par des régions de faible densité [16]. Un des algorithmes les plus bien connus de cette catégorie est le DBSCAN [17]. L'idée principale de DBSCAN est celle pour chaque point dans un cluster, le voisinage d'un rayon donné *Eps* doit contenir au moins un nombre minimum de point *MinPts*, c-à-d la cardinalité du voisinage doit excéder un seuil. DBSCAN est basé sur les concepts de point noyau, point bordure et point bruit, qui sont aussi basés sur des notions d'accessibilité et de connectivité.

Un point noyau est un point avec un voisinage consistant de plus de *MinPts* points. Le concept d'un point *y* qui densité-accessible d'un point noyau *x* est défini comme une séquence finie entre *x* et *y* tel que chaque point successeur appartient au voisinage de son prédécesseur. Le concept de densité-connectivité est défini tel que deux points *x, y* sont dits densité connectés s'ils sont densité-accessibles d'un même point noyau.

L'algorithme de DBSCAN est comme suit:[17]

1. Choisir un point arbitraire *x*.
2. Trouver tous les points qui sont densité-accessibles de *x resp.* ***Eps*** *et* ***MinPts***. Si *x* est un point noyau alors nous avons formé un cluster. Si *x* est un point bordure alors aucun point n'est densité accessible de *x*.

3. Répétition pour le prochain point dans les données.

I.3.4.Clustering basé sur les grilles

Ces algorithmes commencent par partitionner l'espace de données à un nombre fini de cellules et accomplir ensuite des opérations exigées sur l'espace partitionné. Les cellules qui contiennent plus qu'un certain nombre de points sont traitées comme denses et les cellules denses sont connectées pour former les clusters [18]. La forme de base d'un algorithme basé sur les grilles est la suivante [19]:

1. Diviser l'espace sur lequel les données s'étendent en cellules rectangulaires, par exemple, on divise l'intervalle de valeurs de chaque dimension en cellules de taille égales.
2. Rejeter les cellules de grille à densité basse. Ceci assume une définition de cluster basée sur la densité, c'est-à-dire, que les régions à haute densité représentent des clusters, alors que les régions à basse densité représentent le bruit. Cette supposition est souvent bonne, bien que les approches basées sur la densité puissent avoir des difficultés quand il y a des clusters de différentes densités.
3. Combiner des cellules adjacentes à haute densité pour former des clusters.

I.4.Techniques de validation de clustering

L'objectif principal de la validation de clusters est d'évaluer le résultat de clustering afin de trouver le meilleur partitionnement du jeu de données. Il existe des approches de validité de cluster pour évaluer quantitativement le résultat d'un algorithme de clustering [20]. Il existe deux critères qui ont été largement considérés suffisants pour mesurer la qualité du partitionnement de données, tel que:[26]

- La compacité, les membres de chaque cluster doivent être proche l'un de l'autre autant que possible. Une mesure commune de compacité est la variance, qu'on doit minimiser.
- La séparation, les clusters eux-mêmes doivent être largement espacée. La distance euclidienne entre les centroïdes de clusters donne une indication de la séparation de clusters.

I.4.1.Mesures externes

Le premier groupe de fonctions d'évaluation analytiques disponibles pour l'analyse de clusters est le groupe de fonctions conçues pour des problèmes de références pour lesquels le bon nombre de cluster et la classification correcte pour chaque point de données sont connus.

L'évaluation devient beaucoup plus juste et sérieuse dans ces cas, puisque les propriétés désirées du partitionnement (qui conforme avec un certain degré à la définition de problème) peuvent être négligées, et nous pouvons seulement se concentrer sur la validité des assignements aux clusters obtenus [21].

Les mesures externes que nous allons présenter, appliquent directement la connaissance des étiquettes de classes. Elles évaluent les clusters générés en prenant en compte les classes d'appartenances correctes.

- **La F-mesure**

La F-mesure est une fonction utilisée souvent dans la littérature pour évaluer les algorithmes de clustering.

La F-mesure adopte les idées de la précision et du rappel de la recherche documentaire [22]. Elle compare la qualité de clustering en tenant compte des classes correctes connues pour un jeu de données. Soit $C = (C1, C2, \ldots, Ck)$ un clustering donné et $R = (R1, R2, \ldots, Rk')$ les classes correctes.[22]

Chaque classe Ri contient Ni points de données, chaque cluster Cj (généré par l'algorithme) est considéré comme l'ensemble de Nj points de données. Nij donne le nombre de points de la classe Ri dans le cluster Cj et N donne le nombre total des points du jeu de données. Pour chaque classe Ri et un cluster Cj, la précision et le rappel sont alors défini comme :

$$\text{Prec}=(R_i, C_j)=\frac{N_{ij}}{N_j} \quad \text{et} \quad \text{Rep}=(R_i, C_j) = \frac{N_{ij}}{N_i}$$

Et la valeur de F-mesure correspondante est :

$$Fmes=(R_i, C_j) = \frac{(b^2+1).\text{Prec}(R_i,C_j).\text{Rep}(R_i,C_j)}{b^2.\text{Prec}(R_i,C_j)+\text{Rep}(R_i,C_j)}$$

Où des coefficients égaux de Prec (Ri,Cj) et Rep (Ri,Cj) sont obtenu si b=1. La valeur globale de F-mesure F pour toute la partition est calculée comme

$$\text{F(C)}= \sum_{i=1}^{K'} \frac{N_i}{N} max_{c_i \in C}(Fmes(R_i,C_j))$$

Elle est limitée à l'intervalle [0,1] et devrait être maximale.

- **La pureté**

La pureté de cluster $Cj \in C$ est définie comme le pourcentage du type de données prédominant selon la classe réelle connue $Ri \in R$, qui est :

$$Pur\ (C_j)=max_{R_i \in R} \frac{N_{ij}}{N_J}$$

Nj : est la taille du cluster *Cj*

Ni : j est le nombre des points de données de la classe *Ri* dans ce cluster.

La pureté P(C) d'une partition entière est alors calculée comme la pureté moyenne de tous les clusters. Elle est limité à l'intervalle] 0,1] et devrait être maximale [21].

- **L'entropie**

En outre, le degré relatif d'aspect aléatoire du partitionnement peut être évalué en utilisant la notion d'entropie de cluster. C'est une mesure plus complète que la pureté, car elle tient compte de la distribution de toutes les classes dans chaque cluster. L'entropie d'un cluster est :

$$Entr\ (C_j) = -\frac{1}{\log(N)} \sum_{R_i \in R} \frac{N_{ij}}{N_j} \log\left(\frac{N_{ij}}{N_j}\right)$$

Et, encore, l'entropie globale E(C) est calculée en faisant la moyenne des entropies de clusters. L'entropie de cluster est limitée à l'intervalle [0,1] devrait être minimale [21].

- **L'indice Rand**

Pour N relations, la mesure de Rand Index a pour but de comparer la manière dont les N (N −1)/2 paires de relations sont regroupées dans le clustering obtenu par rapport au clustering de référence, l'objectif étant de maximiser le nombre de relations similaires rassemblées dans le même cluster et le nombre de relations dissimilaires séparées.

On mesure les regroupements de paires de relations selon quatre décisions :
Vrai Positif (VP) si deux relations similaires sont dans un même cluster, Vrai Négatif (VN) si deux relations dissimilaires sont dans deux clusters différents, Faux Positif (FP) si deux relations dissimilaires sont dans un même cluster et Faux Négatif (FN) si deux relations similaires sont dans des clusters différents. VP et VN sont des décisions correctes alors que FP et FN correspondent à des erreurs de regroupement. La mesure [23] Rand Index est alors définie par :

$$Rand\ Index = \frac{VP+VN}{VP+FP+FN+VN}$$

À partir de ces décisions, des scores de précision P, rappel R et F-mesure peuvent aussi être définis de la façon suivante :

$$P = \frac{VP}{VP+FP}\ , R = \frac{VP}{VP+FN}\ ,\quad F = \frac{2.P.R}{P+R}$$

Évidemment, *Rand Index* est limité à l'intervalle [0,1].

I.4.2. Mesures interne

Si on aborde des jeux de données dont la structure réelle est inconnue, l'évaluation des résultats de clustering devient beaucoup plus compliquée.

Les mesures qui peuvent être appliquées dans ces cas essayent de capturer les deux objectifs d'analyse de cluster : la minimisation de la distance intra cluster (qui résulte aux clusters compacts) et la maximisation de la distance inter-cluster (qui résulte aux clusters bien séparés).nous abordons quelques mesure :

- **La variance intra cluster**

La variance intra cluster optimisée implicitement par l'algorithme K-means, est basée sur le concept de la minimisation de la distance intra clusters. Elle est donnée par :

$$Var(c)=\sum_{C_i \in C} \sum_{P_j \in C_i} d(P_j - \mu_i)^2$$

Où *pj* dénote un point de données, *c* dénote le nombre de clusters, *μi* représente le centroïde de cluster *Ci*, *d (,)* est la fonction de distance utilisée pour calculer la déviation entre le point de données *pj* et le centroïde *μi*.

La limite inférieure de la variance qu'on peut obtenir dépend des données et le nombre de clusters employé, mais elle peut au meilleur être égale au zéro [24].

- **La connectivité**

La mesure de connectivité de cluster évalue le degré auquel des points de données voisins ont été placés dans le même cluster. Elle est calculée par la formule suivante :

$$\text{Conn}= \sum_{i=1}^{m} \sum_{l=1}^{L} P_{i,nn_{i(1)}} \quad \text{, ou } p_{r,s} = \begin{cases} 1/1 \;\; si \; \neg \exists C_j; r, s \in C_j \\ 0 \; sinon, \end{cases}$$

nni(l) est leme le plus proche voisin du point de données pi et L est le paramètre déterminant le nombre des voisins qui contribuent à la connectivité. m est le nombre des éléments max. La connectivité devrait être minimale [23].

Mais dans la plupart des évaluations expérimentales des algorithmes, des jeux de données 2D sont employés pour que le lecteur soit capable de vérifier visuellement la validité des résultats (c.-à-d., à quel point l'algorithme de clustering a découvert les clusters du jeu de données). Il est clair que la visualisation du jeu de données est une vérification cruciale des résultats de clustering. Dans le cas de grands jeux de données multidimensionnels (par exemple plus de trois dimensions) la visualisation efficace du jeu de données serait difficile. [25]

CHAPITRE II

Techniques de visualisation

II.1.Introduction

Chez les êtres humains, la vision est l'un des sens les plus développés. La grande capacité de l'homme à visualiser les informations très développées joue un rôle majeur dans ses processus cognitifs (la perception visuelle : reconnaissance rapide de motifs, couleurs, formes et textures) [27]. Il n'hésite pas à utiliser des méthodes graphiques afin de mieux appréhender des notions souvent abstraites.

La volumétrie des informations est telle que leur interprétation est de plus en plus hors de portée des capacités humaines. C'est pour ces raisons que les technologies de visualisation de l'information, en plein essor, méritent notre attention.

Aujourd'hui, les techniques reposant sur la visualisation deviennent de plus en plus importantes pour l'exploration d'ensembles de données multidimensionnelles de grande taille.

Nous traitons dans ce chapitre le modèle de visualisation, les différentes techniques de visualisation des données multidimensionnelles et les facteurs essentiels pour une bonne visualisation.

II.2.Définition de la visualisation

Tout d'abord la définition de la visualisation: c'est la présentation visuelle sur un écran des résultats d'un traitement sous forme alphanumérique ou graphique; Selon le dictionnaire de CRNTL[1]. Il y a plusieurs définitions de la visualisation, le verbe visualiser dans le dictionnaire[2] est rendre visible de manière claire (En Anglais : to visualize), son deuxième sens est faire paraître des éléments sur un écran (Informatique). (En Anglais : to display)[27].

Dans la littérature, définir le terme de la visualisation par l'utilisation interactive, assistée par l'ordinateur, des représentations visuelles de données pour amplifier la cognition [28]. Ces définitions montrent que la visualisation est une activité cognitive facilitée par une

[1] Dictionnaire CNRTL : http://www.cnrtl.fr/definition/
[2] Dictionnaire Linternaute: http://www.linternaute.com/dictionnaire/fr/definition/visualiser/

cognitive facilitée par une représentation graphique externe pour aider les utilisateurs et les analystes de construire une représentation mentale interne sur le monde. [28]

II.3.Classification de la visualisation

La projection de la visualisation sur le processus de développement des connaissances elle nous permet de distinguer 3 catégories de la visualisation:

- **La visualisation scientifique [29]**

Permet de comprendre les phénomènes physiques dans les données qui se basent sur des modèles mathématiques.

- **La visualisation d'information [30]**

Qui vise à explorer les données et les informations sous forme graphique qui permet aussi d'extraire et d'identifier les tendances, les corrélations, et structures abstraites dans les données.

- **La visualisation de la connaissance [31]**

Est utilisée pour désigner tout procédé permettant de présenter une structure de connaissances (comme d'un expert à des étudiants) ou encore, comme moyen pour évaluer soi-même des connaissances et aider à la compréhension et à la navigation.

Dans notre cadre d'étude, on se concentre sur la visualisation d'information que nous allons détailler dans la section suivante.

II.4. Définition de la visualisation d'information

Edward R. Tufte définit la visualisation d'information comme *"Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space."*,qui signifie la visualisation la plus parfaite c'est celle qui nous donne le plus grand nombre d'idées dans les plus brefs délais avec le moins d'encre dans le plus petit espace (moins de jeu de couleur dans le plus petit espace).[30]

La visualisation de l'information permet aux gens de devenir plus facilement connaissance de faits essentiels, à voir rapidement des régularités et les valeurs aberrantes dans les données, et donc de développer une compréhension plus profonde des données. Visualisation interactive prend en outre l'avantage de la capacité des gens d'identifier également des faits intéressants lorsque les changements d'affichage visuel, et leur permet de manipuler la visualisation ou les données sous-jacentes à explorer ces changements. **[30]**

II.5.Les objectifs de la visualisation de l'information [32]

Il s'agit de fournir à l'utilisateur une compréhension qualitative du contenu de l'information.

- Cette visualisation doit permettre à l'utilisateur final de faire des découvertes, proposer des explications ou prendre des décisions.
- Ces actions peuvent se faire aussi bien sur des motifs (clusters, tendances, émergences, anomalies) ou sur des ensembles d'éléments ou encore sur des éléments isolés.
- Communiquer efficacement des informations au travers d'une représentation graphique via des cartes cognitives.

II.6.Les facteurs essentiels pour une bonne visualisation [33]

Des interfaces visuelles intuitives peuvent réduire de manière significative la charge cognitive de l'utilisateur lorsqu'il travaille avec des systèmes complexes. La visualisation est une technique prometteuse, à la fois pour améliorer la perception de la structure dans de grands espaces d'informations et pour faciliter la navigation. Elle permet également aux utilisateurs de mettre en œuvre des outils naturels d'observation et de traitement (leurs yeux et leur cerveau). Les techniques de visualisation proposées pour la visualisation d'informations peuvent être divisées en trois catégories : présentation, confirmation et analyse exploratoire.

- Les techniques de ***présentation*** supposent que les faits représentés sont fixés a priori. Les données étant connues, il ne reste dans ce cas qu'à choisir la technique de visualisation la mieux adaptée pour les représenter.
- Les techniques de ***confirmation*** supposent que l'utilisateur a formulé des hypothèses sur les données. L'objectif de la visualisation est alors de l'aider à confirmer ou rejeter ces hypothèses.
- **L'*analyse exploratoire*** traite des données sur lesquelles l'utilisateur n'a fait aucune hypothèse. Grâce à une exploration interactive, l'utilisateur forme des hypothèses, ensuite révélées par une visualisation appropriée. Ce mode de visualisation doit permettre de déceler parmi les données des informations implicites mais potentiellement utiles. [34]

II.7.Modèle de la visualisation

Le processus de la visualisation de l'information est un sujet de recherche qui a été enrichie par les travaux P.K Robertson & De Ferrari, 1994 ou encore par Ed Chi qui a introduit le modèle «Data State Reference Model »(2000) et aussi le travail de Card, Mackinlay et Shneiderman,1999 qui propose le modèle de «Data Flow » [36];ce dernier est le modèle de référence le plus utilisé dans plusieurs travaux de recherches[32][35] car il est simple à mettre en œuvre et montre clairement les différentes étapes de processus depuis la transformation des données source jusqu'à la visualisation (la figure II.1).

Celui-ci part des données brutes pour arriver à une image représentant les données. Pour cela trois étapes sont nécessaires :

1. Transformation des données ("Data transformation").
2. La cartographie Visuelle ("Mapping").
3. le Rendu ("Rendering").

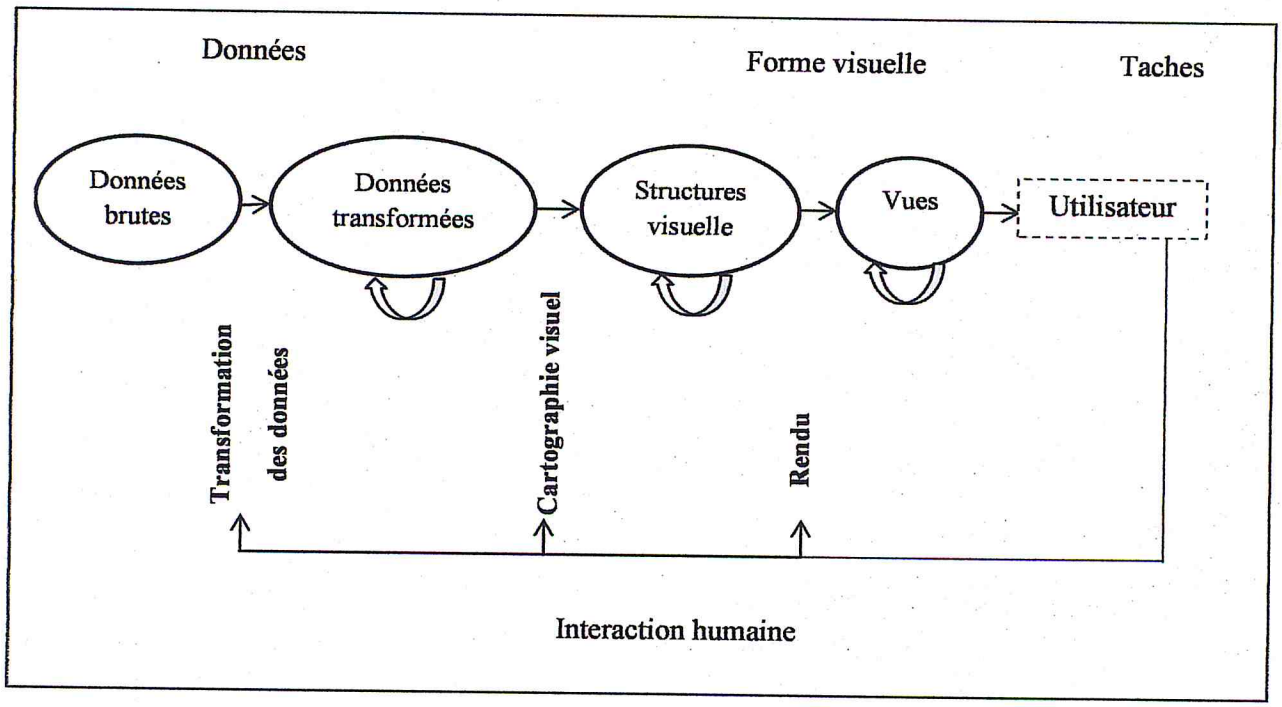


Figure.II.1 : Processus de visualisation de l'information. « Data Flow» [36]

II.7.1.Les étapes de processus

II.7.1.a. Transformation des données

La première étape consiste à transformer les données brutes depuis des sources structurées on non structurées vers des données unifiées. Cette étape produit des données uniformes après l'application de plusieurs opérations comme la sélection des

caractéristiques (attributs) ou la réduction de dimensions, la projection, l'agrégation, l'échantillonnage, le résumé des données et aussi la normalisation des données; surtout lorsque les données sont fortement multidimensionnelles.

II.7.1.b. Cartographie visuelle

La cartographie visuelle est l'étape de base du processus où les dimensions sont mappées sur les aspects visuels pour former des structures visuelles. Les opérations les plus communes à cette étape sont: la génération d'ordres; en attribuant les dimensions données à des axes de visualisation dans des ordres différents. En général, des alternatives peuvent être générés en considérant l'ensemble des caractéristiques visuelles (par exemple, la couleur, taille, forme). [37]

II.7.1.c. Rendu (Transformation des vues)

Cette étape transforme des structures visuelles en vue de spécifier les propriétés graphiques qui transforment ces structures en pixels. A ce stade des vues alternatives de mêmes structures peuvent être générées automatiquement.

II.7.2.Les éléments de processus

Nous allons maintenant décrire les différents éléments utilisés par le processus de la visualisation d'information.

II.7.2.a. Données brutes

Dans la visualisation de l'information, les sources de données se composent d'un grand nombre d'enregistrement constitués chacun d'un certain nombre de variables ou dimensions. Le nombre d'attributs peut varier d'un ensemble de données à un autre on parle de dimensionnalité de l'ensemble des données. Il peut aussi avoir des types plus complexes de données comme du texte / hypertexte ou des hiérarchies ou des graphiques constitués. Dans la littérature, il existe plusieurs taxonomies du type de données pour la visualisation, Selon le nombre de dimensions (aussi appelées attributs ou variables [38]) [39] [30] distinguent sept types de données tel que:

- ***Données unidimensionnelles***

Les données unidimensionnelles ont généralement une dimension dense, l'exemple typique de ce type de données est la donnée temporelle. A noter qu'à chaque point de temps, un ou plusieurs valeurs de données peuvent être associées. Les histogrammes sont une des techniques de visualisation de données unidimensionnelles les plus connues.

- ***Données bidimensionnelles***

Les données bidimensionnelles ont deux dimensions distinctes. L'exemple typique de ce types de données est la représentation es données géographiques où les deux dimensions de longitude et latitude sont représentées de manière distincte. Les graphique X Y (scatter plots) sont une méthode typique pour représenter des données bidimensionnelles, les cartes géographiques sont un type spécial de plan X. bien que ces données soient faciles à traiter, la prudence est conseillée car si le nombre d'enregistrements à visualiser est grand, les axes du graphiques deviennent illisibles et la compréhension s'amoindrit.

- ***Données multidimensionnelles***

De nombreux ensembles de données se composent de plus de trois attributs et par conséquent, ils ne peuvent être visualisés par le biais d'un plan en deux ou trois dimensions. Exemple de données multidimensionnelles (ou multi variées) : les tables de bases de données relationnelles, qui, souvent, contiennent des dizaines de colonnes.et les données peuvent être de différentes natures : numériques, symboliques, discrètes, continues, relations entre données, variations des données. Etant donné qu'il n'est pas aisé de cartographier des attributs ayant plus de deux dimensions, des techniques de visualisation plus sophistiquées sont nécessaires. Dans le cadre de notre projet, on détaillera plus les techniques de visualisation de données multidimensionnelles dans la suite.

- ***Données texte/hypertexte***

Le type de données n'est pas réduit qu'aux dimensions, à l'ère du World Wide Web, le texte et hypertexte devient un type de données très important. Ce type de données diffère des autres types de données et ne peut pas être décrit en nombres et par conséquent, la plupart des techniques de visualisation standards ne peuvent pas être appliquées. Dans la plupart des cas, une transformation de données dans la description des vecteurs est nécessaire avant que les techniques de visualisation ne soient appliquées. Une des techniques les plus connues et le ThemeRiver [40]

- ***Données hiérarchiques et graphiques***

Les enregistrements ont souvent une relation avec d'autres éléments d'information. Les graphiques sont largement utilisés pour représenter ces interdépendances. Un graphe est composé d'un ensemble d'objets, appelés nœuds et les connexions entre ces objets sont appelées arcs. Des exemples pour illustrer ce type de données sont les interrelations par e-

mail entre les gens, leur comportement d'achat, les structures de fichiers du disque dur ou les liens hypertexte dans la world wide web. Il existe un certain nombre de techniques de visualisation spécifiques qui traitent des données hiérarchiques et graphiques telles que le treemap qui reste une technique très utilisée pour visualiser ce type de données. Des outils tels que le framework « Scalable » proposent des visualisations de graphes et de données hiérarchiques.

- ***Données algorithmiques et logiciels***

Une autre catégorie de données est les algorithmes et les logiciels. Faire face à de grands projets de logiciels est un défi. Le but de la visualisation de ce type de données et de soutenir le développement de logiciels en les aidant les développeurs à comprendre les algorithmes, par exemple, en montrant la circulation de l'information dans un programme afin d'améliorer la compréhension du code écrit, ou en représentant la structure des milliers de lignes de code source sous forme de graphiques et de soutenir le programmeur dans le débogage du code comme en visualisant les erreurs, on peut trouver ces techniques utilisées dans « Polaris ».

II.2.7.b. Structure visuelle

Le processus de cartographie visuelle se traduit par des structures graphiques qui représentent l'information [41]. Dans la dernière étape, les vues rendent ces structures graphiques et les rendre accessibles à l'observateur humain, sur les écrans d'ordinateur, par exemple. Voir transformations spécifie les paramètres graphiques qui influencent la vue comme la position et l'échelle [31].

II.8.Techniques de visualisation des données multidimensionnelles

Plusieurs efforts ont été réalisés sur la visualisation de données multidimensionnelle ($d > 3$) [42].

Cependant, les techniques de la visualisation classique ont de la difficulté à traiter des ensembles de données multidimensionnelles. Plusieurs techniques existe tel que :

II.8.1.Techniques à base d'icônes

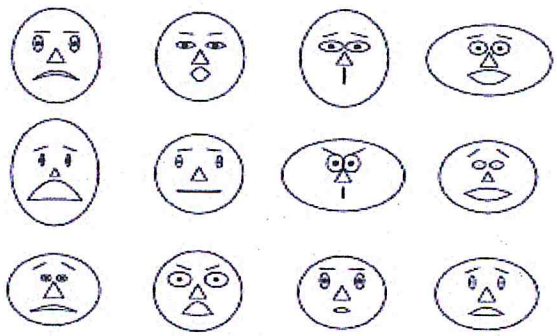
Présentations à base d'icônes sont des techniques relativement âgées de data mining visuel. L'idée des techniques à base d'icônes est de cartographier chaque élément de données multidimensionnelles comme une icône. Plusieurs technique existe tel que :

- **Figures de Chernoff** [43]

Ce paradigme de représentation d'espaces multidimensionnels présenté ici se nomme les « figures de Chernoff »Il exploite l'habitude et la forte capacité de l'homme à percevoir de très légers changements dans les expressions faciales.

Une figure de Chernoff s'obtient en associant à chaque variable de la table de donnée un trait d'une expression faciale. Par exemple, la forme de la figure représente la première variable, la taille des yeux représente la deuxième et ainsi de suite pour chaque variable. En utilisant ainsi les traits les plus frappants d'un visage, il est possible de représenter un nombre de dimensions largement supérieur à trois.

L'avantage des faces de Chernoff est la possibilité de condenser les données, il s'en suit une facilitée de compréhension pour l'usager. L'inconvénient majeur réside dans la subjectivité de l'affectation des expressions faciales aux différentes variables constituants les données l'impossibilité de représenter une dizaines d'attributs.



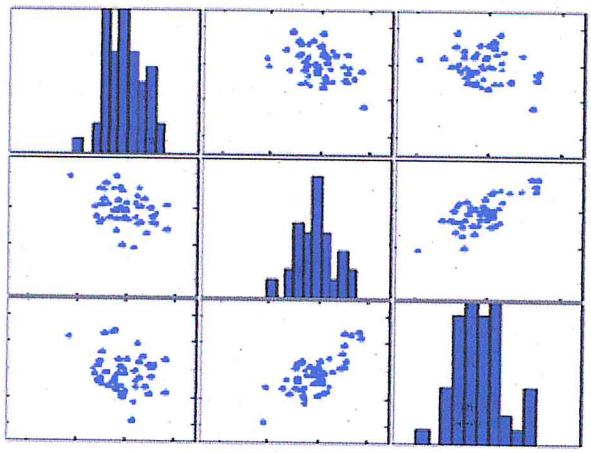
FigureII.2. Figures de Chernoff

II.8.2.Techniques géométriques

Techniques d'affichage géométriques transformées visent à trouver des transformations "intéressants" d'ensembles de données multidimensionnelles. Plusieurs techniques existe tel que :

- **Matrice des diagrammes de dispersion (Matrix of scatter polts)** [44]

Une matrice de diagrammes de dispersion est un tableau d'affichage de diagrammes de dispersion toutes les paires possibles de dimensions ou de coordonnées. Est nommé aussi nuage de point ou Les valeurs des attributs déterminent la position. On peut utiliser plusieurs nuages de points pour résumer de façon compacte les relations entre plusieurs paires d'attributs.

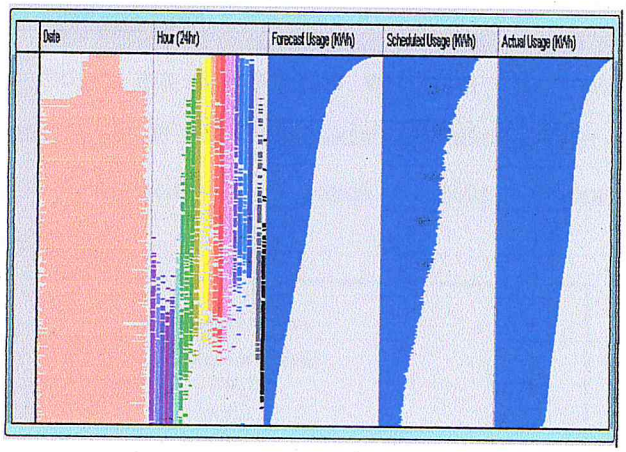


FigureII.3. Matrice de Scatter polts.

- **Lens table** [45]

Permettant la visualisation de larges ensembles de données tabulaires et où chaque ligne d'un pixel de largeur correspond à une donnée.

On peut filtrer les lignes en fonction de certains paramètres représentés par des caractéristiques graphiques, on peut trier les colonnes, zoomer en cliquant sur une ligne haute de 1 pixel et visualiser la ligne du tableur correspondant à la donnée.



FigureII.4. Lens table [44].

- **Les coordonnées parallèles [Parallel Coordinates]** [45]

La technique des « coordonnées parallèles » est une autre technique utilisée pour représenter des espaces multidimensionnels .Les dimensions de l'espace sont représentées par autant d'axes verticaux placés parallèlement.

II.9.Exemple sur les systèmes (software) existants

Dans le domaine que nous traitons il existe de nombreux softwares qui proposent différentes approches afin de représenter les données multidimensionnelles. Au cour de notre travail nous nous sommes intéressé à quelques outils qui sont les suivants :

II.9.1.Ggobi [47]

Ggobi est un logiciel de visualisation open source pour l'exploration des données de grande dimension. Il fournit des graphiques dynamiques et interactifs comme des tours, ainsi que des graphiques familiers tels que le nuage de points, barchart et les coordonnées parallèles. Les tracés sont interactifs et liés avec des brosses et d'identification.

II.9.2.Orange [48]

C'est un logiciel libre d'exploration de données (data mining). Il propose des fonctionnalités de modélisation à travers une interface visuelle interactive d'ensemble des données multidimensionnelles, une grande variété de modalités de visualisation et des affichages variés dynamiques. Développé en python, il existe des versions Windows, Mac et Linux. . Il prend en charge plusieurs méthodes pour afficher des données de forme plate et de données en cluster hiérarchique, et il prend en charge une variété d'interaction : zoom, filtrage,……

II.9.3.Xmdv tool [49]

Est un logiciel du domaine public pour l'exploration visuelle interactive d'ensembles de données multidimensionnelles. Il est disponible sur toutes les grandes plates-formes comme UNIX, Linux, Mac et Windows. Xmdv Tool est développé en utilisant Qt et Eclipse CDT. Il prend en charge cinq méthodes pour afficher des données de forme plate et de données en cluster hiérarchique.

1. Les diagrammes de dispersion.
2. Glyphes étoiles.
3. Coordonnées parallèles.
4. dimensions empilage.
5. Affichage Pixel-orienté.

prend également en charge une variété de modes d'interaction et d'outils, y compris le brossage à l'écran, les données et les espaces de la structure, le zoom, le panoramique et techniques de distorsion, et le masquage et la réorganisation des dimensions il a été appliquée à un large éventail de domaines d'application, dont certains sont mis en évidence dans nos études de cas. Certains de ces domaines comprennent la télédétection, la situation financière, de la géochimie, de recensement et des données de simulation.

II.10.Conclusion

Dans le cas de besoins moins clairement spécifiés, la visualisation d'informations permet de réduire la complexité d'un système grâce à sa représentation et à son exploration par les utilisateurs. Nous avons identifié les facteurs essentiels à une visualisation efficace. Ainsi, une bonne visualisation doit fournir une vue d'ensemble du système afin d'aider l'utilisateur à en avoir une compréhension générale.

CHAPITRE III

Clustering visuel de données

III.1.Introduction

Clustering de grand ensemble de données multidimensionnelles devient très important et difficile au même temps car l'interprétation des résultats de grands ensembles dans le cas de CHA utilisant le dendrogramme est assez impossible. Les techniques de visualisation de l'information contribueront à résoudre le problème. L'exploration visuelle des données a un fort potentiel d'applications car elle facilite l'analyse, l'interprétation, la validation et aussi augmente l'aspect cognitif chez les analystes.

Donc dans ce contexte, notre objectif consiste à modéliser avec UML le processus de clustering multidimensionnelle via l'utilisation des technologies de visualisation pour représenter les résultats de clustering hiérarchique ascendant afin de faciliter l'interprétation des clusters, l'évaluation de la qualité, la validation des résultats de clustering et l'intégration de l'utilisateur dans les différentes étapes de processus de clustering visuel en lui donnant la possibilité d'interagir sur les données.

III.2.Le modèle de clustering visuel [50]

Le clustering (voir figure III.1(a)) est particulièrement adapté aux problèmes d'analyse dans lequel il existe des moyens limités pour évaluer la qualité des solutions proposées. Cependant, très souvent, les résultats deviennent des boîtes noires dans les mains des utilisateurs finaux ou les algorithmes fournissent des résultats qui ne conduisent pas à une solution à ce problème, car ils ne tiennent pas en compte des connaissances des experts.
En revanche, les méthodes de visualisation (voir figureIII.1(b)) utilisent les connaissances de fond, la créativité et la visualisation des méthodes intuition pour résoudre le problème à la main. Bien que ces approches donnent souvent des résultats acceptables pour les petits ensembles, ils échouent lorsque les données fournies sont trop gros pour être capturé par un analyste humain. [51]
La (figureIII.1) compare le processus de clustering et le processus de visualisation de l'information.

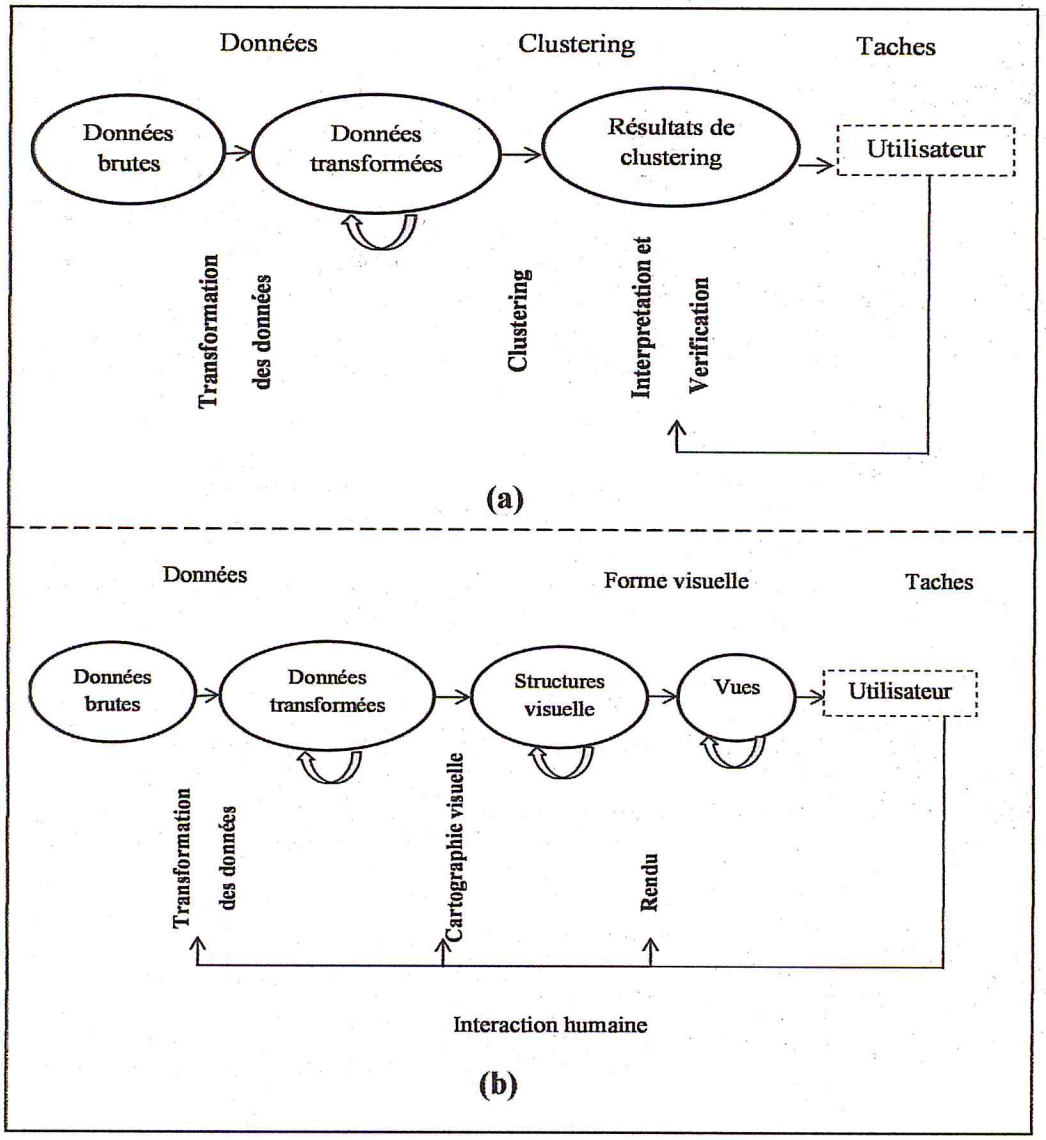


Figure III.1. Processus analytiques comparant le clustering (a) et la visualisation des informations(b). [51]

Aujourd'hui, une troisième approche a commencé à émerger pour répondre aux limite de processus de clustering (figureIII.1.(a)) et le processus de visualisation (figureIII.1 (b)), à savoir, l'approche analytique visuel(Clustering visuel) [50] (figureIII.2) vise à coupler étroitement les méthodes de clustering et les techniques visuelles interactives à travers l'interaction humaine dans le but d'acquérir des connaissances à partir des données.

La visualisation permet aux analystes d'interagir avec les méthodes automatiques en modifiant les paramètres ou sélectionnant d'autres algorithmes d'analyse. Alors, La Visualisation doit être utilisée pour évaluer les résultats des modèles générés (résultats de

clustering). Ce qui conduit à des meilleurs résultats. Le processus de clustering visuel est comme suit :

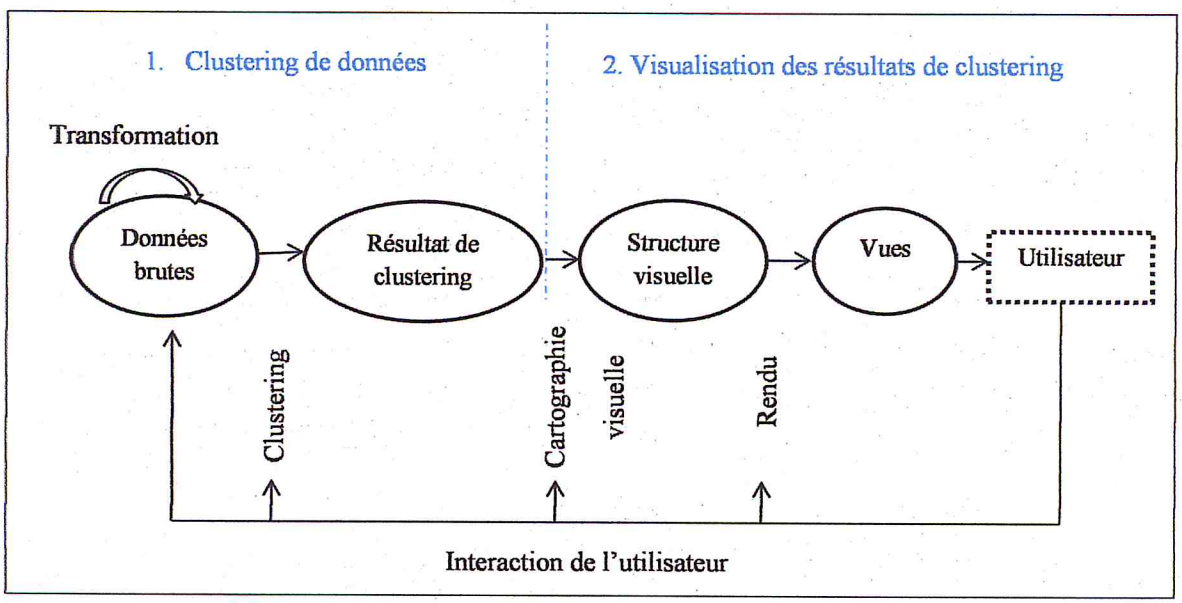


Figure III.2. Modèle de clustering visuel [51]

Dans de nombreux scénarios d'applications, les données hétérogènes et multidimensionnelles doivent être intégrés avant les méthodes d'analyse visuelle ou automatique peuvent être appliquées. La première étape c'est l'étape de prétraitement des données qui sert à transformer les données en standardisant, normalisant les données. Et permet aussi de filtrer les dimensions inutiles.

La deuxième étape c'est l'application de clustering hiérarchiques ascendants pour générer des regroupements des individus similaires (on obtient alors l'arbre des clusters).

La dernière étape de notre solution c'est la représentation visuelle des résultats de clustering hiérarchique qui permet à l'utilisateur d'obtenir un aperçu sur les données, de tirer des conclusions et d'interagir directement pour obtenir des détails sur les données à la demande en utilisant la technique de treemap qu'on va détailler ci-dessus.

III.3.Cartographie par arbre(Treemap) [60]

Les cartes d'arbres d'enveloppées par Schneiderman et son équipe [60] donnent une vue d'ensemble de l'arbre. Elles utilisent la totalité de l'espace disponible ; cet espace est alternativement divisé horizontalement ou verticalement, chaque case formée correspond à un nœud de premier niveau de l'arbre. Puis les cases qui contiennent des éléments fils sont-

elles mêmes divisées, et ainsi de suite récursivement .Chaque élément de l'arbre est donc représenté par une surface rectangulaire. Des variantes de l'algorithme existent, notamment pour le choix du sens de la division (horizontal ou vertical).

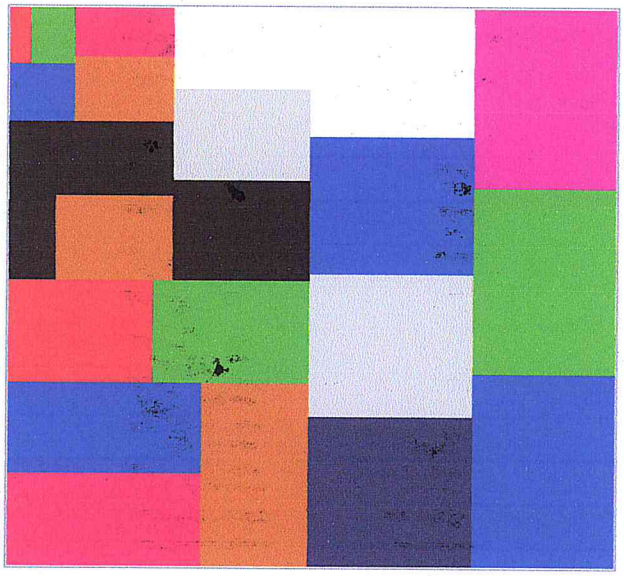


Figure III.3. Treemap

Afin de réaliser et implémenter notre solution nous modélisons ce système avec UML.

III.4. Présentation de langage de modélisation

UML (*Unified Modeling Language*) est un langage unifié pour la modélisation objet. Il se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue. [52], mais il ne définit pas le processus d'élaboration des modèles. Cependant dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche.

Pour standardiser les démarches, plusieurs modèles de démarches ont été décrits et parfois formalisés, parmi ces derniers, UP.

III.5. Présentation de la démarche utilisée

III.5.1. Le Processus Unifié [53]

Le Processus Unifié (PU ou UP en anglais pour **Unified Process**) est un processus de développement logiciel construite sur UML ; il est itérative et incrémentale, centré sur l'architecture et piloté par les cas d'utilisation.

- **Itérative et incrémentale**

La méthode est itérative dans le sens où elle propose de faire des itérations lors de ses différentes phases, ceci garantit que le modèle construit à chaque phase ou étape soit affiné et amélioré. Chaque itération peut servir aussi à ajouter de nouveaux incréments.

- **Piloté par les cas d'utilisation**

Oriente l'utilisateur pour répondre aux besoins de celui-ci.

- **Centré sur l'architecture**

Les modèles définit tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide.

L'objectif d'un processus unifié est de maitriser la complexité des projets informatiques en diminuants les risques à l'aide de définir des priorités pour chaque fonctionnalité.

Dans notre cas, Les activités de processus unifié sont basées sur un modèle de développement en cascade.

III.6.Le modèle en cascade

Ce modèle a été décrit par Royce en 1970[51], il a été largement employé depuis, pour la description générale des activités liées aux logiciels.

Le modèle en cascade permet de couvrir le cycle de vie d'un logiciel depuis l'analyse jusqu'au test. (FigureIII.4)

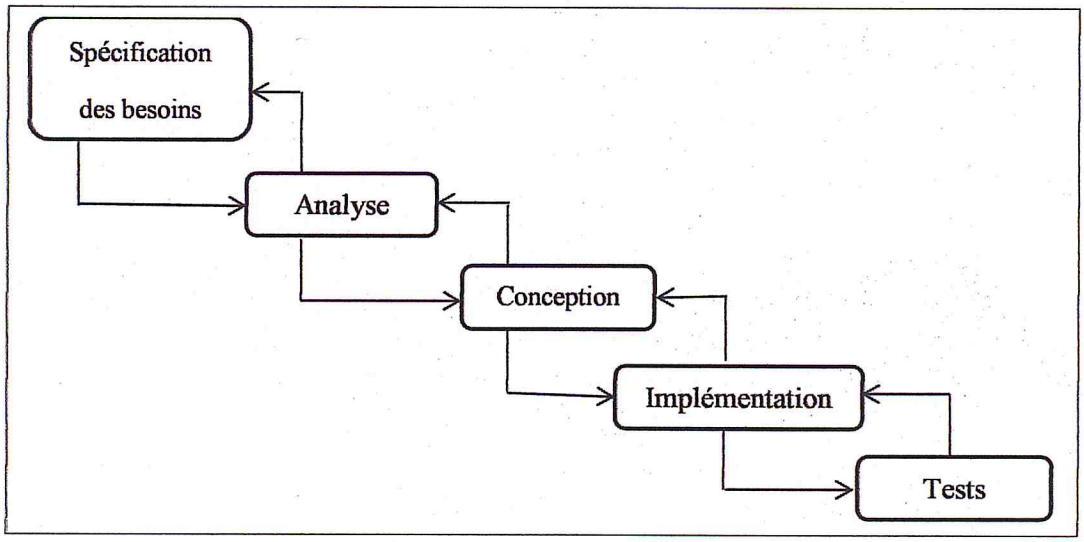


Figure III.4 : Modèle en cascade. [51]

III.6.1.Expression des besoins [54]

La spécification des besoins est une étape essentielle au début de processus de développement, elle consiste généralement à déterminer précisément les besoins des utilisateurs du système afin d'éviter de développer un logiciel non adéquat.

Cette étape ne préoccupe pas des solutions mais des questions : elle identifie le « quoi faire ?»Et identifie les entités de l'environnement du système. Pour modéliser ces besoins on utilise le diagramme des cas d'utilisation d'UML.

III.6.2. Analyse [55]

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client, il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilite la compréhension(Scenario) en utilisant le diagramme de séquence d'UML pour représenter les interactions entres les objets.

III.6.3.Conception [55]

C'est la phase la plus importante du processus de développement d'un logiciel. Elle s'intéresse d'abord au «comment ? », à savoir la solution du problème énoncé.

La conception a pour but de décomposer le logiciel en module, de préciser les interfaces et les fonctions de chaque module. A l'issue de cette étape, on obtient une description de l'architecture du logiciel et un ensemble de spécifications de ces divers composants en utilisant le diagramme de classe d'UML.

III.6.4.Implémentation [56]

L'implémentation est le résultat de la conception pour implémenter le système sous forme de composants, c'est-à-dire, de code source, de scripts exécutables et d'autres éléments du même types.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous forme de code source.

III.6.5.Tests [54]

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les

implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

III.7.Expression des besoins

Cette phase consiste à définir les besoins fonctionnels de notre futur système, nous allons parler des fonctionnalités que peut offrir ce dernier afin de bien connaître les acteurs qui interagissent avec lui par la suite nous allons les modéliser en utilisant le diagramme des cas d'utilisation d'UML.

III.7.1.Identification des acteurs

La première étape de cette phase permet d'énumérer les acteurs susceptibles d'interagir avec le système. Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système).qui interagissent directement avec le système étudié. [54]

- **Les acteurs**

Acteur	**Désignation**
Utilisateur	Les utilisateurs sont des utilisateurs non privilégiés.

Tableau III.1 : les acteurs de système.

III.7.2.Identification de cas d'utilisation

L'identification de cas d'utilisation donne un aperçu des fonctionnalités futures que doit implémenter le système. Cependant, il nous faut plusieurs itérations pour ainsi arriver à constituer des cas d'utilisation complets.

Dans notre cas d'étude nous avons défini le diagramme de cas d'utilisation général et détaillé. Pour les cas globaux, on distingue :

- L'importation du fichier.
- Prétraitement des données.
- Spécification de paramètres de clustering.
- Spécification de paramètres de visualisation.
- Exploration les résultats de visualisation.

Nous détaillerons chaque cas d'utilisation qui doit faire l'objet d'une définition a priori qui décrit l'intention de l'acteur lorsqu'il utilise le système, et les séquences d'action principales qu'il est susceptible d'effectuer.

Les descriptions détaillées vont être organisées dans des tableaux de la façon suivante :

Elément	Signification
Cas d'utilisation	Le nom du cas d'utilisation.
Acteur	L'acteur qui réalise le cas d'utilisation.
But	Le but du cas d'utilisation.
Description	Une explication du cas d'utilisation.
Pré condition	Les conditions qui doivent être vérifiées afin d'effectuer les cas d'utilisation.
Post condition	Les résultats
Exception	Les informations entrées par l'acteur.

Tableau III.2. Modéle de représentation des descriptions détaillées des cas d'utilisation.

III.7.3.Diagrammes de cas d'utilisation

- **Diagramme de cas d'utilisation global**

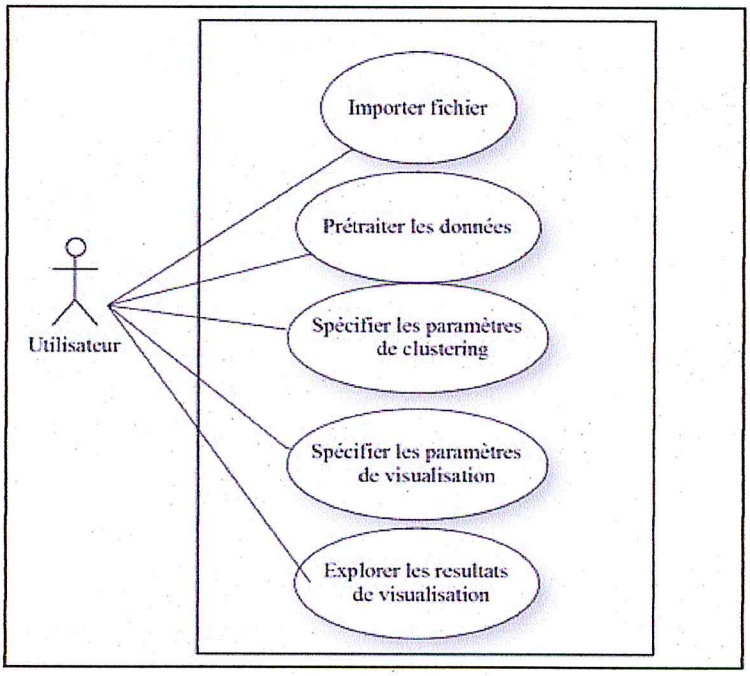


Figure III.5. Diagramme de cas d'utilisation global du système.

Importation du fichier	
Acteur	Utilisateur
But	Charger l'ensemble de données à traiter.
Description	L'utilisateur après être accéder peut : Sélectionner un fichier sur lequel le travail sera effectué.
Pré condition	Accéder au système.
Post condition	fichier importé.
Prétraitement des données	
Acteur	Utilisateur
But	Améliorer la performance de l'algorithme hiérarchique en nettoyant, standardisant et normalisant les données. pour obtenir des données nettoyées sans anomalie.
Description	Le prétraitement des données doit passer par la sélection des attributs qui vont être transformé ou filtrer puis effectuer la normalisation si nécessaire.
Pré condition	Chargement des données réussi.
Post condition	fichier nettoyé et les données sont du même type.
Spécification de paramètres de clustering.	
Acteur	Utilisateur.
But	Regrouper les individus similaires avec CHA.
Description	Le regroupement des individus doit passer par le choix de la méthode de calcul des distances et la stratégie d'agrégation.
Pré condition	Données standardisées.
Post condition	Afficher les résultats de clustering « l'arbre de cluster »ou bien le dendrogramme.
Spécification des paramètres de visualisation	
Acteur	Utilisateur.
But	L'obtention d'un schéma visuel pour les données traitées.
Description	La visualisation des résultats de clustering doit passer par la sélection des clusters à visualiser, les couleurs et le paradigme voulu.
Post condition	Présentation graphique.
Exploration des résultats de visualisation	

Acteur	Utilisateur.
But	L'utilisateur sort avec des hypothèses (informations).
Description	L'utilisateur peut interagir sur les résultats « filtrer »en spécifiant le nombre de clusters pour afficher les détails afin de sortir avec des hypothèses.
Pré condition	une vue d'ensemble graphique des données « présentation graphique ».
Post condition	L'utilisateur peut interagir sur les résultats.

Tableau III .3. Description détaillée du diagramme de cas d'utilisation global.

- **Diagramme de cas d'utilisation** « L'importation du fichier »

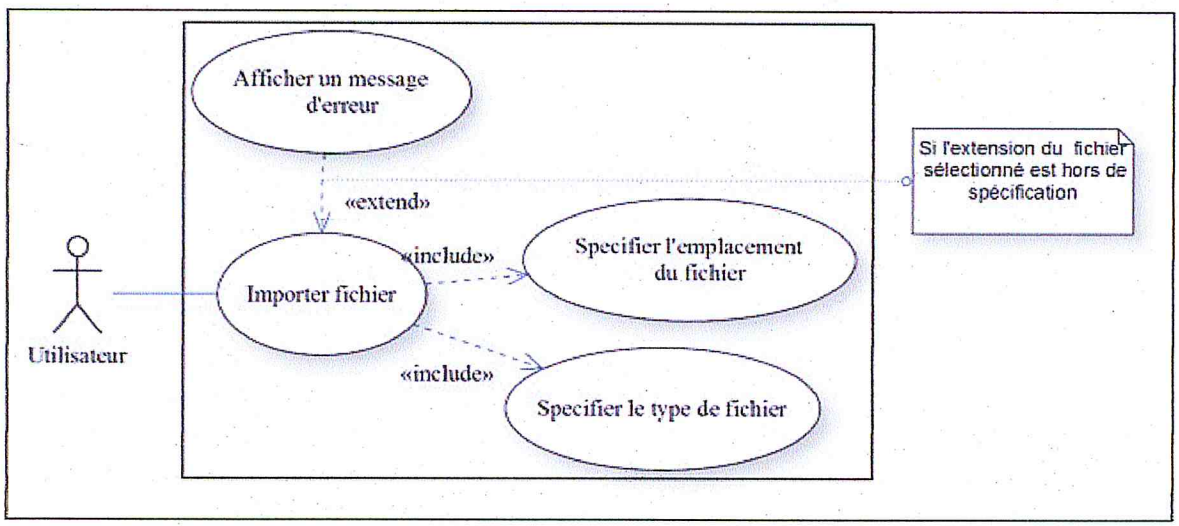


Figure III .6. Diagramme de cas d'utilisation détaillé « Importation du fichier».

Spécification de type du fichier	
Acteur	Utilisateur.
But	Choix de type de fichier à importer.
Description	L'utilisateur doit sélectionner le type de la source de données à importer.
Post condition	Afficher la boite de sélection du fichier
Spécification de l'emplacement du fichier	
Acteur	Utilisateur.
But	Localisation de la source de données à traiter.

Description	L'utilisateur doit sélectionner la source de données à traiter
Pré condition	Sélection de format de fichier.
Post condition	Retourner l'adresse du fichier(Path).

T

Tableau III .4. Description détaillé de l'importation du fichier.

- **Diagramme de cas d'utilisation** « Prétraitement des données»

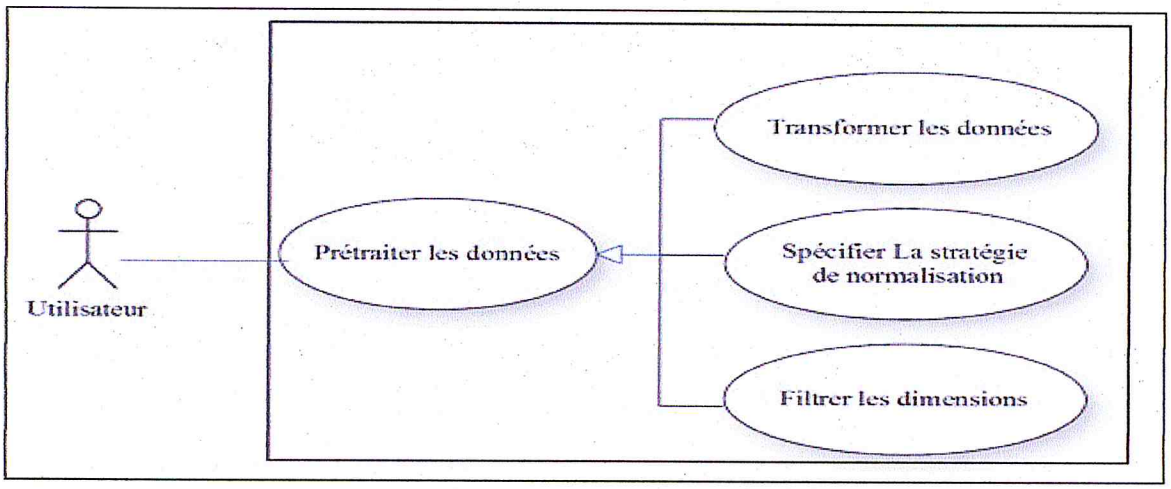


Figure III.7. Diagramme de cas d'utilisation détaillé « Prétraitement des données».

Transformation des données	
Acteur	Utilisateur.
But	Standardiser les données.
Description	L'utilisateur sélectionne les dimensions à transformer en choisissant le type de données.
Pré condition	détecter le type de chaque dimension.
Post condition	Toutes les données sont du même type.
Exception	Message d'erreur : Si le type n'est pas transformable.
Spécification la stratégie de normalisation	
Acteur	utilisateur
But	normaliser les données.
Description	L'utilisateur sélectionne le type de normalisation.
Pré condition	Données standardisées.
Post condition	Données normalisées.
Filtrage des dimensions	
Acteur	Utilisateur.

But	Réduction de dimensions.
Description	Utilisateur sélectionne les dimensions à filtrer.
Pré condition	L'ensemble de toutes les dimensions.
Post condition	Le sous ensemble de dimensions sélectionnées.

Tableau III.5. Description détaillé de prétraitement des données.

- **Diagramme de cas d'utilisation** « Spécification des paramètres de clustering »

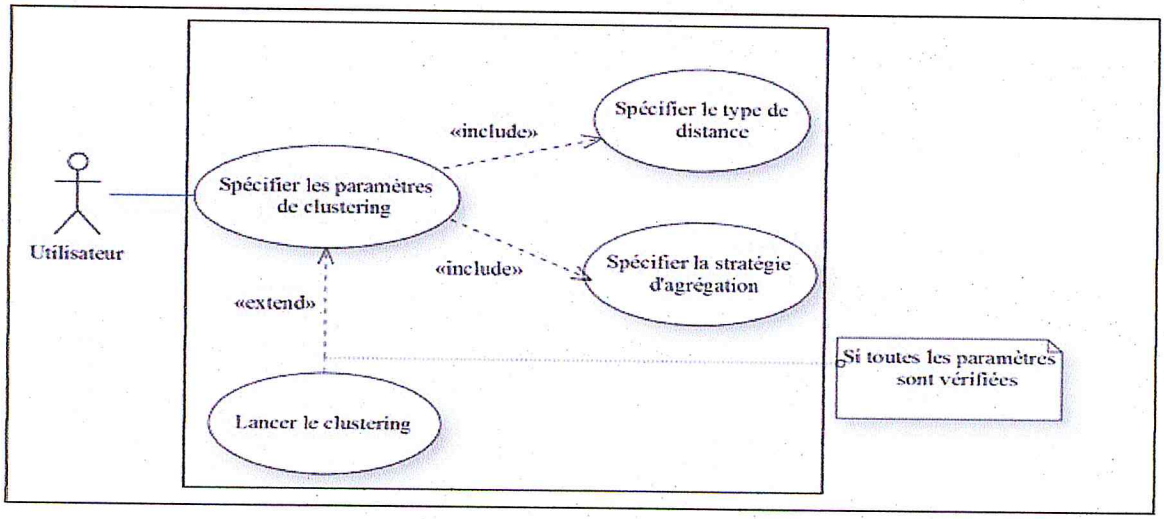


Figure. III.8. Diagramme de cas d'utilisation détaillé « spécification des paramètres de clustering

Spécification du type de distance	
Acteur	Utilisateur.
But	Calcul des distances.
Description	L'utilisateur sélectionne le type de la distance utilisée par le système pour calculer la similarité entre les individus.
Post condition	Afficher la matrice de proximité.
Spécification de la stratégie d'agrégation	
Acteur	Utilisateur.
But	Spécifier la méthode d'agrégation pour le regroupement des individus.
Description	L'utilisateur sélectionne la méthode d'agrégation utilisée par le système pour regrouper les individus.
Post condition	L'arbre des clusters.

Tableau III.6. Description détaillée « Spécification des paramètres de clustering».».

- **Diagramme de cas d'utilisation** « Spécification des paramètres de visualisation»

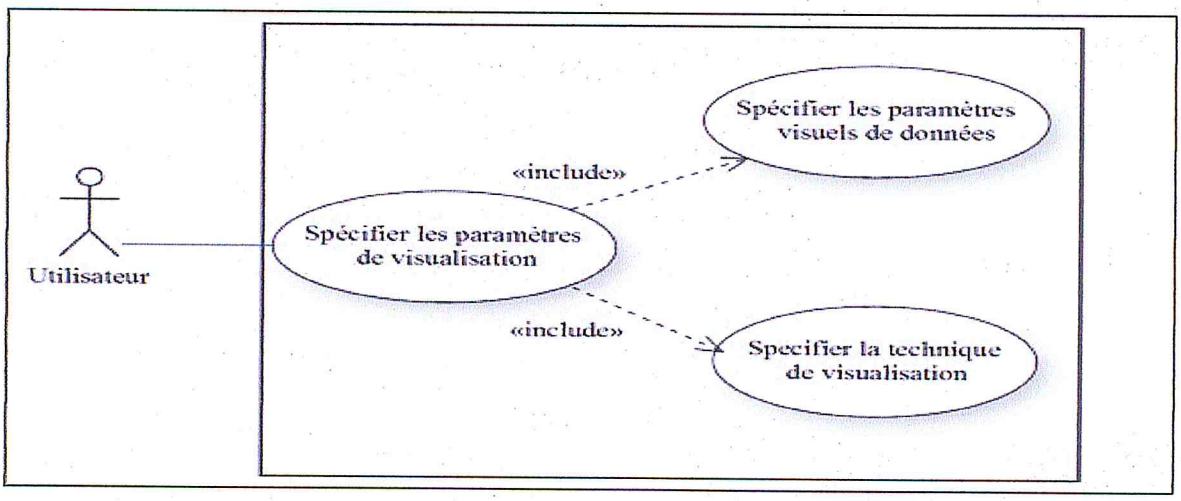


Figure. III.9. Description détaillée « Spécification des paramètres de visualisation».

Spécification des paramètres visuels de données	
Acteur	Utilisateur
But	Codifier visuellement les éléments.
Description	L'utilisateur spécifie la palette de couleur pour codifier les individus par exemple.
Pré condition	L'arbre des clusters + détails de dimensions.
Post condition	Codification visuelle des éléments.
Spécifier la technique utilisée	
Acteur	Utilisateur.
But	Visualiser les résultats.
Description	L'utilisateur sélectionne le paradigme de visualisation des résultats de clustering.
Pré condition	Les éléments visuels codifiés.
Post condition	Une vue d'ensemble. (représentation graphiques des résultats).

Tableau III.7. Diagramme de cas d'utilisation « spécification des paramètres de visualisation ».

- **Diagramme de cas d'utilisation** « Exploration des résultats de visualisation»

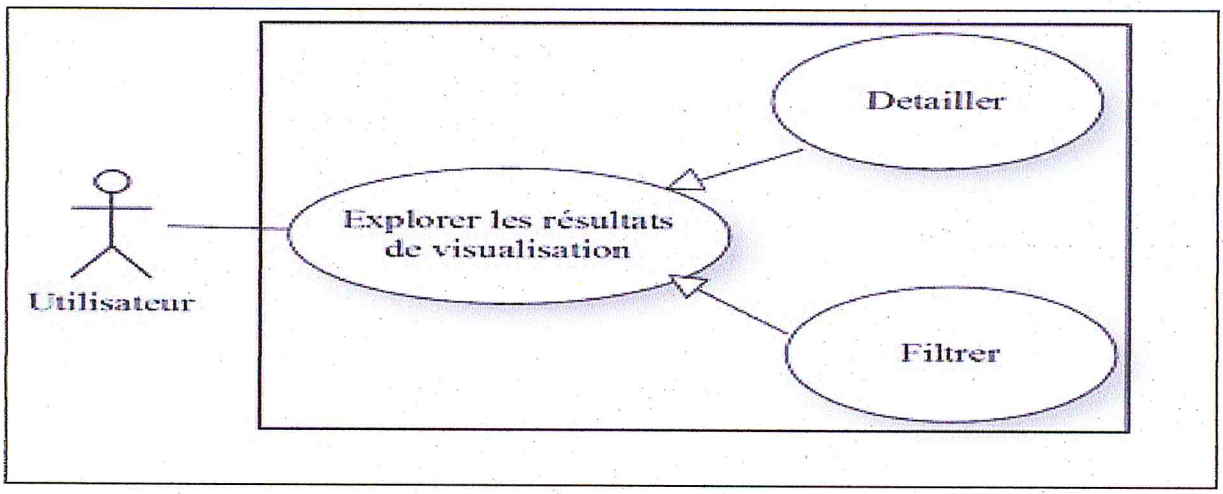


Figure III.10.*Diagramme de cas d'utilisation détaillé « exploration des résultats».*

Filtrage	
Acteur	Utilisateurs.
But	Analyser les résultats d'un sous ensemble bien précis.
Description	L'utilisateur peut choisir directement le sous-ensemble désiré à analyser ou en spécifiant des propriétés du sous-ensemble désiré.
Pré condition	La présentation graphique des résultats de clustering.
Post condition	Afficher en détails le sous ensemble selectionné.
Détails	
Acteur	Utilisateurs.
But	Voir les détails.
Description	L'utilisateur peut voir les détails (pourcentage des valeurs de chaque dimensions+ les individus) de cluster sélectionné.
Pré condition	La vue globale des données « présentation graphique ».
Post condition	l'affichage des détails.

Tableau III .8.*Description détaillé de cas d'utilisation « exploration des résultats ».*

III.8. Analyse

L'analyse permet de lister les résultats attendus, en terme de fonctionnalités, de performance, de robustesse, de maintenance,… etc.

L'analyse répond donc à la question « *que faut-il faire ?* » et a pour but de se doter d'une vision claire et rigoureuse du problème posé et du système à réaliser en déterminant ses éléments et leurs interactions.

L'analyse livre une spécification plus précise des besoins grâce à l'utilisation du diagramme de séquence. Elle peut être envisagée comme une première ébauche du modèle de conception. [57]

III.6.1.Scenarios et diagramme de séquences :

Qu'est qu'un scénario ? **[58]**

Un scénario représente un ensemble ordonné de messages échangés par des objets. On parle ici d'objet au sens large : instance de classe d'analyse ou instance d'acteur.
Les échanges de messages entre objet peuvent être représentés en UML dans une sorte de diagramme complémentaire appelé *diagramme de séquence*.

Qu'est qu'un diagramme de séquence ? **[59]**

Un diagramme de séquence est une représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et / ou de ses acteurs.
Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.

Dans ce qui suit nous allons présenter les diagrammes de séquence afin de formaliser les scénarios des cas d'utilisation vus précédemment. Nous allons voir le système comme un ensemble d'objet en interaction.

III.8.1.a. Diagramme de séquences «Importation de fichier»

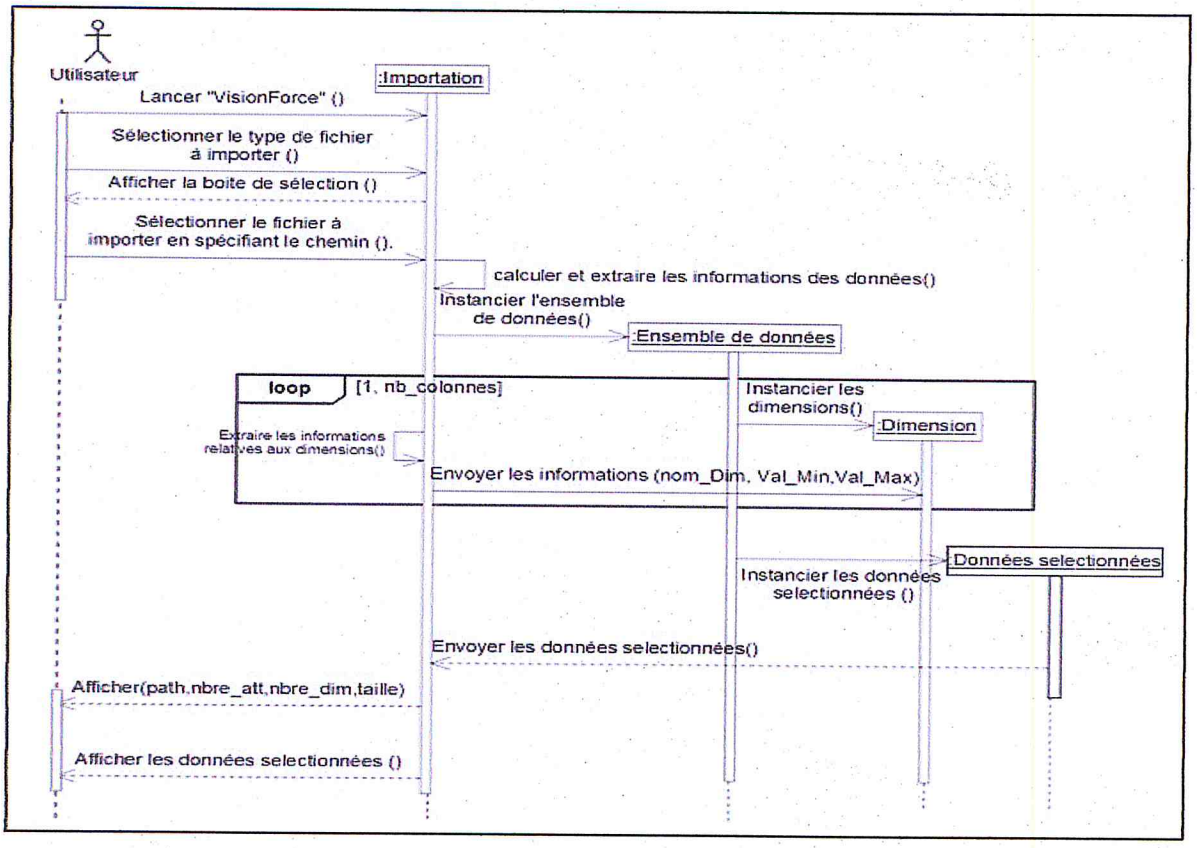


Figure III.11.Diagramme de séquence « Importation de fichier ».

III.6.1.b. Diagramme de séquences «Prétraitement des données »

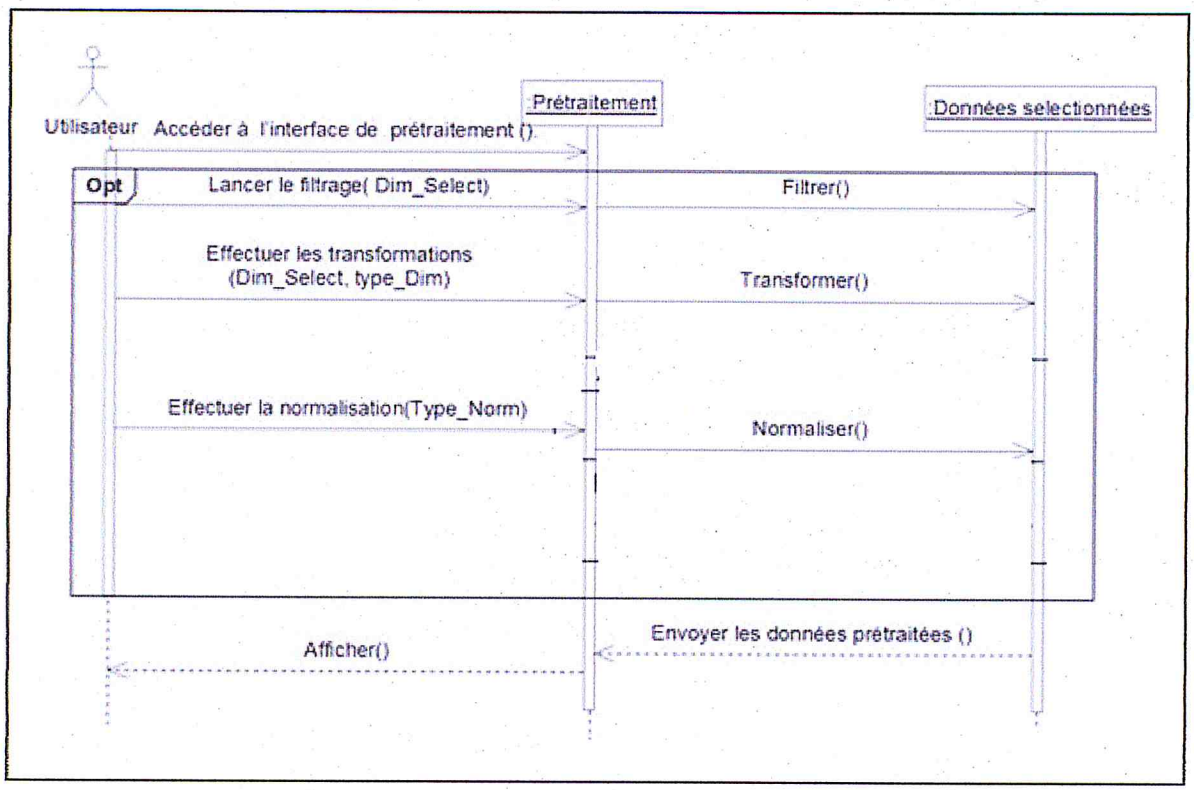


Figure III.12.Diagramme de séquence « Prétraitement des données »

III.8.1.c. Diagramme de séquences «Spécification des paramètres de clustering »

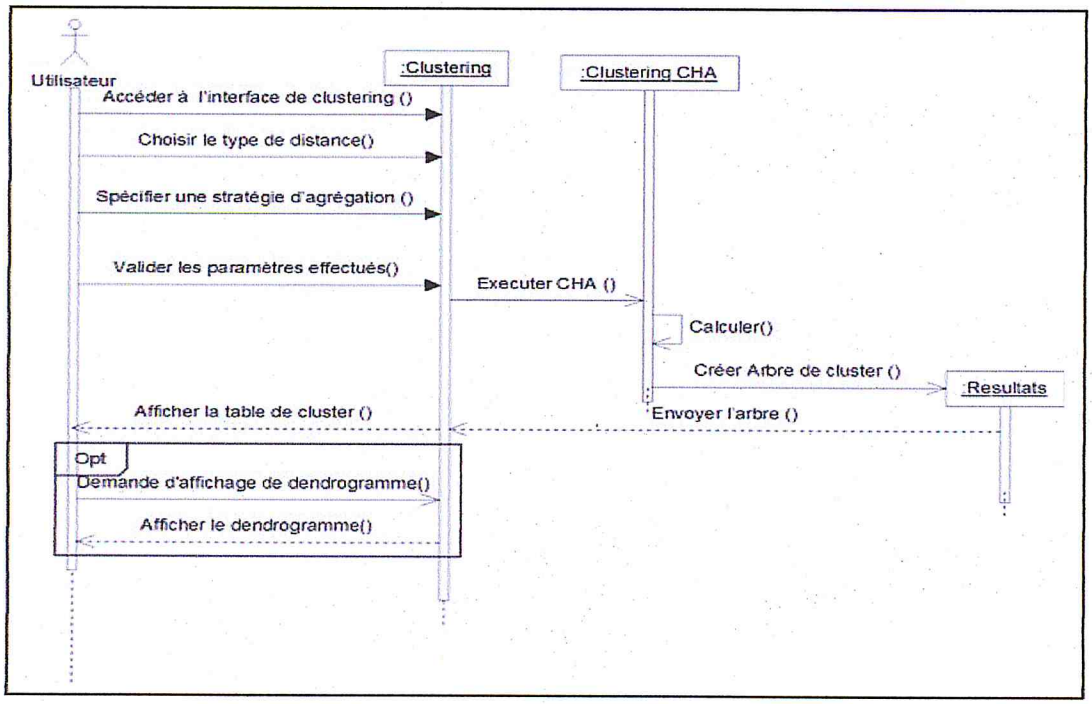


Figure III.13.Diagramme de séquence « Spécification des paramètres de clustering»

III.8.1.d. Diagramme de séquences «Spécification des paramètres de visualisation»

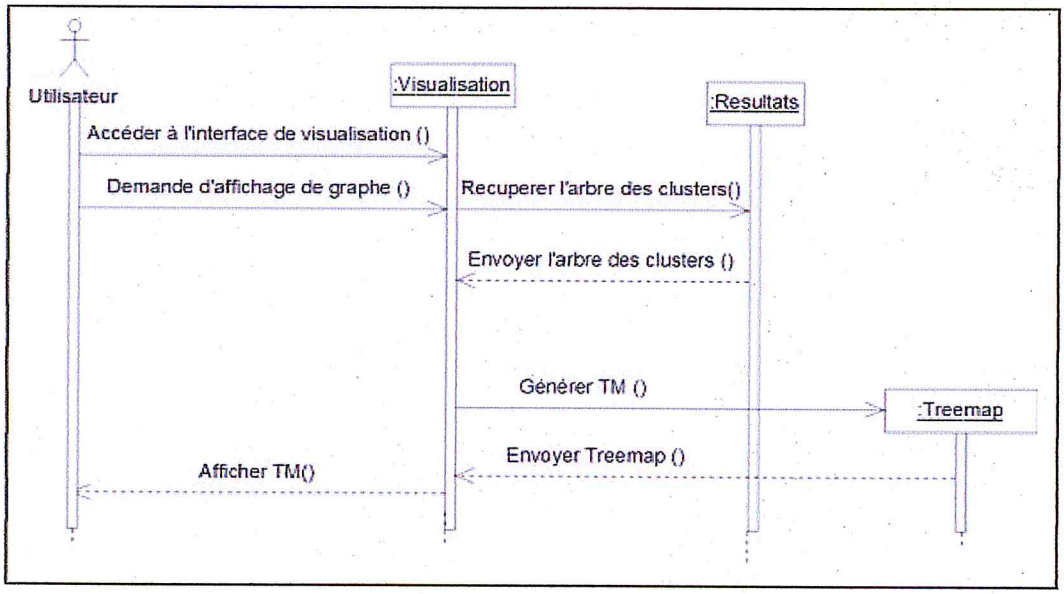


Figure III.14.Diagramme de séquence « Spécification des paramètres de visualisation »

III.8.1.e. Diagramme de séquences «Exploration des résultats de visualisation

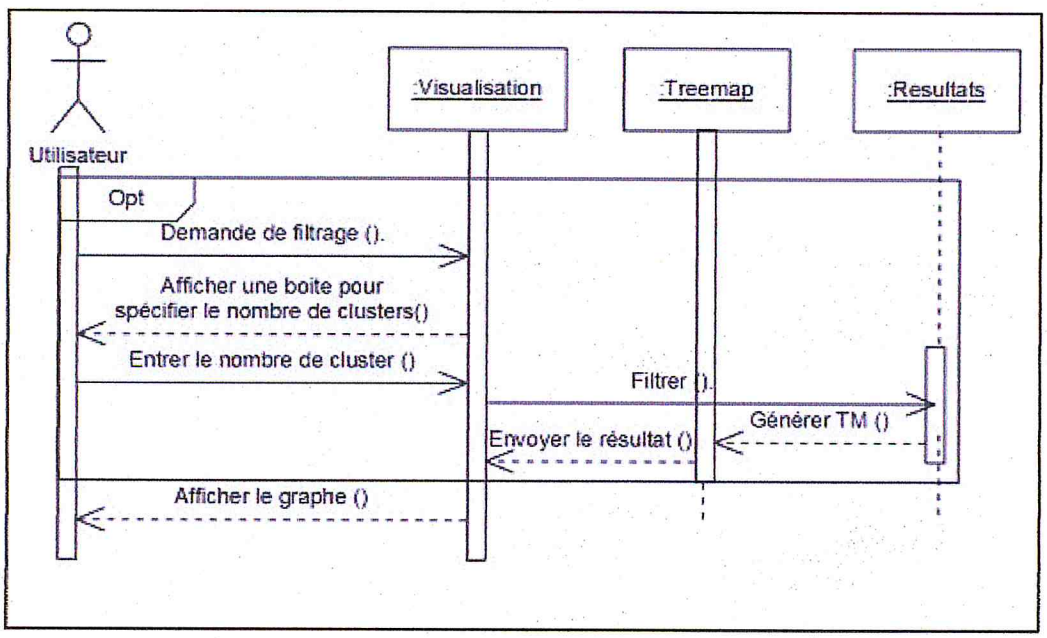


Figure III.15.Diagramme de séquences «Exploration des résultats de visualisation».

III.9.Conception du système

Nous arrivons maintenant à la phase ultime de modélisation avec UML, après la modélisation des besoins puis l'organisation de la structure de la solution, la conception consiste à construire et à documenter précisément les classes, les relations et les méthodes qui constituent le codage de la solution.

III.9.1. Diagramme de Classes

Le diagramme de classe exprime la structure de notre système VisionForce statique du système en termes de classe et de relations entre ces classes.

Il permet de structurer les informations qui sont gérées par le domaine dans des classes. [60]

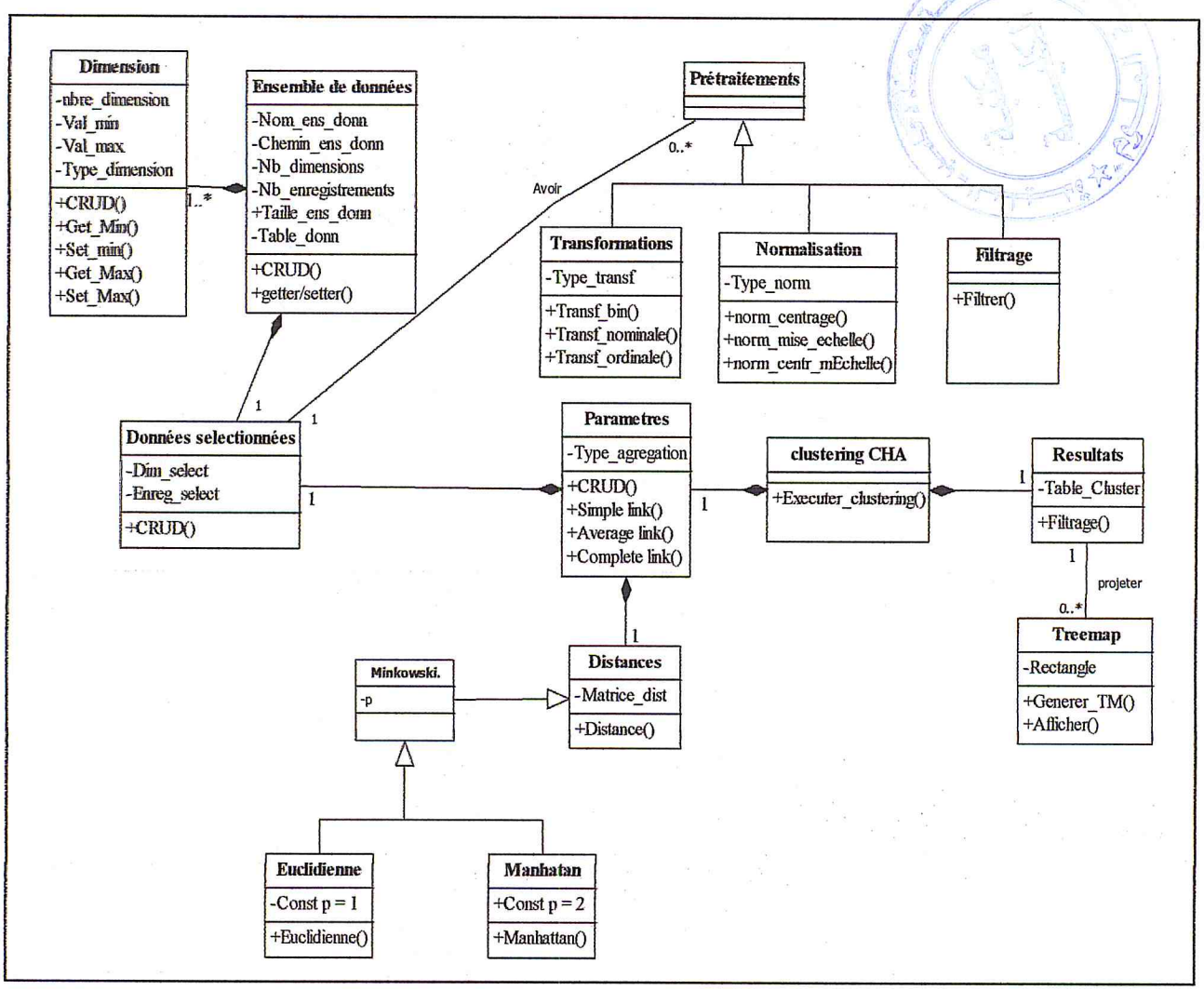


Figure III.16. Digramme de classes.

III.9.1.a.Description du diagramme de classes

Nom de la classe	Le rôle de la classe	Les attributs de la classe
Ensemble de données	Cette classe représente les caractéristiques du l'ensemble de données importée.	Nom_ens_donn Chemin_ens_donn Nb_dimensions Nb_enregistrement Taille_ens_donn Table_donn.
Données sélectionnées	Cette classe représente le sous ensemble de données sélectionné. une copie est crée lors de l'importation de l'ensemble de données « Données sélectionnées » est l'unité élémentaire de notre système. C'est sur les données sélectionnées que vont se faire les traitements et le clustering.	Dim_selec Enreg_selec
Dimension	C'est la classe qui contient les caractéristiques sur les dimensions constituant l'ensemble de données initial.	Nom_dim Type_dim Val_Min Val_min Nbre_dim
Prétraitements	C'est la classe mère de Transformations, Normalisations et filtrage. Cette classe représente les différents traitements (calcul)	

	qui peuvent être effectués sur l'ensemble de données sélectionnées.	
Transformations	Cette classe permet de standardiser les données (toutes les données sont du même type).	Type_transformation
Normalisation	Cette classe permet de normaliser les données (les rendre de la même échelle) Cette classe contient 3 types de normalisation (par centrage, par mise à l'échelle, par centrage et mise à l'échelle).	Type_norm
Paramètres	Cette classe contient le paramétrage de clustering (les stratégies d'agrégation)	Type_agregation
Clustering CHA	Cette classe récupère tous les paramètres de clustering CHA.Le paramétrage de distance et le type d'agrégation choisi et exécute l'algorithme CHA à l'aide de l'opération Exécuter-clustering ().	
Résultats	Cette classe vient représenter les résultats du clustering par l'attribut Table_clusters. C'est par le biais de l'opération filtrage () de cette classe que l'utilisateur pourra sélectionner les clusters qu'il voudra visualiser.	Table_clusters

Treemap	Cette classe dessine le graphique de Treemap (La technique de visualisation).	Rectangle.
Distance	C'est la classe mère de la distance Minkowski. Elle regroupe dons tous les types de distances.	Matrice des distances
Minkowski	C'est la classe mère de Manhatan et Euclidienne. Elle regroupe donc toutes les caracteristiques de Manhatan et Euclidienne.	P
Manhattan	Cette classe contient tous les propriétés de Manhattan.	Const P = 1
Euclidienne	Cette classe contient tous les propriétés Euclidienne.	Const P = 2

Tableau III.9. Description du diagramme de classes.

III.10.Conclusion :

Dans ce chapitre, nous avons présenté les étapes détaillées que nous avons suivi pour la construction de notre outil de visualisation « VisionForce».L'étude conceptuelle nous a permis de mettre en évidence les étapes nécessaires pour la création de l'outil visuel. Cette étude nous a permis aussi de mettre en évidence les différentes classes du système.

Dans le chapitre suivant, nous allons implémenter et mettre en œuvre ce que nous avons proposé dans l'étude conceptuelle, en d'autres termes, l'implémentation de notre approche.

CHAPITRE IV

Implémentation et tests

IV.1.Introduction

Après avoir effectué la conception, nous allons à présent entamer la réalisation de VisionForce. Nous présenterons alors, dans un premier lieu, l'environnement de développement (matériels, langages et outils) ensuite, quelques captures d'écran avec les différents tests effectués.

IV.2.Environnement de développement

IV.2.1.Materiel utilisé

Nous avons développé notre application sur des machines dotées du système d'exploitation Windows 7 et ayant des processeurs Intel Core i3.

La fréquence d'horloge du CPU est de 2.3 GHz et la mémoire installée (RAM) est de 4.00 Go.

IV.2.2.Langages utilisés

IV.2.2.a. Java

Java est un langage de programmation informatique orienté objet crée par James Gosling Patrick Naughton de Sun Microsystèmes.

Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris).

Le langage Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (JSP) [58]

IV.2.3.Outils

IV.2.3.a.Eclipse

Eclipse **IDE** est un environnement de développement intégré libre, le terme *Eclipse* désigne également le projet correspondant, lancé par IBM. [59]

Il est extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. [60]

Eclipse IDE est principalement écrit en Java à l'aide de la bibliothèque graphique SWT, IBM. [49]

IV.3. Présentation de VisionForce:

Lors de l'exécution de notre application, une interface simple est affichée, contenant principalement les étapes qu'offre VisionForce.

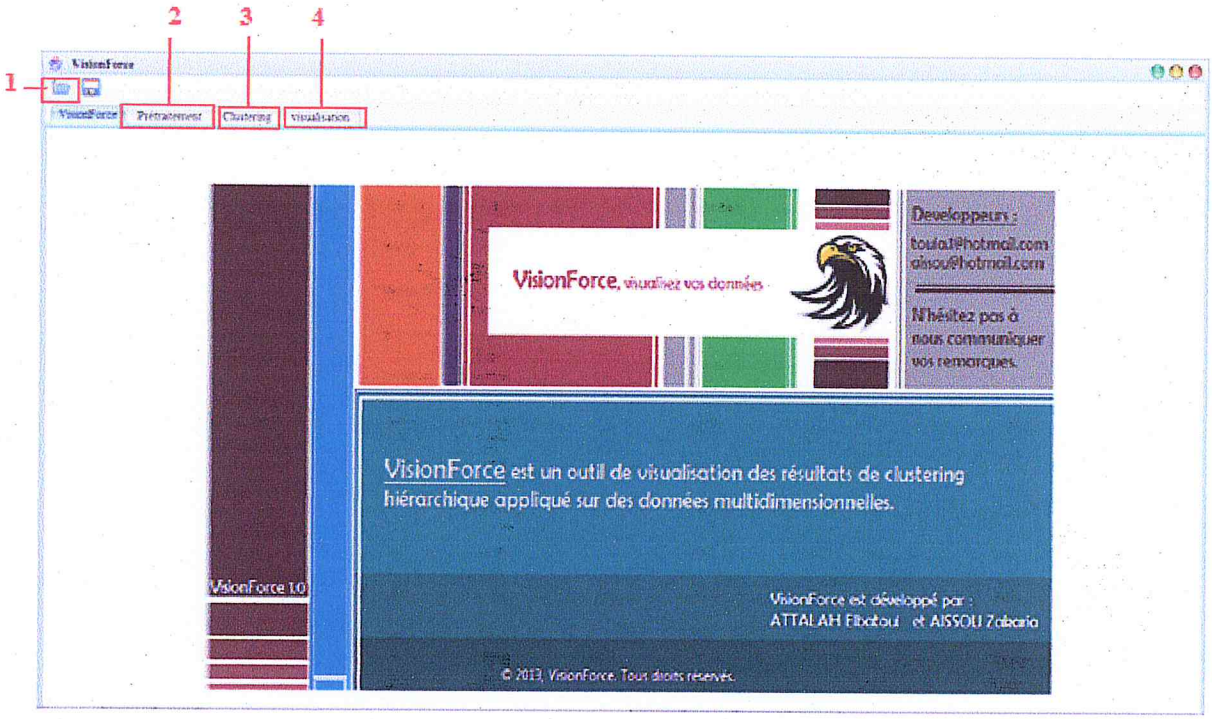


Figure IV.1. Interface principale de VisionForce.

L'interface d'accueil de VisionForce se base sur le processus de clustering visuel :

1. Prétraitement, 2.Clustering, 3.Visualisation et exploration.

Nous détaillerons les fonctionnalités de chaque interface.

L'utilisateur doit indiquer le type de fichier à importer (1) à partir d'un sous un menu comme ceci :

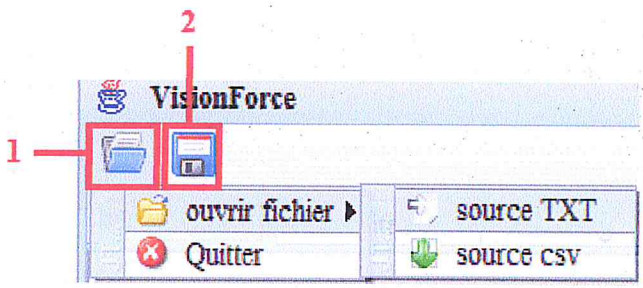


Figure IV.2. Specification de type de la source à importer.

Une fois choisi et importé, l'utilisateur peut enregistrer en cliquant sur (2) l'ensemble de données importé sous le nom save.txt.

IV.3.2.Prétraitements

L'utilisateur clique sur le bouton « 1 » pour accéder à l'interface de prétraitement. Une fois accéder, l'utilisateur peut charger «bouton 2 » les informations relatives à l'ensemble de données importé (path. taille de fichier, nombre d'enregistrement, nombre de colonnes)
Le bouton « 3 »permet d'afficher les données afin d'effectuer les traitements nécessaire :

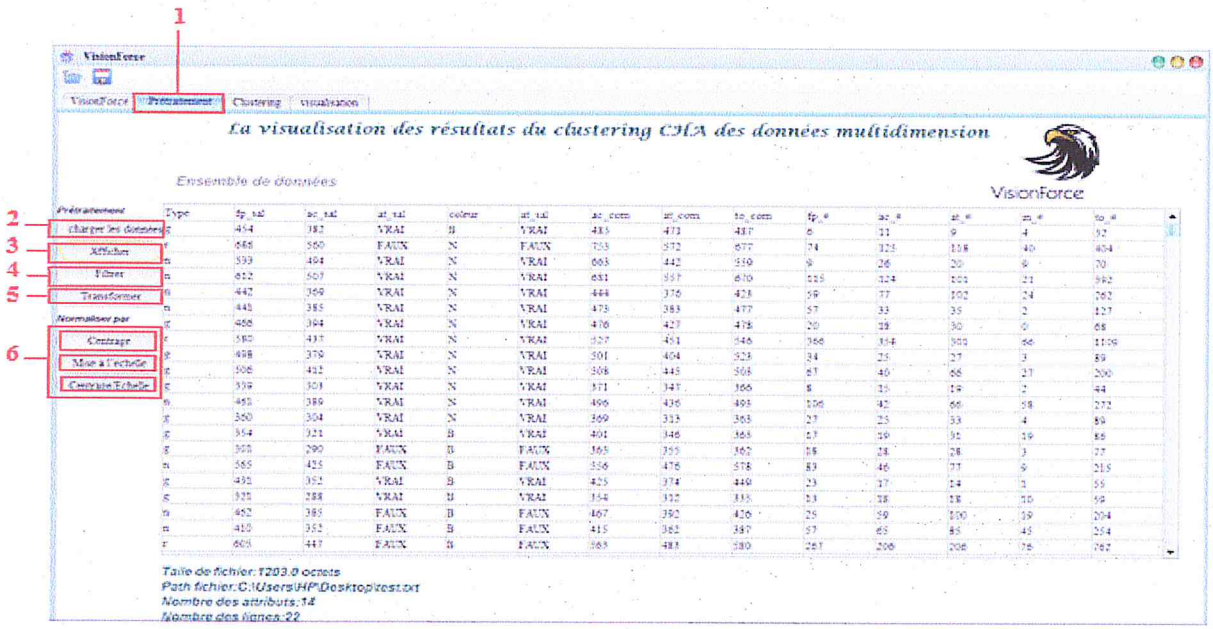


Figure IV.3. Interface de prétraitements.

L'utilisateur peut éliminer les dimensions inutiles en sélectionnant la dimension puis un clic sur le bouton « 4 » afin d'obtenir un sous ensemble de données. C'est une étape optionnelle.

L'utilisateur sélectionne la dimension à transformer puis le bouton « 5 » dans le but de standardiser ces données(les rendre de même type), une interface de transformation est affichée contenant le nom de la dimension à transformer « 7 » , le type de la dimension « 8 » qui permet à l'utilisateur de choisir le type de la dimension « 9 » puis le bouton« 10 » pour valider la transformation.(Figure IV.4)

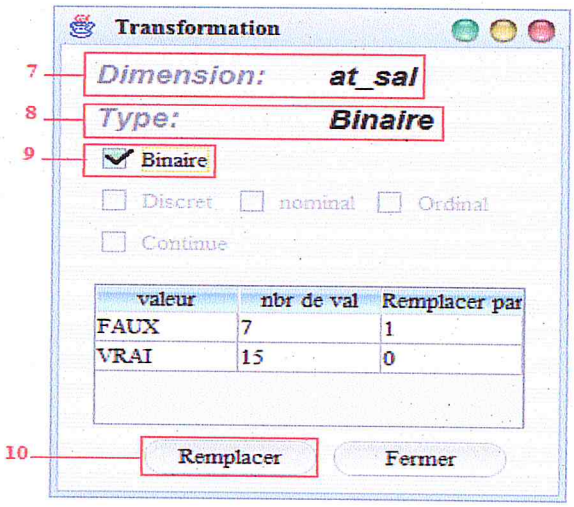


Figure IV.4. Interface de transformation.

En choisissant un type parmi les trois types de normalisation « 6 » pour normaliser le sous ensemble de données sélectionnée. « C'est une étape optionnelle »

IV.3.3.Clustering

Après le prétraitement de données l'utilisateur doit suivre le processus en cliquant sur le bouton « 1 » pour spécifier les paramètres de clustering en choisissant une distance « 2» est une stratégie d'agrégation « 3 »afin de regrouper les individus similaires. Après la spécification des paramètres, l'utilisateur clique sur le bouton « 4 » pour afficher la matrice de ressemblance, et « 5 » pour afficher la table de clusters. Comme suit :

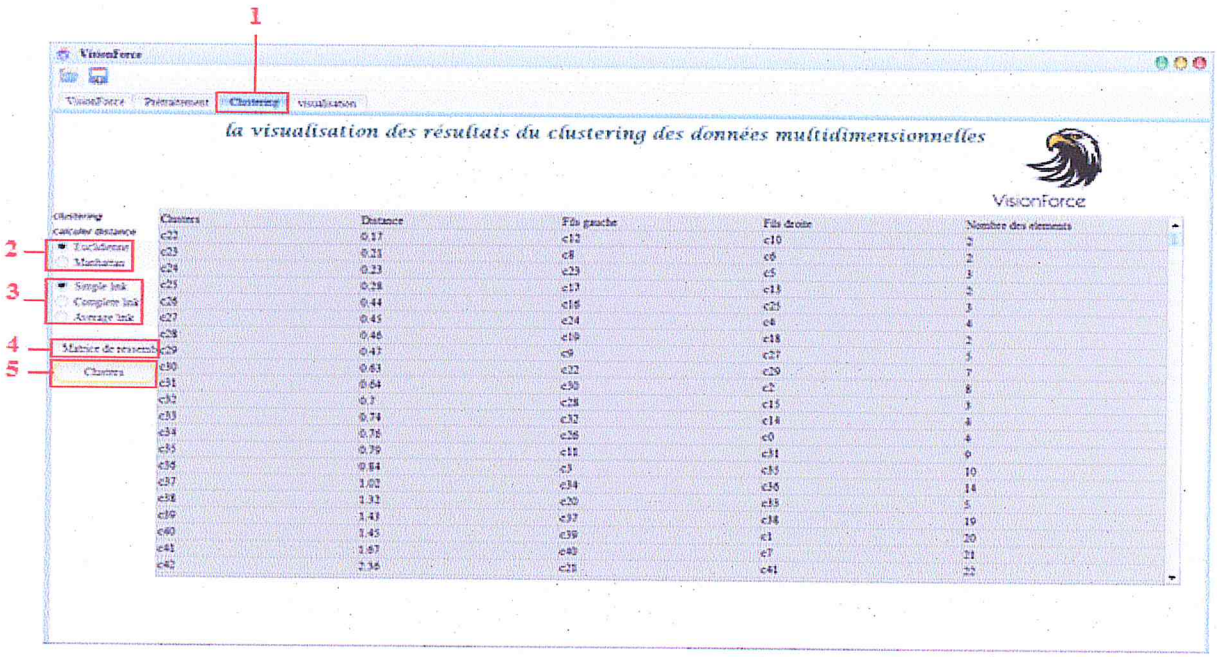


Figure IV.5. Interface de l'arbre de clusters.

IV.3.4.Visualisation

Après le regroupement des individus avec le clustering hiérarchique ascendant, l'utilisateur doit sélectionner le bouton « 1 » pour accéder à l'interface de visualisation, l'interface de visualisation contient quatre boutons :

- Le bouton «2 »permet d'afficher le dendrogramme.
- Le bouton « 3 »permet de visualiser le Treemap « une vue globale »
- Le bouton « 4» permet de spécifier les clusters à visualiser afin d'explorer les résultats « des détails à la demande ».
- Le bouton « 5» permet d'afficher le Treemap sans couleur.

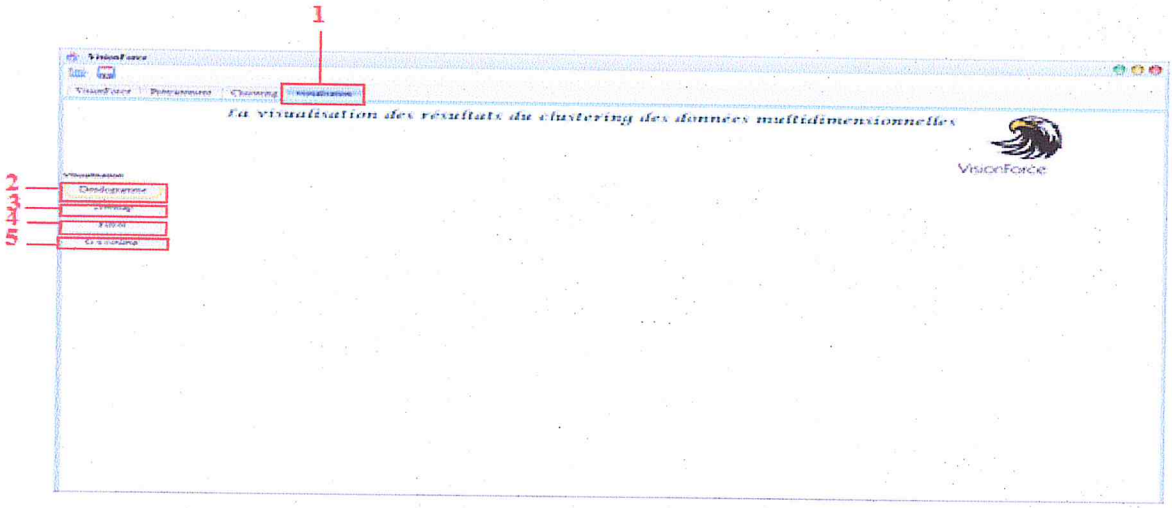


Figure IV.6. Interface de visualisation

Pour pouvoir confirmer le bon fonctionnement de VisionForce , il est impératif d'établir des tests en utilisant des ensembles de données multidimensionnelles en entrée. Dans ce mémoire, nous présenterons notre ensemble de données ainsi que les étapes nécessaires avant la génération de Treemap final, nous montrerons aussi les différentes vues engendrées par des paramétrages distincts sur le même ensemble de données.

IV.4.Tests et résultats

IV.4.1. Présentation de l'ensemble de données :

Pour effectuer nos tests, nous avons utilisé l'ensemble de données des opinions des personnes concernant les films d'horreur ainsi que leurs préférences. Cet ensemble de données contient 10 dimensions et 12 personnes.

Le tableau suivant décrit l'ensemble de dimensions constituant l'ensemble de données à tester :

Dimension	Description	Type et valeur
regader	Avez-vous déjà regardé des films d'horreur ?	cette dimension contient deux valeurs oui ou non.
Combien	Le temps passé à regarder des films d'horreur dans le mois précédant l'enquête, en heure.	Contient des valeurs réelles.
Aimer	Aimez-vous les films d'horreur ?	peut contenir des valeurs nominales discrètes : oui, non ou moyenne.
Lieu	Lieu ou vous regardez les films d'horreur.	contient des valeurs nominales continues : • dom (domicile) : indique que l'individu regarde les films d'horreur à la maison. • Cinéma : indique que l'individu regarde les films dans le cinéma. • Ord : indique que l'individu regarde les films d'horreur sur son ordinateur. .
Fréquence	A quelle fréquence vous regardez les films d'horreur.	contient les valeurs nominales discrètes suivantes : • Semestr : indique que l'individu regarde les films chaque semestre. • Mensu : indique que l'individu regarde les films chaque mois. • Hebdo : indique que l'individu regarde les films d'horreur à une fréquence hebdomadaire. • Quoti : indique que l'individu regarde les films d'horreur quotidiennement.
Occupé	Vous regardez des films d'horreur lorsque vous êtes occupés.	contient des valeurs nominales au nombre de deux : oui ou non.
Sous_titre	Aimez-vous les films	contient des valeurs nominales au

	d'horreur ?	nombre deux: oui ou non.
Age	Quel âge avez-vous ?	est une dimension contenant des valeurs numériques continues.
Classe	A quelle classe vous classez les films d'horreur ?	est une dimension qui contient des valeurs nominales ordinale représentant la classe des films d'horreur pour chaque individu (A, B, C, D, etc.).

Tableau IV.1. Description de l'ensemble de données utilisé.

Après l'importation de ce derniers l'utilisateur peut accéder à l'interface de prétraitement après l'affichage l'ensemble de données afin d'effectuer les prétraitements nécessaires « c'est une étape optionnelle». La figure IV.7 présente l'ensemble de données de fichier à traiter.

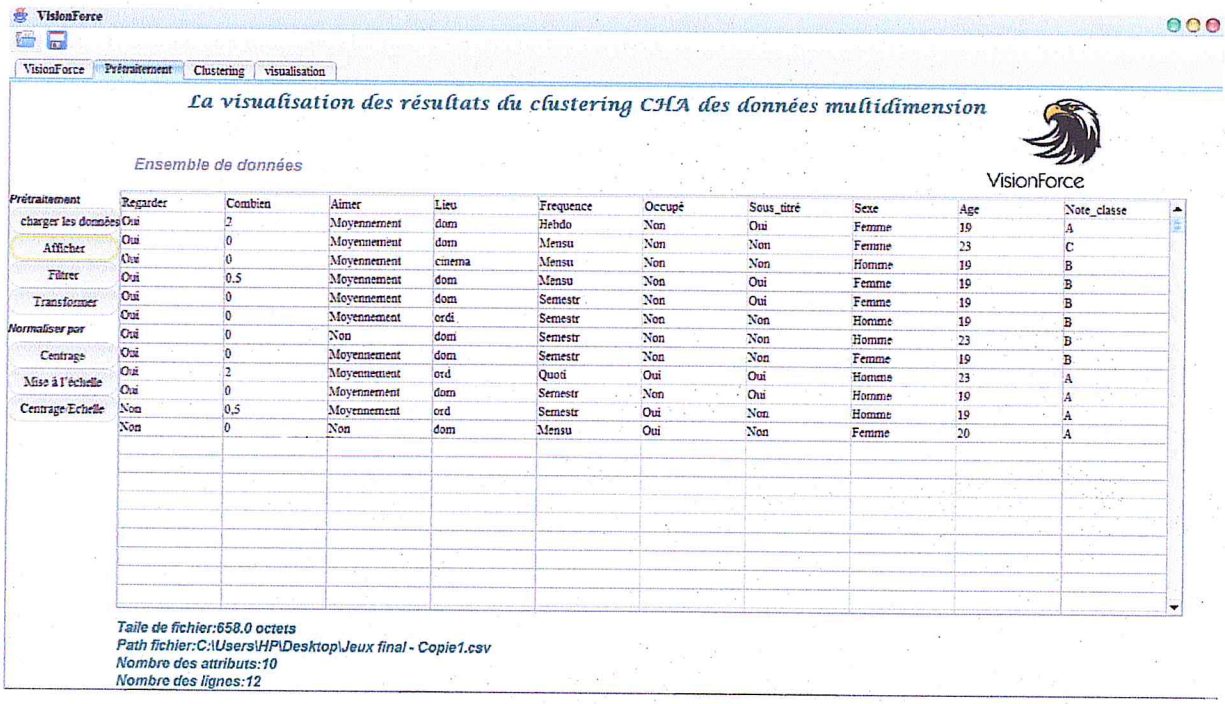


Figure IV.7. L'ensemble de données utilisé.

On commence le processus de prétraitement par le filtrage de dimensions inutiles, dans notre cas nous filtrons la dimension « Note_classe » pour obtenir le sous ensemble de données à traiter.

Dans le cas où les dimensions ne sont pas du même type L'utilisateur doit transformer ces données en sélectionnant la dimension à transformer puis le bouton « transformer », une interface de transformation est apparu comme c'est indiquer dans la figure .IV.8 dans le but de standardiser ces dernières.

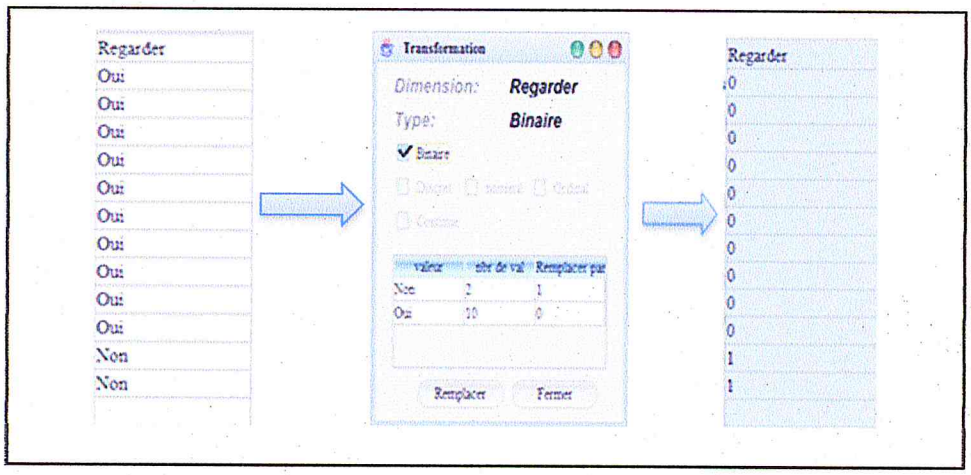


Figure IV.8. Processus de transformation de l'attribut 'Regarder'.

La figure IV.9 montre l'ensemble de données choisi après les transformations jugées nécessaires à effectuer pour les étapes suivantes :

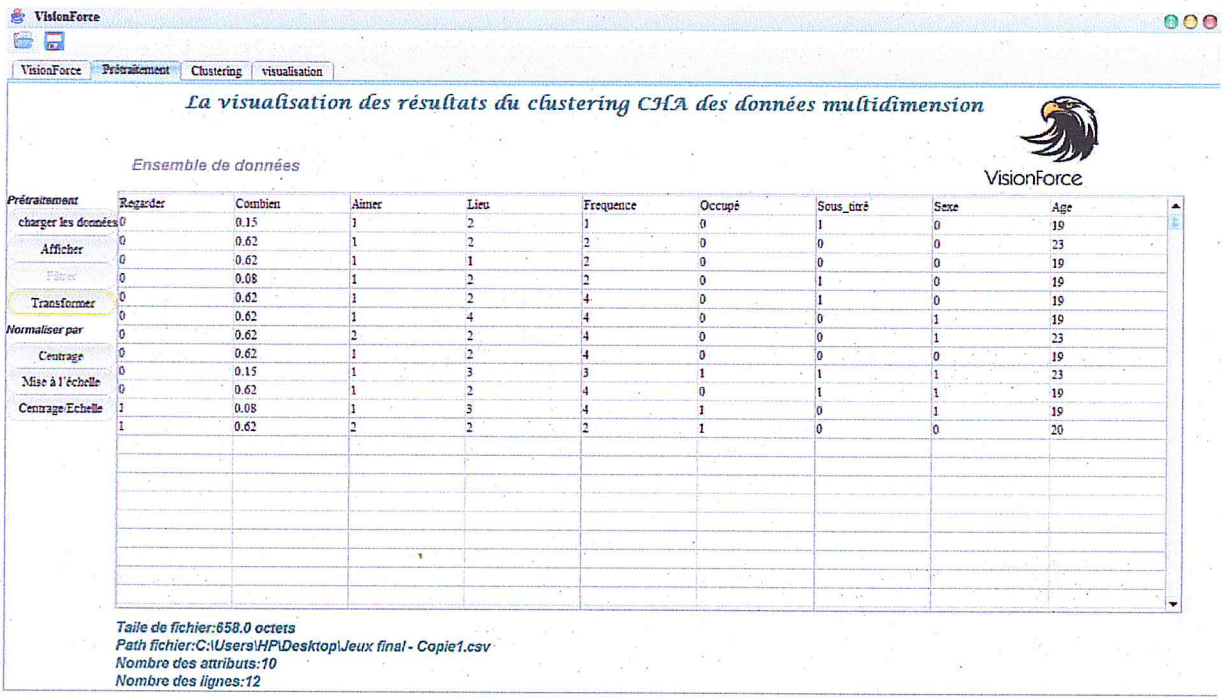


Figure IV.9. La table de données standardisées.

Après la transformation et pour avoir un meilleur regroupement on passe par l'étape de normalisation pour réduire l'échelle de grandeur de variables. Choisissant une parmi les trois existants :

- Normalisation par centrage : [62]

Le plus simple moyen de normaliser les données est d'extraire de chaque attribut sa moyenne

$$X_{ij} = X^*_{ij} - m_{ij}$$

- Normalisation par mise à l'échelle : [63]

Une deuxième méthode de normalisation consiste à redimensionner les données dans l'intervalle [0 ,1], par la transformation linéaires suivantes

$$X_{*j} = \frac{X^*_{*j}}{X^*_{*j\,max} - X^*_{*j\,min}}$$

tel que X_{*j} et X^*_{*j} représentent les vecteurs d'attributs de l'ensemble X respectivement $x^*_{*j\,max}$ Et $X^*_{*j\,min}$ représentent les valeurs maximale et minimale observées de l'attribut x^*_{*j}.

- Normalisation par centrage et mise à échelle : [62]

Le troisième type de normalisation revient à translater les données et à les redimensionner de telle sorte qu'elles soient de moyenne nulle et de variance unitaire [62]. La loi de normalisation est donnée par :

$$x_{ij} = \frac{x^*_{ij} - m_j}{s_j}$$

La normalisation par mise à l'échelle est la meilleure pour cet ensemble des données car elle garde la structure des attributs du type binaire.

La figure(IV.10) montre la table des données normalisées :

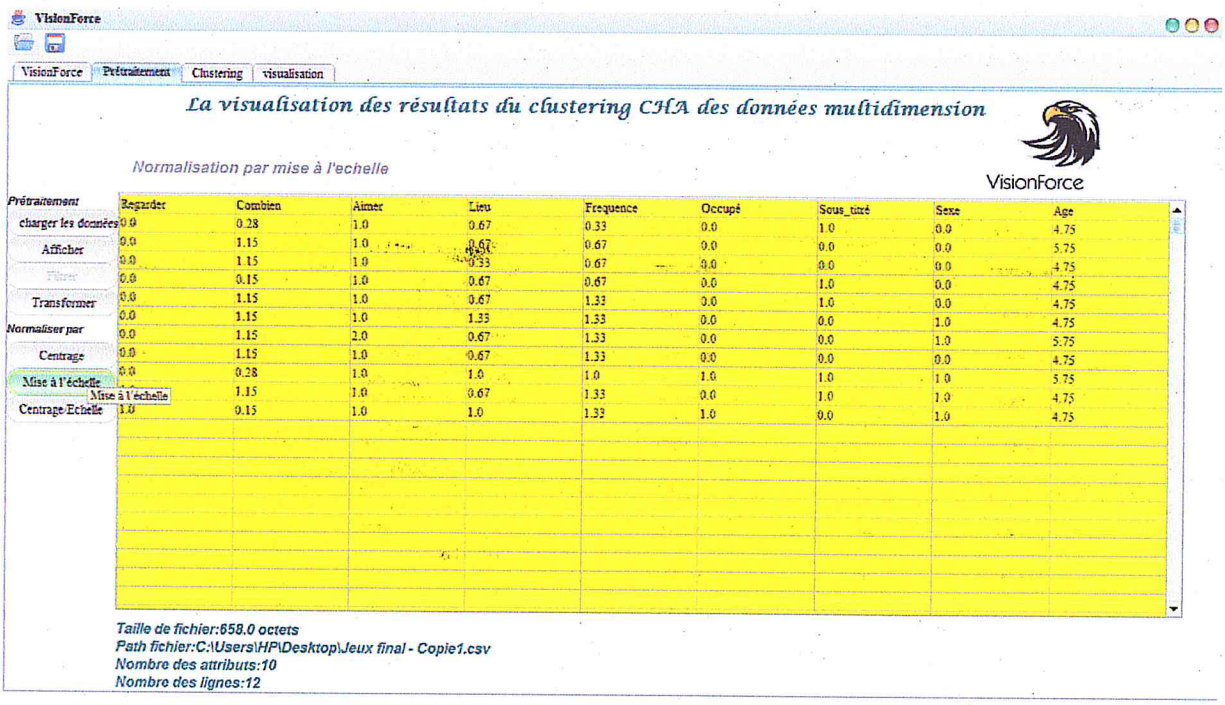


Figure IV.10. La table de données normalisée.

L'utilisateur doit suivre le processus de clustering visuel par la spécification des paramètres de clustering nécessaire tel que la distance et la stratégie d'agrégation.

Dans notre cas, Nous effectuons nos tests sur le même ensemble de données avec un paramétrage du clustering différent Afin de visualiser l'impact des mesures de distances et les stratégies d'agrégations sur les résultats de clustering, nous verrons des résultats différents et nous les interpréterons par la suite, Nous effectuerons les deux tests suivants :

- Dans le 1er test nous fixons la distance Euclidienne et la stratégie simple link.

(Figure IV.11)

- Dans le 2eme nous fixons la distance Manhattan et la stratégie complete link.

(Figure IV.12).

Après la spécification des paramètres de clustering l'utilisateur peut visualiser la matrice initiale de ressemblance et la table des clusters.

Nous présentons la matrice des deux tests effectuées :

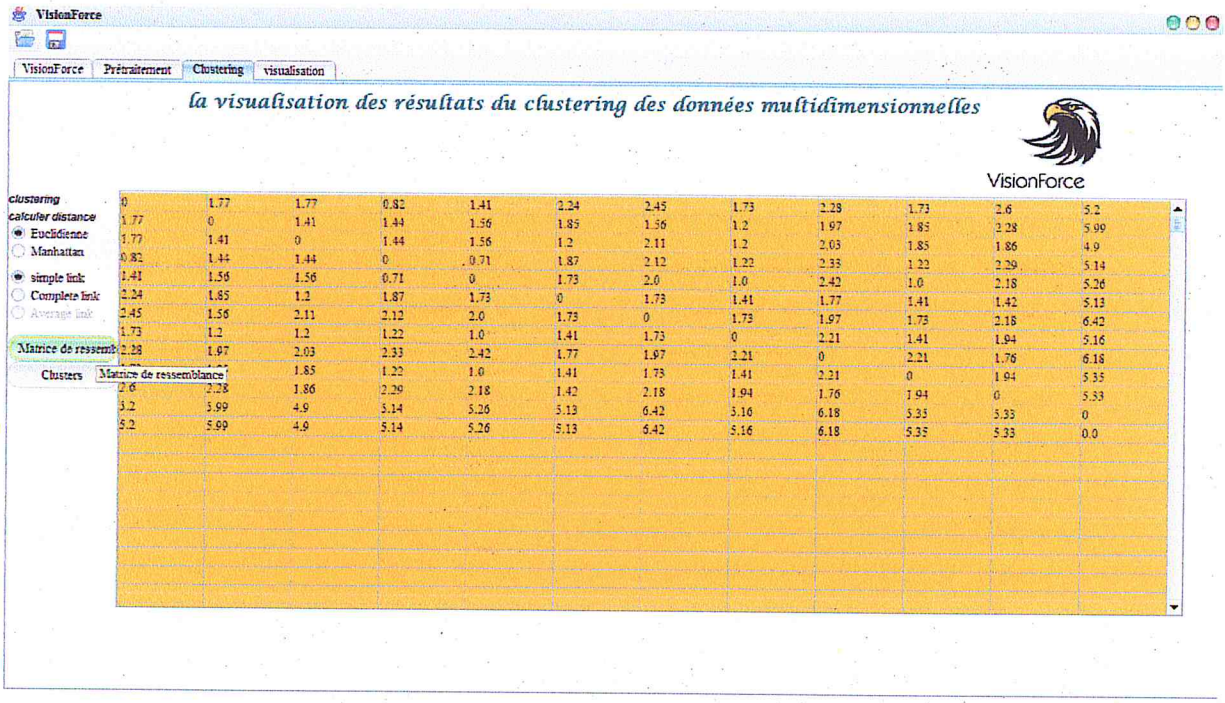


Figure IV.11. Matrice de ressemblance initiale (1er test)

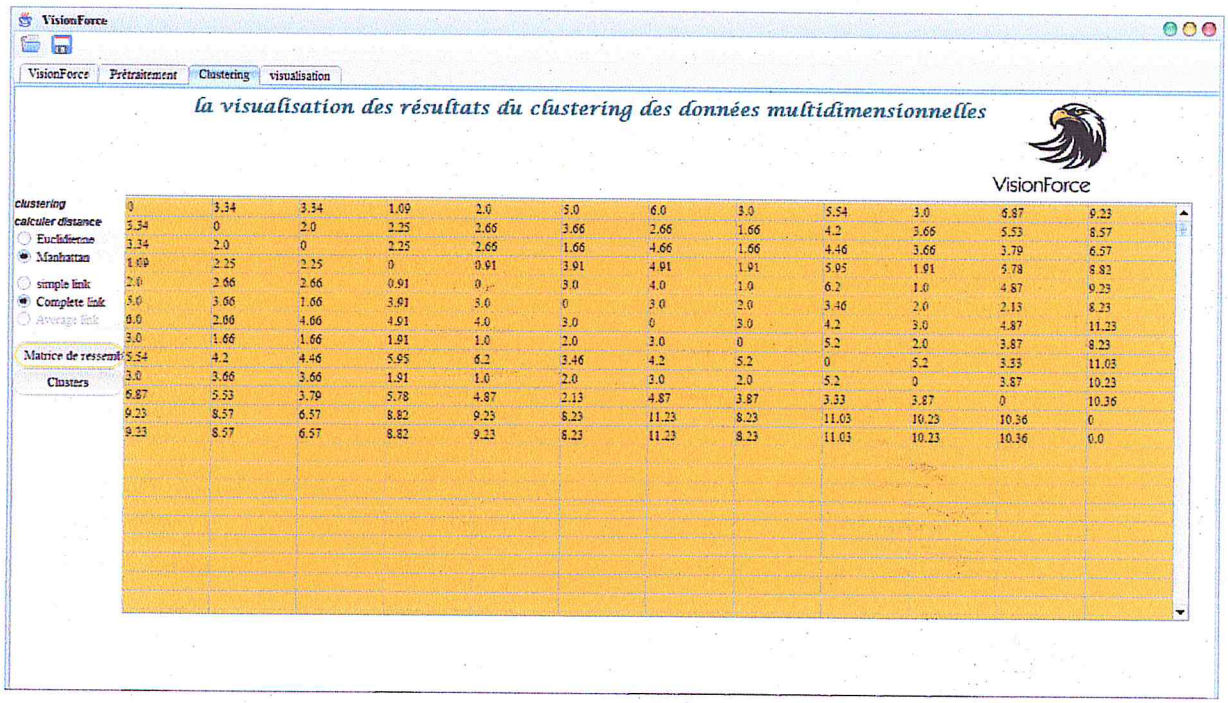
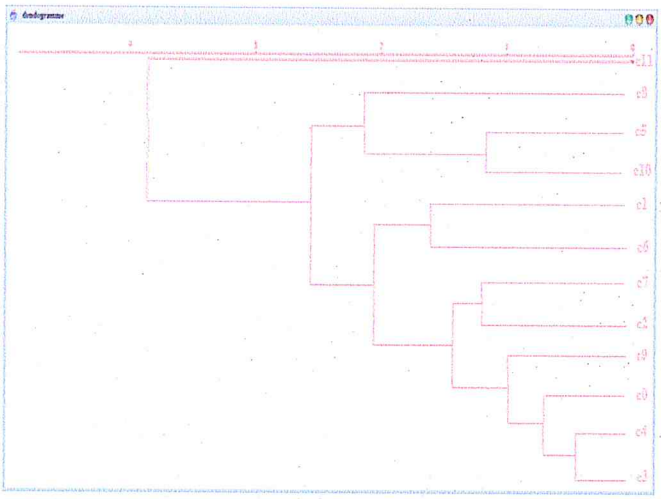
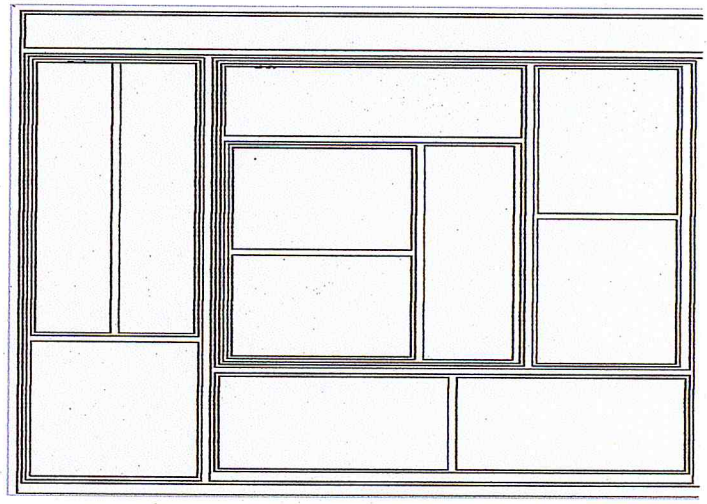


Figure IV.12. Matrice de ressemblance initiale (2eme test)

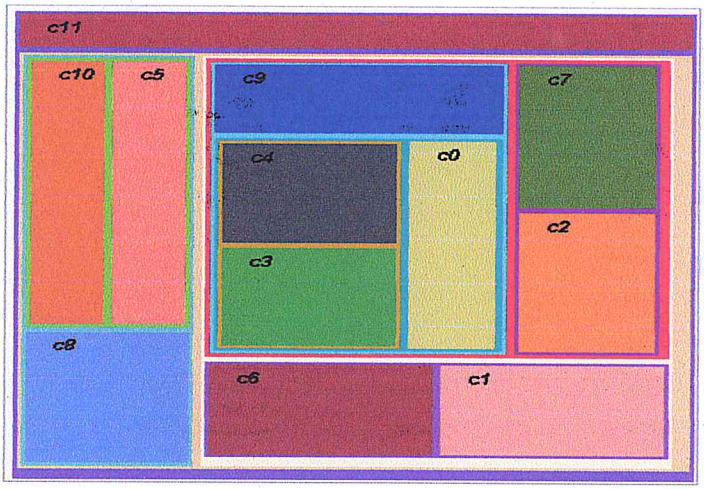
Après le regroupement des individus avec le clustering hiérarchique ascendant, L'utilisateur peut apercevoir la hiérarchie des clusters en passant à l'interface de visualisation pour visualiser les résultats de clustering en utilisant le dendrogramme et le Treemap des deux tests afin de les analyser.



(a)Dendrogramme (1er test)

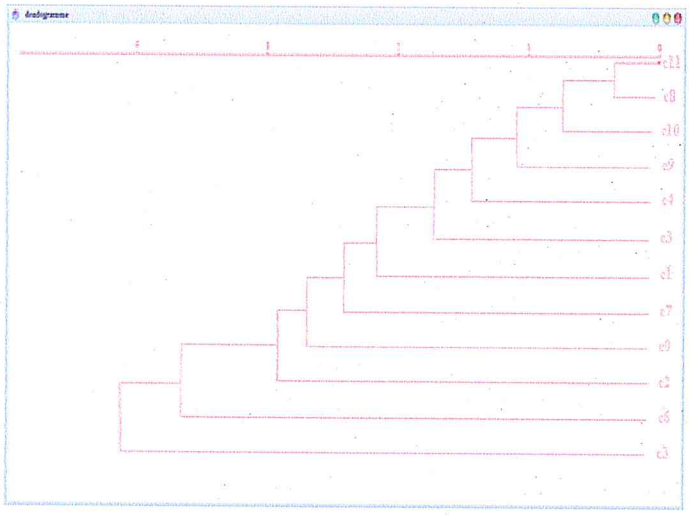


(b)Treemap sans couleur (1er test)

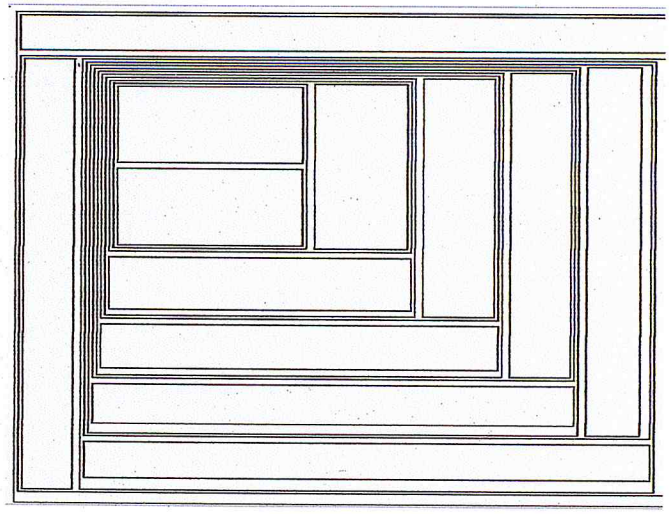


(c)Treemap avec couleur et détails (1er test).

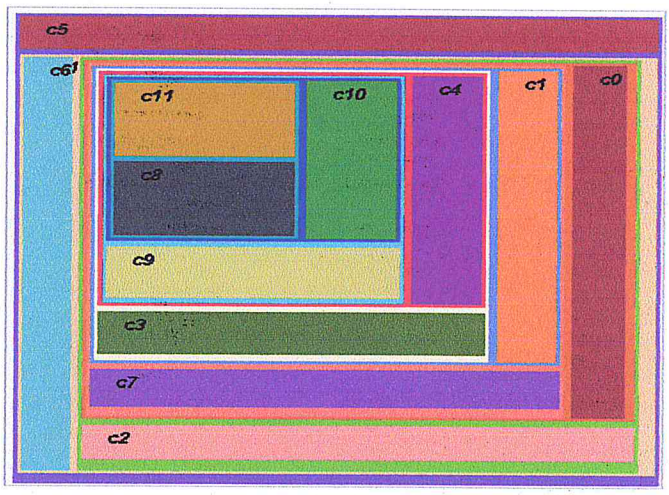
Figure IV.13. Visualisation des résultats du clustering du 1er test utilisant le Treemap et le dendrogramme.



(a)Dendrogramme (2eme test)



(b)Treemap sans couleur (2eme test)



(c)Treemap avec couleur et détails (2eme test)

Figure IV.14. Visualisation des résultats du clustering du 2eme test utilisant le Treemap et le dendrogramme.

Treemap contient une caractéristique importante car il utilise un espace d'affichage très important, il est possible d'afficher de grands arbres avec de nombreux niveaux hiérarchiques dans un minimum d'espace (2D).Treemap peut être particulièrement utile lorsqu'il s'agit de grands arbres en cluster. Arborescences se prêtent naturellement à montrer les informations encapsulées dans l'arbre de classification.

La visualisation d'un arbre à un certain niveau d'abstraction, l'utilisateur peut définir le niveau de découpage pour une vue précise. Pour plus de détails, VisionForce fournit à l'utilisateur le détail de chaque cluster sous forme d'un tableau contenant les individus de ces derniers et le pourcentage des valeurs de chaque dimension avec ses statistiques pour une meilleure analyse.

L'interface de la visualisation offre la possibilité à l'utilisateur d'interagir avec l'application afin d'explorer les résultats de clustering hiérarchique. En choisissant le nombre de clusters à visualiser (Figure IV.15).

- (a)montre le Treemap après le filtrage pour 1er test.
- (b) montre le Treemap après le filtrage pour 2eme test.

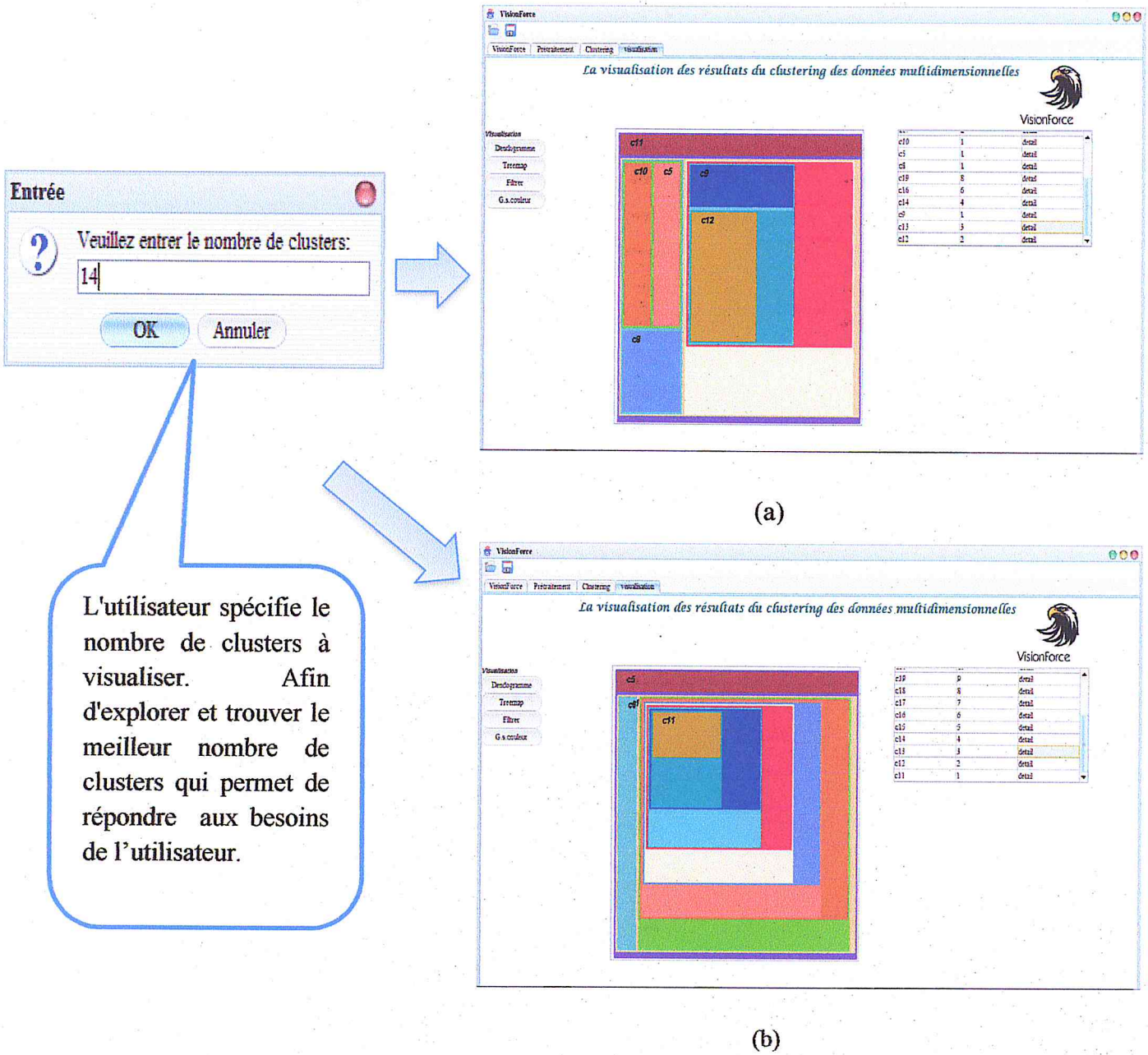
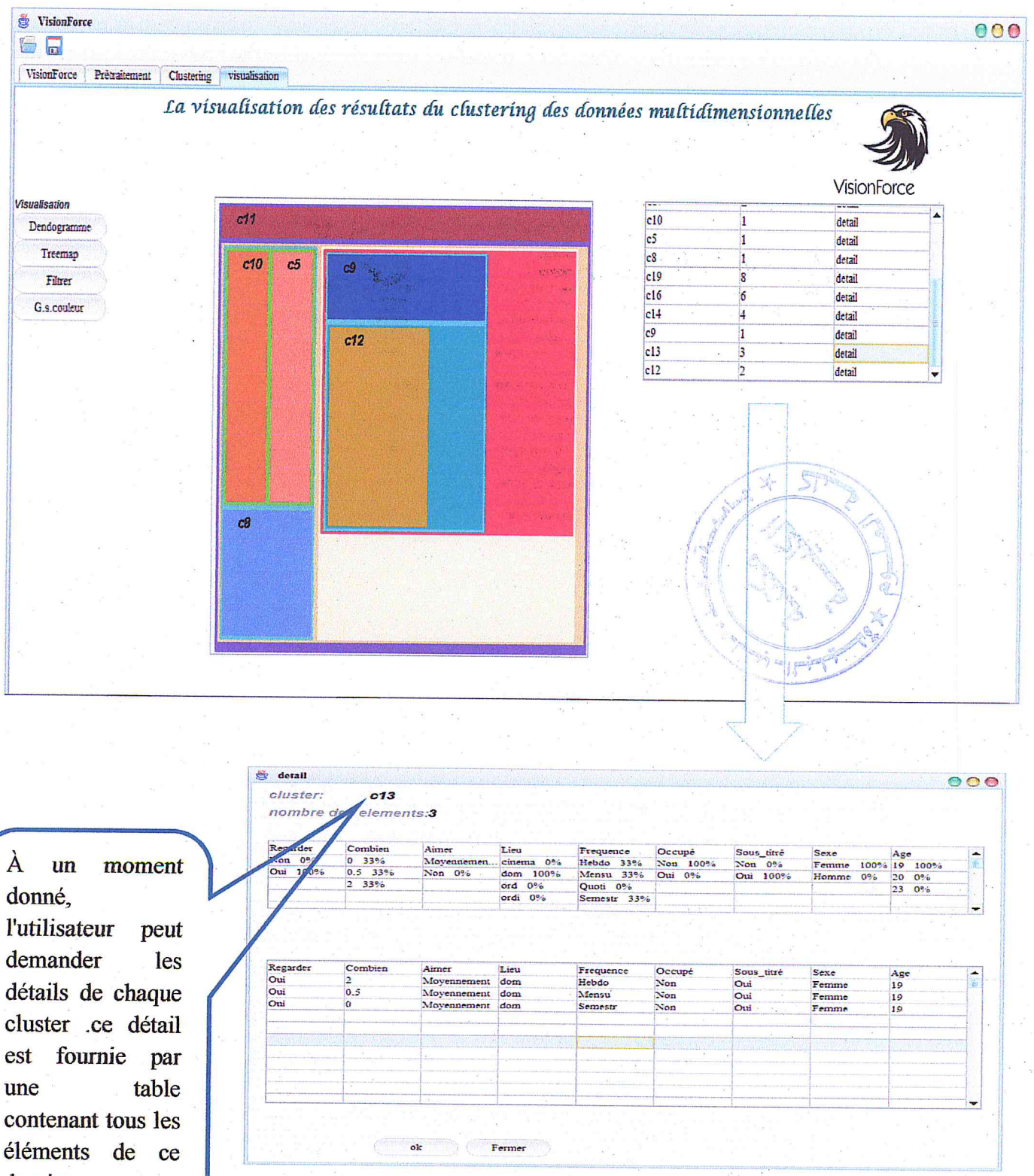


Figure IV.15.la matrice de ressemblance et son dendrogramme pour chaque distance appliquée sur notre cas d'étude.



À un moment donné, l'utilisateur peut demander les détails de chaque cluster .ce détail est fournie par une table contenant tous les éléments de ce dernier.

Figure IV.16.Détails de cluster sélectionné (c13) montrant ses individus pour le 1er test.

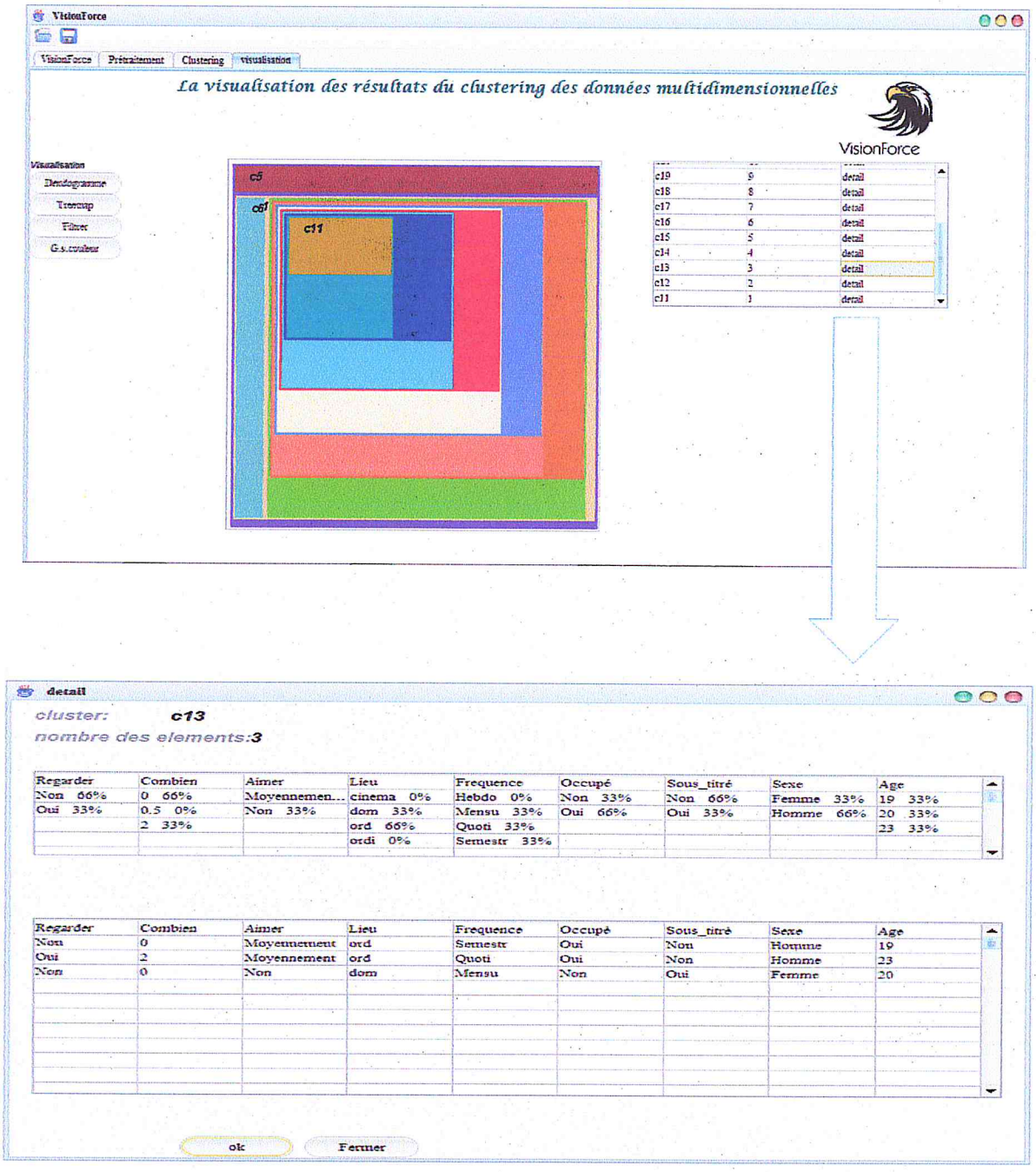


Figure IV.17.Détails de cluster sélectionné (c13) montrant ses individus pour le 2eme test.

A ce stade, nous avons démontré le bon fonctionnement du paramétrage du clustering ainsi que la visualisation et les interactions disponibles sur VisionForce.

Dans ce travail, nous sommes intéressés par l'utilisation de nouvelle technique de visualisation hiérarchique "Treemap" afin de présenter la structure hiérarchique des résultats de clustering CHA. Pour une meilleur exploration de ces derniers, nous avons fourni un niveau de filtrage qui conduit à une vue précise on spécifiant le nombre de clusters à visualiser et à l'utilisateur d'explorer les détails sur les clusters pour détecter le meilleur regroupement et aussi les caractéristiques intéressantes de ces données.

IV.5.Conclusion :

Dans ce chapitre nous avons présenté l'environnement de développement, les interfaces et quelques tests sur notre système de visualisation VisionForce. Cette étape nous a par ailleurs, permis de nous familiariser avec les différentes fonctionnalités qu'offre notre système.

Conclusion Générale

Sommaire

Conclusion générale

Les travaux menés dans ce mémoire nous ont permis d'approfondir nos connaissances dans le domaine de visualisation et de clustering, notre objectif a été tirer profit des travaux menés dans cette voie et nous nous sommes intéressés dans notre application plus particulièrement aux prétraitements possibles sur les données sources afin de les standardiser pour les différents calculs nécessaires puis le regroupement des individus en utilisant l'algorithme CHA, ensuite la visualisation de ces derniers en utilisant la technique de treemap qui permet à l'analyste d'explorer les résultats et sortir avec des hypothèses.

Nos recherches et constatations, nous ont conduits à la mise en place d'un outil graphique «VisionForce ».L'originalité de notre application réside dans la combinaison des caractéristiques suivantes :

- Robustesse : Le système VisionForce peut être appliqué sur n'importe quel ensemble de données contenant les extensions (.csv et .txt).
- Ergonomie : Son interface très malléable, rend VisionForce très agréable d'utilisation. Lui offrant la possibilité de charger un ensemble de données, de prétraiter ses données ,regrouper et aussi les visualiser avec possibilité d'interagir et d'explorer ce dernier ou de le sauvegarde.
- Les mesures de similarités et les stratégies d'agrégation qu'adopte VisionForce lui confère un haut niveau de pertinences dans le calcul des correspondances entre les individus. Ce qui permet de produire un regroupement très fidèle.

En résumé, on dira que l'objectif initial qu'on a fixé est conçu sortant avec quelques prescriptives.

Perspectives

Durant les mois consacrés à la réalisation de notre projet de fin d'études nous nous sommes tenues d'atteindre l'objectif qui nous a été fixés au départ.

Toutefois, dans le souci d'améliorer « VisionForce » et d'étoffer le spectre de ses fonctionnalités, nous proposons les perspectives suivantes :

- Ajouter d'autres algorithmes de clustering. pour permettre à l'utilisateur de sélectionner l'algorithme de clustering voulu pour assurer le résultat.
- En Plus du treemap, ajouté d'autres techniques de visualisation pour la cartographie visuelle des données.
- Intégrer d'autres sources de données par exemple Base de données SQL.

Annexe

1. Présentation de java : [61]

Java est un langage de programmation informatique compilé et interprété, orienté objet. Il a été créé en 1991 par Sun pour pallier aux contraintes que posait le C++,et cela dans le but de développer des logiciels pour l'électronique grand public(appareils ménagers).la syntaxe de ce langage est assez proche du C et est plus claire que le C++,l'objectif pour lequel JAVA a été conçu nécessite certaines caractéristiques comme :la robustesse, la comptabilité, la facilite de programmation.JAVA présente beaucoup d'avantages grâce auxquels il a mérité son succès :

- **Orienté objet :**

La brique de base du programme est donc l'objet, instance d'une classe.

- Indépendant des architectures
- **Portable :** une fois le programme est compilé, il pourra fonctionner aussi bien sous des stations Unix, que sous Windows ou autre.
- **Interprété :** par la virtuelle machine de JAVA (JVM).
- Gere la mémoire automatiquement.
- Possède une API très riche : différentes packages permettent d'accéder au réseau, aux entrées /sorties et aux différents composants graphiques.
- Distribué, simple, robuste, sécurisé, multithread et dynamique.

Sa portabilité et son indépendance des architectures ou plates-formes et grâce auquel il peut fonctionner sur différentes plates-formes et sous différentes systèmes d'exploitation. Ces deux caractéristiques sont dues à la machine virtuelle Java(JVM).a cote de ses nombreux avantages, JAVA présente un seul inconvénient : la lenteur lors de la conversion des instructions de la JVM en instructions compréhensibles par la machine. Cependant deux solutions sont mises en place pour éviter cet inconvénients :

- Compiler le code pour une machine spécifique, à ce moment-là le programme n'est plus portable.
- Doter la JVM d'un JIT (Just In Time compiler) qui traduit le code JVM d'une classe dès sa première utilisation en code spécifique à la machine.

2. JAVA virtuelle machine(JVM) : [59]

La machine virtuelle appelée aussi interpréteur, joue le rôle d'un traducteur. Elle Traduit en ByteCode le code compilé (ensemble de fichier de classe) qui est sous forme de pseudo code et non en code machine, et cela afin d'être exécuté sur n'importe quel plate-forme matérielle et logicielle :que l'on soit sur un pentium, un PowerPC, un Sparc ou sur un alpha, sous Windows ,MacOs, Solaris ou Linux, etc. Cependant, la présence de la JVM au niveau de la plate-forme est nécessaire à l'exécution des programmes JAVA sur une plate-forme quelconque.

Les JVMs sont fournies soit par le JDK (Java Développement Kit), soit par les navigateurs ou bien par les environnements de développement spécifiques tel Borland JBuilder. Des JVM capable d'exécuter du code Java, peuvent être fournit également par certaines solutions telle Java Plug-in de Sun la JVM peut être obtenue à partir du site de Sun Microsystèmes.

3. Environnement de développement :

Il existe plusieurs environnements de développement pour JAVA tel le JDK (Java Développement Kit) qui est gratuit, mais la plus part sont payants. Ces environnements mettent à la disposition des développeurs : un éditeur intégré, un compilateur, un débuggeur sophistique et de nombreux générateurs de code (surtout concernant les interfaces graphiques).

3.1. Le Java Développement Kit(JDK) :[58]

Le JDK est un environnement gratuit comportant un compilateur et une machine virtuelle (minimal et suffisant). Il est téléchargeable à partir du site de Sun. Il comporte l'ensemble des éléments qui ont pour but le développement, la mise au point pour l'exécution des programmes JAVA.il peut être considérer comme un ensemble d'outils plus un jeu de classes et de service plus un ensemble de spécifications. Il existe plusieurs versions de JDK, il est important de connaitre la version employé car les classes disponibles peuvent être différentes d'une version à une autre.

4. Eclipse :[60]

Eclipse est un environnement de développement intégré (Integrated Development Environment), développer par I.B.M, dont le but est de fournir une plateforme modulaire pour permettre de réaliser des développements informatiques. Eclipse utilise énormément le concept de modules nommés « Plugins » dans son architecture.

Hormis le noyau de la plateforme nommé « Runtime», tout le reste de la plateforme est développé sous la forme de plugins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont fournies en standard par Eclipse. Les principaux modules fournis en standard avec Eclipse concernent Java mais des modules sont en cours de développement pour d'autres langages notamment C++, Cobol, mais aussi pour d'autres aspects du développement (base de données, conception avec UML,…..). Ils sont tous développés en Java soit pour le projet Eclipse en open source. Les modules agissent sur des fichiers qui sont inclus dans l'espace de travail (Workspace). L'espace de travail regroupe les projets qui contiennent une arborescence de fichiers. Bien que développé en Java, les performances a l'exécution d'Eclipse sont très bonnes car il n'utilise pas Swing pour l'interface homme-machine mais un toolkit particulier nommé SWT associé a la bibliothèque JFace, SWT (Standard Widget Toolkit) est développé en Java par IBM en utilisant au maximum les composants natifs fournis par le système d'exploitation sous-jacent, JFace utilise SWT et propose une API pour faciliter le développement d'interface graphique.

- **Les points forts d'Eclipse :[58]**

Eclipse possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Support de plusieurs plateformes d'exécution : Windows, Linux, Mac, OS X,…
- Une plateforme ouverte pour le développement d'applications et extensible grâce a un mécanisme de plugins.
- Plusieurs versions d'un même plugin peuvent cohabiter sur une même plateforme.
- Un support multi langage grâce à des plugins dédiés : Cobol, C, PHP, C#,….
- Malgré son écriture en Java, Eclipse est très rapide à l'exécution grâce à l'utilisation de la bibliothèque SWT.

- Un historique local des dernières modifications.
- Une exécution des applications dans une JVM dédiée sélectionnable avec possibilité d'utiliser un débogueur complet (points d'arrêts conditionnels, visualiser et modifier des variables, évaluation d'expression dans le contexte d'exécution, changement du code à chaud avec l'utilisation d'une JVM,….).
- Propose le nécessaire pour développer de nouveaux plugins.

Bibliographie

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

[2] M.Kherroubi,F.Bouguerra, « utilisation des techniques de DATAMINING pour l'extraction de modèle pedagogique»,2009/2010.

[3] Ch.Ramdane, « Le clustering des données : une nouvelle approche évolutionnaire quantique»,mémoire de magister, informatique Dept. Univ.constantine,2006

[4] V. Kumar ,«An Introduction to Cluster Analysis for Data Mining».Technical report, C.S. Dept. Univ. Minnesota, 2000.

[5] P. Berkhin, «Survey of Clustering Data Mining Techniques», Technical report, Accrue software, San Jose,California, 2002.

[6] A. K. Jain et R.C. Dubes . «Algorithms for Clustering Data», Prentice Hall advanced reference series,1988.

[7]W. Pedrycz. «Clustering and Fuzzy Clustering». chapitre1, In Knowledge Based Clustering, Wiley & Sons, Hoboken , 2005.

[8] G.Celeux., Nakache J.-P «*Analyse discriminante sur variables qualitatives.* Polytchnica (eds.),Paris». 1994.

[9] A.K. Jain, M.N. Murty et P.J. Flynn. «Data Clustering: A Review», ACM Computing Surveys, Vol. 31, No. 3,pp 264- 323 , 1999.

[10]M.Hammouda, « Utilisation des techniques de data mining pour la modelisation du parcours scolaire et de la prediction du succes et de risque d'echec».mémoire de magister,departement d'informatique.USDB.2009.

[11] N.Belacel, MR.Boulassel «*PROAFTN* classification method: A useful tool to assist medical diagnosis". *Rapport de recherche IS-MG 99/24*, Université Libre de Bruxelles, C.P. 210/01, B-1050 Bruxelles, Belgique, July 1999.

[12] R.J.Wilson et J.J.Watkins. «Graphs : an introductory approach : a first course in discrete mathematics. John Wily and Sons», New York, NY,1990.

[13] T. N. Tran, R. Wehrens, and L. M. Buydens.Clustering multispectral images: a tutorial.Chemometrics and Intelligent Laboratory Systems, 77(1–2):3–17, May 2005.

[14]H.Späth «*Cluster Analysis Algorithms for data reduction and classification of objects.* Ellis Horwood, Willy & Sons, New York. » 1980

[15]E.Diday «Optimisation en classification automatique et reconnaissance de formes», *Note Scient.IRIA n° 6*. 1972

[16] M. Halkidi, Y. Batistakis, M. Vazirgiannis, «On Clustering Validation Techniques », Intelligent Information Systems Journal, Kluwer Pulishers, vol.17, n°2-3, pp. 107-145,2001.

[17] M.Ester, H.P.Kriegel, J.Sander, et X. Xu. «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise ».In Proceeding of 2nd Int. Conference on Knowledge Discovery and Data Mining, pp. 226–231, 1996.

[18]S.B. Kotsiantis, P. E. Pintelas, «Recent Advances in Clustering: A Brief Survey », WSEAS Transactions on Information Science and Applications, Vol 1(1), pp.73–81,2004..

[19] M. Steinbach, L. Ertöz, et V. Kumar . «Challenges of Clustering High Dimensional Data », In L. T. Wille, editor, New Vistas in Statistical Physics– Applications in Econophysics, Bioinformatics, and Pattern Recognition. Springer-Verlag, 2003.

[20]M. Halkidi, Y. Batistakis, M. Vazirgiannis, «On Clustering Validation Techniques», Intelligent Information Systems Journal, Kluwer Pulishers, vol.17, n°2-3, pp. 107-145, 2001.

[21] J.Handl, «Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative techniques. Masters Thesis, Universitéd'Erlangen-Nurnberg, Erlangen, Germany», 2003.

[22] C. van Rijsbergen. «Information retrieval, second edition.Butterworths, 1979.

[23] W.Rand. «Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association», Vol 66 n336, pp 846-850, 1971.

[24]S.Lioyd.Least squares quantization in PCM.IEEE «Transactions on InformationTheory,Vol 28 n2,pp 127-138», 1982.

[25]M. Halkidi, Y. Batistakis, M. Vazirgiannis. «Cluster Validity Methods: Part II»,SIGMOD Record, 2002.

[26] M. Halkidi ,M.Vazirgiannis. «Clustering Validity Assessment: Finding the optimal partitioning of a data set »,in the Proceedings of ICDM Conference ,California, USA, 2001.

[27] M. HASCOET . «Visualisation d'information et interaction», Hermes.2004

[28] M .Riccardo, « Introduction to Information Visualization», University of Lugano, 2004

[29] Z.Bin & Ch. Hsinchun' « information Visualization », *Annual Review of Information Science and Technology*, Vol. 40, pp. 139-177, 2004

[30]D. Keim, «Information Visualization and Visual Data Mining», IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 7, NO. 1, JANUARY-MARCH 2002.

[31]C.HURTER «Caractérisation de visualisations et exploration interactive de grandes quantités de données multidimensionnelles», thèse, 2010

[32]G.Balmisse. «REFLEXIONS - Visualisation de l'information : quelques repères. » Retrieved jeudi 24 février 2005 from http://www.gillesbalmisse.com/blog/index.php?2005/02/24/35-visualisation-de-linformation-quelques-reperes.2005

[33] B.Shneiderman, «*The eyes have it: a task by data type taxonomy for information visualizations*», Proceedings of 1996 IEEE Visual Languages, Boulder, CO, pp. 336-343, 1996.

[34]http://membres.liglab.fr/leroy/documents/IFD3.pdf/ , 13/03/2013

[35]B. Shneiderman. «The eyes have it : A task by data type taxonomy for information visualizations». In Proceedings of the 1996 IEEE Symposium on Visual Languages, pages 336. IEEE Computer Society, 1996.

[36] S.Card, J.Mackinlay, B. Shneiderman, «B.: Readings in Information Visualization. Morgan Kaufmann », San Francisco (1999) 1-34

[37] G.L.Lohse , K. Biolsi, N.Walker, , H.Rueter: «A Classification of Visual Representations ». In: Communications of the ACM, Vol. 37, No. 12. ACM Press (1994) 36-49

[38]G.Jaeschke, P.Gupta, and M. Hemmje, Modelling Interactive, « Three-Dimensional Information Visualizations»,.E.J. Neuhold Festschrift, LNCS 3379, pp. 197 . 206, 2005. © Springer-Verlag Berlin Heidelberg 2005

[39]B. Shneiderman, «The eye have it: A task by data type taxonomy for information visualizations» in Visual Languages, 1996.

[40] L. Nowell, S. Havre, B. Hetzel and P. Whitney.Voir« information visualisation and visual.pdf»,2001

[41] J,Gerald and all, «Modelling Interactive, Three-Dimensional Information Visualizations»,2005

[42]F. Oliveira, H. Levkowitz, «From Visual Data Exploration to Visual Data Mining: A Survey», IEEE Trans.Vis.Comput. Graph, Volume 9(3), pp.378-394 (2003)

[43] K .Andrews . «Information Visualisation » http://www2.iicm.edu/ivis/ivis.pdf,2002

[44] J.Fekete and C .Plaisant. (2002). «Interactive Information Visualization of a Million Items».INFOVI,, IEEE Symposium on Information Visualization, Boston.http://hcil.cs.umd.edu/trs/2002-01/2002-01.html,2002

[45] DA. Keim. «Information Visualization and visual datamining».IEEE Transactions on Visualization and Computer Graphics.http://fusion.cs.uni-magdeburg.de/pubs/TVCG02.pdf,2002

[47] site officiel de l'outil GGobi, www.ggobi.org, 2000

[48] Site officiel de l'outil orange, http://orange.biolab.si/, 2010

[49] Site officiel de l'outil XmdvTool, http://davis.wpi.edu/xmdv/, 2007

[50] D.Keim et al,"Mastering the Information Age Solving Problems with Visual Analytics",2010

[51] D. A. Keim, F. Mansmann, and J. Thomas. Visual analytics: How much visualization and how much analytics? SIGKDD Explorations, 11(2):5–8, December 2009.

[52] N. Pitman, *UML 2 en concentré*. O'Reilly.2006

[53] K.Muller « modélisation objet avec UML », EYROLLES, 1997

[54]J.STEFFE-ENITA de Bordeaux « Cours UML», mars 2005

[55]K. Muller « modélisation objet avec UML », EYROLLES, 2002

[56]M. Bres, « atelier de génie logiciel », Masson 1993.

[57] G .pierre.Processus de développement logiciel (Université de Paris 13)2008

[58]http://www.istantic.com/v2/programmation/Java/Generalites/Generalites.htm,10/05/2013

[59]http://www.eclipse.org ,10/05/2013

[60]http://www.techno-science.net ,10/05/2013

[61] J.BOUGEAUT, « java la maitrise édition »2008.

[62] A. K. Jain and R. C. Dubes. « Algorithms for Clustering Data». Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1988.

[63] K. Doherty, R. Adams, and N. Davey. «Non-euclidean norms and data normalisation». In Proc. 12th Euro. Symposiumon Artificial Neural Networks, pages 181–186, Brugges, Begium, 2004.