République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saâd DAHLAB, Blida

N° D'ordre.....

Faculté des sciences
Département d'informatique

Présenté par :

KHEBBAB Houssem Eddine BENHAMZA Ryadh El Mahdi

En vue d'obtenir le diplôme de master
Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Sujet :

Analyse et exploration des résultats de la classification automatique de données multidimensionnelles

Soutenu le :

Mme. S.BENSTITI	Présidente
Mlle. F.Z. REGUIEG	Examinatrice
M. CHIKHI	Examinateur
Mlle. N.BENBLIDIA	Promotrice
Mlle. K.AMEUR	Encadreuse

Promotion 2012/2013

Remerciements

La reconnaissance et le merci reviennent au bon dieu de nous avoir aidés à concrétiser nos ambitions et de faire voir le jour à ce projet.

Nous tenons à remercier dans un premier temps, notre promotrice Mlle N. BENBLIDIA, pour nous avoir proposé ce sujet, pour la confiance qu'elle nous a attribué ainsi que pour nous avoir orientés tout au long de notre étude.

Nous ne saurions remercier de la meilleure des façons notre encadreuse, Mlle K. AMEUR, pour sa disponibilité ainsi que pour son importante aide qu'elle n'a cessé de nous apporter et ce dés le début de ce projet.

Merci de tout cœur.

Que nos parents respectifs trouvent ici l'expression de notre immense amour envers eux et de notre profonde reconnaissance pour leur soutient inébranlable.

Nous remercions les membres du jury pour avoir accepté de juger ce travail, et d'honorer notre soutenance de leur présence.

Nous tenons aussi à remercier toute personne ayant contribué à notre formation intellectuelle de prés ou de loin.

Résumé

Le développement rapide des outils informatiques permet au système informatique de stoker de très grandes quantités de données. On parle de grandes bases de données selon les deux axes : nombre d'enregistrements et nombre de dimensions "attribut, variable". L'analyse de ces données devient à la fois très importante mais difficile. Les techniques de visualisation de l'information contribueront à résoudre ce problème, l'exploration visuelle des données possède un fort potentiel d'application étant donnée qu'elle facilite l'analyse, l'interprétation, la validation et augmente par conséquent l'aspect cognitif chez les analystes. Dans ce contexte, le but de notre étude est la conception et la réalisation d'un système permettant d'améliorer le processus d'analyses de résultats du clustering des données multidimensionnelles via l'utilisation des technologies de visualisation d'informations. Nous nous sommes intéressés à combiner les résultats de l'algorithme de clustering non supervisé, "K-means", avec la technique de visualisation multidimensionnelle du "parallel coordinates" afin d'analyser de grands ensembles de données.

ملخص

التطور السريع لأدوات الكمبيوتر سمح للنظم المعلومات بحفظ كمية كبيرة جدا من البيانات ذات أبعاد متعددة. نحن نتحدث عن قواعد البيانات واسعة على أساس بعدين: عدد البيانات وعدد الأبعاد. تحليل هذه البيانات أصبح مهم جدا وفي نفس الوقت صعب. يمكن لتقنيات العرض متعددة الأبعاد للمعلومات استخدامها لحل هذه المشكلة لان تقنيات الاستكشاف البيانية لها إمكانات كبيرة من التطبيقات كونها تسهل تحليل وتفسير البيانات وكذا تطور الجوانب المعرفية لدى المحللين. في هذا السياق، فإن الهدف من دراستنا هو تصميم وتنفيذ نظام الذي يحسن من عملية تحليل نتائج تجميع بيانات متعددة الأبعاد من خلال استخدام تقنيات التصور المعلومات. لذا فنحن مهتمون في الجمع بين خوارزمية التجميعK-Means مع تقنية العرض متعددة الأبعاد Parallel Coordinates لعرض نتائج هذه الخوارزمية.

Abstract

The rapid development of computer tools allows the computer system to stoke very large amount of data with many parameters. We talk about large data bases along both dimensions: number of recordings and number of dimensions "attribute, variable". Analysis of these data becomes very important and difficult in the same time. Techniques of information visualization can be used to solve this problem. The visual

data mining has great potential for applications because it facilitates the analysis, interpretation, validation and increases the cognitive aspect among analysts.

So in this context, the aim of our study is the design and the implementation of a system that improves the analysis process of clustering results of multidimensional data through the use of technologies of information visualization. We are interested in the combining clustering algorithm K-means with the technique of visualization multidimensional parallel coordinates to visualize the clustering results.

Chapitre I — Clustering de données

Chapitre II — Les techniques de visualisation

Liste des figures

Liste des tableaux

Introduction Générale

Introduction générale

- **Contexte global**

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). Il n'hésite pas à utiliser des méthodes graphiques afin de mieux appréhender des notions souvent abstraites.

De nos jours, le développement rapide des outils informatiques permet aux systèmes de stocker de très grandes quantités de données, nous avons affaire aux déluges de données, l'analyse de ces dernières devient très difficile, voir fastidieuse. Des méthodes de classification et de visualisation efficaces sont des alternatives crées pour représenter ces immenses ensembles de données. L'exploration visuelle des données offre donc des solutions grâce à ses nombreux apports tels que la facilité d'analyse, des techniques de visualisation simples à interpréter et les possibilités d'interactions graphiques.

L'objectif de ce travail est de concevoir et de réaliser un outil graphique permettant à l'utilisateur ou à un expert de visualiser les résultats de la classification automatique de données avec différentes techniques afin d'explorer et d'analyser ces dernières.

- **Problématique**

L'évolution de l'informatique, matériel tout comme logiciel ainsi que l'avènement de l'internet ont conduit au développement des structures et des techniques de création, de stockage et de consultation de données. Ceci n'est pas sans conséquences quant à la difficulté rencontrée par l'utilisateur lors de l'analyse de ces grandes structures. De nombreuses solutions ont émergé à travers les générations, le clustering était une des solutions les plus intéressantes, l'idée 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. Tout ceci dessine notre problématique qui est d'offrir à l'utilisateur un moyen fiable de visualiser les données multidimensionnelles et sur plusieurs enregistrements.

- **Objectif**

Notre but est d'arriver à concevoir un système à la fois simple d'utilisation mais surtout capable de réaliser un clustering précis dans un premier temps puis la représentation fidèle des résultats de ce regroupement à travers une des nombreuses techniques de visualisations. Les interactions réalisées sur l'outil graphique feront partie de notre objectif afin d'assurer une meilleure exploration à l'utilisateur.

- **Organisation**

Une fois notre problématique cernée, nous proposons de réaliser notre objectif et de rédiger notre mémoire dans la structure suivante :

Chapitre I : Concerne l'introduction aux différentes méthodes de classifications. Nous présenterons les différentes mesures de distances existantes entre les différents types de données pris en charge par notre outil graphique. Nous nous orienteront à fur et à mesure vers le clustering automatique (regroupement par partitionnement) nous décrirons l'algorithme choisi et toutes ces itérations. Il sera ensuite question de valider les résultats du regroupement choisi, c'est pour cela que nous expliquerons les différentes manières de valider un clustering.

Chapitre II : Dans ce chapitre, nous introduirons le domaine de visualisation de données et ces différentes techniques existantes. Nous allons au départ décrire les objectifs d'une visualisation pour ensuite présenter le processus de visualisation adopté. Une fois ces étapes franchies, nous expliqueront brièvement le fonctionnement de quelques techniques de visualisation connues et ce dans le but d'en choisir une dans un premier temps.

Chapitre III : Ce chapitre sera le chapitre qui présentera la solution à notre problématique. Nous définirons notre approche et le processus à réaliser, vient ensuite l'analyse des besoins et des ressources afin de concevoir notre outil graphique (EyeViz). C'est dans ce chapitre qu'on présentera le cycle de vie de notre application ainsi que le

langage utilisé pour sa modélisation et les différents diagrammes présentant ses interactions et ses fonctionnalités.

Chapitre IV : Le dernier chapitre de notre mémoire portera sur la réalisation de notre application et les résultats des différents tests effectués, clôturant ainsi le cycle de vie de la réalisation de notre système présenté dans le chapitre précédent. C'est dans ce chapitre qu'on décrira l'environnement de travail ainsi que le langage de programmation choisi. Des captures d'écrans présenteront des interfaces de différentes étapes de l'exécution d'EyeViz.

Enfin, nous arriverons à la conclusion générale dans laquelle nous parlerons des perspectives à venir.

Chapitre I

Clustering de données

1. Introduction

« Un être intelligent ne peut traiter des objets considérés comme entité unique différente du reste de l'univers. Il doit mettre ces objets dans des catégories afin d'appliquer sa connaissance acquise sur des études similaires rencontrées dans le passé… » [Notre traduction] [1].

L'une des capacités les plus élémentaires des êtres vivants implique le regroupement d'objets similaires pour produire une classification. L'idée de ranger les objets similaires dans des catégories adéquates est clairement une définition primitive depuis le début de l'humanité, par exemple l'homme devait se rendre compte que certaines choses qu'il possédait ou qu'il partageait pouvaient être comestibles, toxiques ou féroces et ainsi de suite. [2]

Dans la littérature il n'existe pas de définition précise au regroupement. Les classes ou clusters regroupent des objets similaires, mais la notion de similarité elle-même est ambigüe car elle peut varier d'une application à une autre. Les données peuvent présenter des structures de formes et/ou tailles différentes et donc, celles-ci peuvent être regroupées sous les hypothèses très différentes. [3]

2. Généralités sur le regroupement de données (clustering)

Le regroupement effectué par l'homme durant des milliers d'années a été automatisé au cour des dernières décennies grâce au développement des technologies, on parle alors de la classification automatique de données. Mais jusqu'ici les termes cluster, groupe et classe ont été utilisés de manière totalement intuitive sans aucune définition formelle.

En fait il s'avère que la définition formelle d'un « cluster » est non seulement difficile mais peut même être déplacée. En 1964, Bonner, par exemple, a suggéré que le critère ultime pour évaluer la signification des termes du cluster est la valeur du jugement de l'utilisateur. [2].

On peut retenir les définitions suivantes [3]:

- Un cluster est un ensemble d'entités qui sont semblables, et les entités des différents clusters ne sont pas semblables.

- Un cluster est une agrégation des points dans l'espace d'essai tel que la distance entre deux points quelconques dans un cluster est inférieure à la distance entre n'importe quel point de ce cluster et n'importe quel point qui ne se trouve pas dans ce cluster.

- Les clusters peuvent être décrits en tant que régions dans un espace multidimensionnel caractérisées par une haute densité de points, séparées d'autres telles régions par une région caractérisée par une densité de points relativement faible.

Dans les deux dernières définitions les objets sont considérés comme des points dans un espace multidimensionnel.

En général, un algorithme de clustering présente les étapes suivantes :

- ***Le choix de mesure de proximité approprié au domaine de données :***

Cette mesure doit assurer que tous les attributs contribuent au calcul de la mesure de proximité et qu'aucun attribut ne domine l'autre.

- ***Définition du critère de clustering :***

qui peut être exprimé par l'intermédiaire d'une fonction objective ou d'un autre type de règles.

- ***Représentation simple et compacte de l'ensemble de données et l'ensemble de clusters.***
- ***Le clustering :***

Etape qui peut être réalisée de nombreuses façons, selon la technique adoptée.

- ***La validation des résultats :***

La qualité des résultats du clustering est vérifiée en employant des mesures de validités des clusters, ces critères de validations sont internes et externes.

3. Concepts de bases

3.1. La matrice de données

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.

La figure 1.1 ci-dessous présente une matrice de données d'un objet et ses attributs.

Attributs

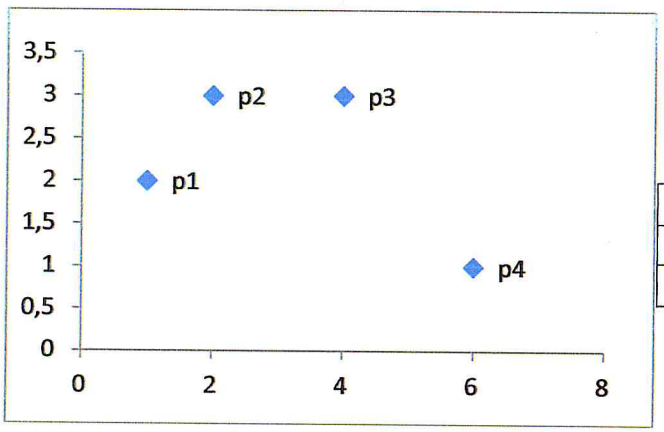
$$\begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1D} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{iD} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{N1} & \cdots & x_{Nj} & \cdots & x_{ND} \end{bmatrix}$$

Figure 1.1. Matrice de données d'un objet.

3.2. Matrice de proximité

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 P_i et P_j sont le $i^{ème}$ et le $j^{ème}$ objets, respectivement, alors l'entrée à la $i^{èmme}$ ligne et la $j^{èmme}$ colonne de la matrice de proximité est la similarité, ou la dissimilarité, entre P_i et P_j de cette manière, la matrice de proximité est une matrice carrée avec des valeurs de diagonales nuls.

Les figures 1.2 présente la matrice de donnée de quatre points ainsi que leur représentation graphique



	P1	P2	P3	P4
x	1	2	4	6
y	2	3	3	1

Figure 1.2. Quatre points et leur matrice de données (jeux de données)

La figure 1.3 représente la matrice de proximité des quatre points précédents.

	P1	P2	P3	P4
P1	0	1.41	3.16	5.09
P2	1.41	0	2	4.47
P3	3.16	2	0	2.82
P4	5.09	4.47	2.82	0

Figure 1.3. Matrice de proximité des quatre points

3.3. Types d'un cluster

La définition d'un cluster et de ce qu'il peut représenter n'est pas bien définie [35], et le terme 'cluster' n'a pas de définition précise, cependant plusieurs définitions d'un cluster sont généralement utilisées

1. **Clusters bien séparés :** dans cette définition, un cluster est représenté par un ensemble de points où chaque point est plus proche (similarité) d'un autre point au sein d'un même cluster que d'un autre point se trouvant dans un autre cluster. Un seuil est parfois utilisé pour garantir que n'importe quel point est plus proche d'un autre point dans le même cluster. Ce pendant, dans beaucoup de matrice de données, un point représenté sur le bord d'un cluster peut être plus proche d'autres points appartenant à un cluster voisin, par conséquent plusieurs algorithmes de clustering utilisent la définition suivante.

2. **Clusters basés sur le centre :** dans ce type de définition, chaque point appartenant à un cluster est en réalité plus proche (similaire) du centre de ce cluster que du centre de n'importe quel autre cluster. Le centre d'un cluster est appelé « centroïde » la moyenne de tous les points dans le cluster est appelée « médoïde » et c'est le point le plus représentatif d'un cluster.

3. **Cluster contigüe :** basé sur le voisin le plus proche (clustering transitif) ici, le cluster est un ensemble de points tel qu'un point dans un cluster est plus proche (similaire) d'un ou de plusieurs autres points dans le cluster que de n'importe quel autre point qui n'est pas dans ce cluster.

4. **Clusters basés sur la densité :** 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.

3.4. Types et échelles de données

La mesure de proximité et le type de clustering utilisé dépendent des types et échelles des attributs de données [34]. Le tableau suivant représente les types d'attributs les plus utilisés.

Binaire	Deux valeurs (vrai ou faux)
discret	Un nombre fini de valeurs
continu	Un nombre infini de valeurs

Tableau 1.1. Types d'attributs.

Le tableau suivant représente les échelles de données :

qualitative	nominal	Les valeurs sont juste des noms différents, exemple : la couleur, le sexe.
	ordinal	Les valeurs reflètent un ordre, exemple : bon, moyen et mauvais.
quantitative	intervalle	La différence entre les valeurs est significative, exemple : l'intervalle de température.
	ratio	Rapport entre deux grandeurs.

Tableau 1.2. Echelles de donnés : une donnée peut être qualitative ou quantitative.

3.5. Distance et similarité

Le concept de similarité ou de dissimilarité est le composant essentiel de n'importe quelle forme de clustering qui nous aide à naviguer dans l'espace de données pour former un cluster [36].

Une bonne méthode de regroupement permet d'assurer :

- Une faible similarité inter-groupe.
- Une grande similarité intra-goupe.

En calculant la similarité, nous pouvons constater à quel point deux points sont proches, et sur la base de cette proximité, nous pouvons les assigner au même cluster.

Formellement, la similarité $d(x,y)$ entre x et y est considérée 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 utilisée 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**

Les fonctions de distances entre variables continues les plus connues sont représentées les suivantes :

Nom	Fonction
distance de Manhattan	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$
distance euclidienne	$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
distance de Minkowski	$\sqrt[p]{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p}$

Tableau 1.3. Les fonctions de différentes distances connues [36]

- Avec : x_i et y_i sont les différentes valeurs qui représentent les deux points x et y qui disposent de n dimensions.
- $sup_{1 \leq i \leq n} \lvert x_i - y_i \rvert = max_{i=1,2,\ldots,n} \lvert x_i - y_i \rvert$

  Ce qui signifie que cette distance est basée sur la valeur maximale entre les attributs de x et y.

- **Distance des variables 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.

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 1.4. Matrice de contingence pour les données binaires.

Exemple :

F (1, 1, 0, 1, 0) et G (1, 0, 0, 0, 1) ici nous avons a = 1, b = 2, c = 1, d = 1.

- Variables symétriques :

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

Exemple : le sexe peut être codé comme suite :

Si deux personnes sont du même sexe on attribut 0 à la distance et 1 si elles sont de sexe différent.

- Variable asymétriques :

$$d(i,j) = \frac{b+c}{a+b+c}$$

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.

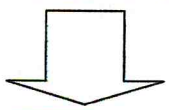
- **Distance pour les valeurs qualitatives**

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.

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 1.5. Matrice de données initiale



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 1.6. Matrice précédente traitée et préparée pour le calcul de distances

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.

$$d(1,2) = \frac{0+1}{2+0+1} = \frac{1}{3} = 0.33 \qquad d(1,3) = \frac{1+3}{0+1+3} = \frac{4}{4} = 1$$

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

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.

4. Principales techniques de clustering

Depuis de nombreuses années, plusieurs méthodes et techniques ont été mises en œuvre pour le regroupement de données (clustering) chaque techniques est en fait centrée sur un algorithme réalisant des itérations propres à la technique et qui différent entre eux dans la philosophie du regroupement. Les algorithmes de clustering peuvent être classifiés comme selon [37] :

- Le type de données en entrée.
- Les critères de clustering définissant la similarité entre les points de données.
- La théorie et les concepts fondamentaux sur lesquels les techniques de clustering sont basées.

Les techniques de clustering ont été largement étudiées dans les domaines de statistique, l'apprentissage automatique et le datamining.

Ainsi, les algorithmes de clustering différent selon les méthodes auxquels ils appartiennent, ces méthodes sont les suivantes :

- Clustering par partitionnement
- Clustering hiérarchique
- Clustering basé sur les grilles
- Clustering basé sur la densité

Chacune des techniques de clustering offre de nombreuses sous-types et plusieurs algorithmes différents. Dans notre travail nous nous intéressons au clustering par partitionnement.

4.1. Clustering par partitionnement

Ce type de clustering appartient au clustering non supervisé ou les données ne sont pas étiquetées. Les techniques par partitionnement créent un partitionnement des points de

données, d'un seul niveau. Si k est le nombre désiré de clusters, alors les approches par partitionnement trouvent typiquement tous les k clusters immédiatement.

Les techniques par partitionnement sont divisées en deux sous-catégories principales [38], les algorithmes basés sur les centroïdes et les algorithmes basés sur le medoïdes. Nous allons décrire les deux algorithmes les plus connus : K-moyenne (Kmeans) et Kmedoïd.

Ces deux techniques sont basées sur l'idée qu'un point de centre peut représenter un cluster. Pour Kmeans on emploie la notion du centroïdes qui est le point de la moyenne ou la médiane d'un groupe de points. Pour Kmedoïd on utilise la notion d'un medoïde qui est le point le plus représentatif (central) d'un groupe de points.

- **K-means**

Kmeans est une technique importante de clustering, il est très populaire dans de nombreux domaines d'applications tels que l'analyse d'images, la recherche marketing, la bioinformatique et l'informatique médicale.

En général, le processus de recherche de clusters en fonction du Kmeans commence par K centroïdes provisoires et s'applique en deux étapes :

(a) La collecte de clusters autour de centres de gravité (centroïdes),
(b) Mettre à jour les centroïdes (la moyenne des clusters), jusqu'à la convergence.
(c) Rapprochement de chaque centre de cluster vers le cluster lui correspondant.
(d) Représentation des différents clusters et leurs centres.

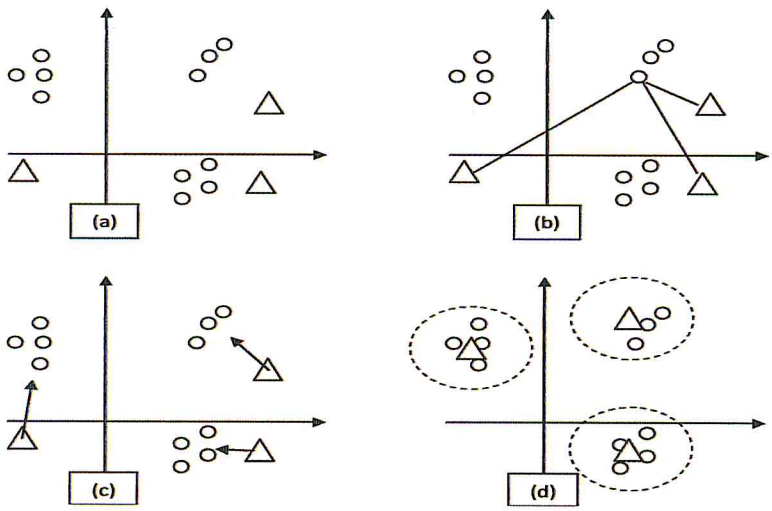


Figure1.4. Etapes de l'algorithme K-means

En général les étapes du Kmeans sont schématisées dans la figure ci-dessus en 4 étapes qui sont les suivantes :

Les points appartiennent à l'ensemble de donnée I.

Etape 0. L'initialisation : l'utilisateur défini le nombre K de clusters et choisi K hypothétiques centres de clusters parmi l'ensemble des points (Figure1.4. (a)).

Etape 1. Mise à jour des clusters : compte tenu des K centres, chacune des entités (points) $i \in I$ est affectée à l'un des centres de gravité selon la règle de la distance minimale. Les distances entre i et chaque centre (K) sont calculées et i est affecté au centre le plus proche (Figure1.4. (b)). Pour chaque centre K, les entités qui lui sont affectées forment un cluster (Figure 1.4. (c)).

Etape 2. Mise à jour des centroïdes : pour chaque cluster K donné, son centre de gravité est calculé et défini comme le nouveau centre de gravité Ck', Figure 1.4. (d).

Etape 3. Fin des itérations : les nouveaux centres Ck' sont comparés à ceux de l'itération précédente, si Ck' = Ck pour tout K = 1, 2, …, K alors mettre terme aux itérations de l'algorithme et sortie. Dans le cas contraire, mettre Ck' = Ck et allez a Etape 1.

La complexité temporelle du Kmeans est $O(I*k*m*n)$ où I est le nombre d'itérations exigés pour la convergence, K le nombre de clusters, m le nombre de points et n le nombre d'attributs. Kmeans est efficace et simple tant que le nombre de clusters est significativement inferieur au nombre de points m.

Le choix approprié des centroïdes initiaux est l'étape clef de l'algorithme. Il est facile de choisir des centroïdes initiaux de manière aléatoire mais l'inconvénient est que la partition finale dépend fortement de la partition initiale.

Le principal avantage de l'algorithme Kmeans est sa complexité temporelle, qui le rend efficace dans le traitement de grands jeux de données, mais il a un certain nombre de limitations et de problèmes tels que :

- Le résultat dépend fortement de la conjecture initiale de centroïdes,
- Il se termine souvent à un optimum local,
- Le processus est sensible aux bruits,
- Il tend à trouver des clusters sphériques de tailles égales,
- Seulement les attributs numériques sont couverts.

Plusieurs variantes du Kmeans existent, certaines d'entres tentent de choisir une bonne partition initiale pour que l'algorithme trouve la valeur du minimum global. Une autre

variante consiste à fractionner ou fusionner les clusters résultants. Typiquement un cluster est fractionné quand sa variance est au-dessus du seuil prés spécifié et deux clusters sont fusionnés lorsque la distance entre leurs centroïdes est au-dessous d'un autre seuil pré spécifié [33]. L'algorithme le plus connu qui utilise la technique de fusion et de fractionnement de clusters est ISODATA, il est utilisé particulièrement dans le domaine du traitement de l'image [34].

- **Kmedoid**

Dans l'approche Kmedoid, un cluster est représenté par un de ses points. Ce point représentatif est appelé medoïde, c'est le point le plus centré. Dans l'algorithme Kmedoid certaines mesures sont prises en compte, comme par exemple la distance.
Le Kmedoid est décrit de manière simple comme suit [35] :

1. Choisir k points initiaux. Ces points sont les medoïdes candidats qui sont destinés à être les points les plus centraux de leurs clusters.

2. Remplacer les points choisis (medoïdes) par des points non choisis comme suit :
   On calcule la distance entre chaque point non choisi et la medoïde candidate la plus proche, puis on calcule la somme de toutes les distances, cette somme représente le « coût » de la configuration actuelle. Tous les échanges possibles d'un point non choisi par un point choisi sont considérés, et le coût de chaque configuration est calculé.

3. Choisir la configuration présentant le coût le plus bas. Si c'est une nouvelle configuration alors répéter l'étape 2.

4. Sinon, associer chaque point non choisi au point choisi le plus proche (medoïde) et arrêter.

Le $i^{ème}$ medoïde est calculé en utilisant $\sum_{j=1}^{n_i} p_{ij}$ où p_{ij} est la proximité entre le $i^{ème}$ medoïde et le $j^{ème}$ point dans le cluster. Pour la similarité (dissimilarité) la somme devra être la plus grande possible (petite).

Cette approche n'est pas limitée aux espaces euclidiens. En outre, l'utilisation de medoïde pour définir des clusters rend cette méthode résistante aux bruits mais la complexité temporelle est $O(k\ (m - k)^2)$ où m est le nombre de points du jeu de données [37]. La dégradation est faite dans les étapes 2 et 3 de l'algorithme, puisque la découverte d'un meilleur medoïde exige l'essai de tous les points qui ne sont pas medoïdes. Cela est très couteux en temps de calcul.

Plusieurs algorithmes basés sur la notion de medoïde existent comme PAM (Partitioning Around Medoids) qui est un algorithme Kmedoid qui regroupe un ensemble de m points en k clusters en exécutant les étapes décrites ci-dessus. CLARA (Clustering LARge Applications) est une adaptation de PAM pour manipuler de grands jeux de données.

CLARANS est né des deux clustering PAM et CLARA. CLARANS est un des premiers algorithmes de clustering dédié au datamining spatial.

4.2. Clustering hiérarchique

Les techniques du Clustering hiérarchiques sont généralement des méthodes dures et consistent à trouver une organisation arborescente des classes ou un dendrogramme. La plupart de ces méthodes dérivent des algorithmes de lien minimal *single-link*, de l'algorithme de lien maximal *complete-link* et de la méthode de variance minimale ou méthode de Ward. Le dendrogramme ainsi obtenu peut être coupé à n'importe quel niveau pour obtenir le nombre de classes désiré (Figure 1.5). Pourtant, déterminer le nombre exact de classes est très difficile. La visualisation du dendrogramme représente un moyen mais ceci est utile seulement pour un nombre réduit de données.

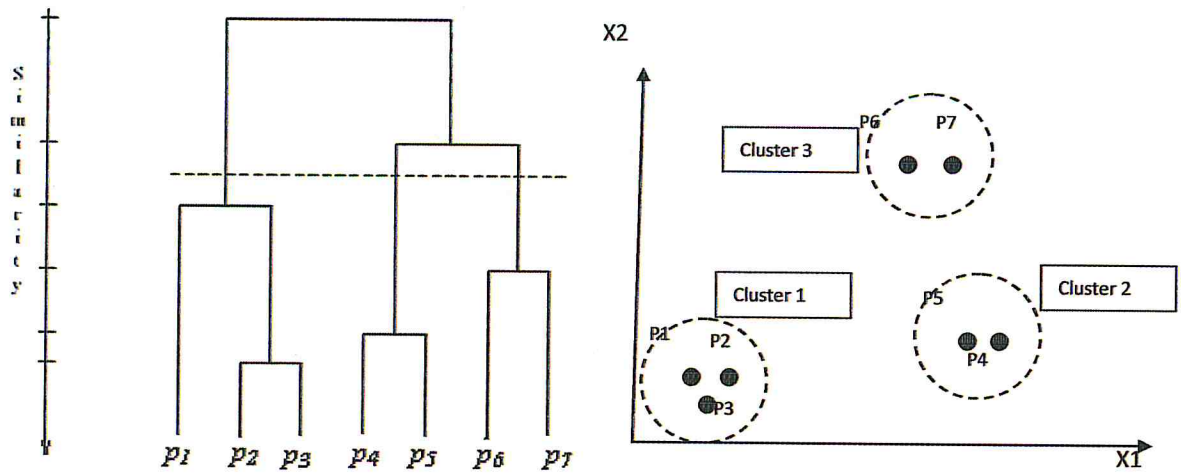


Figure 1.5. Clustering de sept points et le dendrogramme correspondant

Deux approches du clustering hiérarchique sont connues :

Clustering hiérarchique ascendant : commence par des clusters d'un seul point (singleton) et fusionne récursivement deux clusters ou plus selon leur similarité.

Clustering hiérarchique descendant : tous les points forment un seul cluster initialement, cet algorithme fractionne récursivement les clusters selon leur dissimilarité.

Le clustering hiérarchique descendant étant le moins commun, nous nous somme concentré sur les techniques du clustering hiérarchique ascendant.

- **Techniques de clustering hiérarchique ascendant (CHA)**

Plusieurs techniques hiérarchiques ascendantes peuvent être exprimées par l'algorithme suivant connu sous le nom de l'algorithme de Lance-Williams [39] :

1. Calculer la matrice de proximité.
2. Fusionner les deux clusters les plus proches (les plus similaires).
3. Mettre à jour la matrice de proximité pour refléter la proximité entre le nouveau cluster et les clusters originaux.
4. Répéter les étapes 3 et 4 jusqu'à obtention d'un seul cluster regroupant tous les points initiaux.

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.

[3]

Rapprochement de deux clusters

- **Lien simple (simple link)**

Il s'agit de définir la distance entre deux clusters comme étant la plus petite distance parmi celle entre toutes les paires d'objets entre deux clusters.

$D(C_P, C_Q) = \{ min\ d(X_i, X_j) / X_i \in C_P, X_j \in C_Q \}$

5. Techniques de validation du clustering

La plupart des méthodes de classification ne sont pas capables de détecter le nombre optimal de classes automatiquement. Souvent, le nombre de classes doit être défini soit de manière directe (Kmeans), soit de manière indirecte (méthodes hiérarchiques). En général le nombre de classes est déterminé à l'aide d'un critère de validité. Les indices de validité sont des critères permettant de choisir le nombre optimal de classes pour un certain algorithme, ou de valider les résultats de la classification en le comparant avec d'autres résultats obtenus soit par un autre algorithme, soit par le même algorithme paramétré différemment. On distingue deux catégories de mesures : interne et externe.

- ***Indices de mesures internes :*** confirment ou infirment les résultats de la classification en se basant sur les données existantes dans l'ensemble initial ;
- ***Indices de mesures externes :*** se basent sur les informations à priori (généralement une vérité de terrain) pour la validation des résultats.

5.1. Indices internes de validité de clustering

- **Indice de Davies-Bouldin**

  L'indice de Davies-Bouldin [40] tient compte en même temps de la compacité et de la séparabilité des classes. Si les classes sont compactes et séparées, alors la valeur de cet indice est faible. Si l'indice de Davies-Bouldin est estimé comme une fonction dépendante du nombre de classes, alors le nombre optimal de classes est donné par le point de minimum global de cette fonction. Cet indice favorise les classes de forme hypersphérique et il est adapté aux méthodes à centre mobile, Kmeans par exemple.

$$\text{DB} = \frac{1}{C}\sum_{i=1}^{C} max_{j=1,\dots,C}(d_{ij}), \text{ où } d_{ij} = \frac{\sigma_i+\sigma_j}{d(g_i,g_j)}$$

  Dans cette expression, C représente le nombre de classes, σ_i est la distance moyenne entre les objets et le centre de la classe C_i et $d(g_i, g_j)$ est la distance entre les centres de classes g_i et g_j. Ainsi, la distance d_{ij} aura une valeur faible si les classes sont compactes et bien séparées. La complexité de calcul de cet indice est faible.

- **Indice de Dunn**

  Soit d_{min} la distance minimale entre deux objets de classe différente et d_{max} la distance maximale entre deux objets de la même classe. Alors, l'indice Dunn D, est défini par [41] :

$$D = \frac{d_{min}}{d_{max}}$$

  Une bonne classification est indiquée par des valeurs élevées de cet indice. Le temps de calcul de cet indice est un inconvénient majeur quand on manipule de très grands ensembles de données. Ceci, ainsi que sa faible sensibilité au bruit font que cet indice est rarement utilisé.

- **Indice C_0**

  L'indice C_0 mesure la compacité des classes et est défini par [42] :

$$C_0 = \frac{S - S_{min}}{S_{max} - S_{min}}$$

  Dans cette formule, S représente la somme des distances entre toutes les paires d'objets dans une classe. Soit ι le nombre des paires d'objets dans une classe, alors S_{min} représente la somme des ι plus petites distances si toutes les paires d'objets sont considérés et S_{max} représente la somme des ι plus grandes distances issues de toutes les paires d'objets. Le dénominateur sert à la normalisation, $C_0 \in [0, 1]$.

  Cet indice est particulièrement bien adapté si les classes sont de taille similaire et il est de plus petit si la classe est plus compacte.

- **Indice de compacité-séparabilité**

  L'indice de compacité-séparabilité CS est décrit pour une partition floue et il utilise le même principe que l'indice DB : il tient compte à la fois de la compacité c_0 et de la séparabilité δ_e des classes. Pour une classification dure il est défini par [43] :

$$CS = \frac{c_0}{\delta_e}$$

$$\text{Où } c_0 = \frac{1}{C}\sum_{i=1}^{C} \sigma_i \text{ et } \delta_e = min_{i \neq j}(d(g_i, g_j))$$

5.2. Indice externes de validité de clustering

- **La F-mesure**

  La F-mesure est une fonction utilisée souvent pour évaluer les algorithmes de clustering [44]. La F-mesure adopte les idées de la précision et du rappel de la recherche documentaire. Elle compare la qualité d'un regroupement en tenant en compte des classes correcte connues pour un jeu de données. Soit C = $(C_1, C_2, \ldots, C_k)$ un clustering donné et R = $(R_1, R_2, \ldots, R_{k'})$ les classes correctes.

  Chaque classe R_i contient N_i points de données, chaque cluster C_j (généré par l'algorithme) est considéré comme l'ensemble de N_j points de données. N_{ij} donne le nombre de points de la classe R_i dans le cluster C_j et N donne le nombre total des points du jeu de données. Pour chaque classe R_i et un clusterC_j, la précision et le rappel sont alors définis comme :

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

  Et la valeur de F-mesure correspondante est :

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

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

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

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

- **La pureté**

  La pureté d'un cluster est définie comme le pourcentage du type de données prédominant selon la classe réelle connue et qui est [45] :

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

  Où N_j est la taille du cluster C_j et N_{ij}est le nombre de points de données de la classe R_i dans ce cluster. La pureté P(C) d'une partition entière est alors calculée comme la pureté moyenne de tous les clusters. Elle est limitée à l'intervalle] 0,1] et devrait être maximale.

- **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 [45] :

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

  L'entropie global E(C) est calculé en faisant la moyenne des entropies de clusters, l'entropie de cluster est limité à l'intervalle [0,1] devrait être minimal.

6. Problèmes liés au clustering

Le clustering de données est identifié comme une des problématiques majeures en extraction de connaissances à partir de données.

Bien qu'intuitif, le clustering est néanmoins difficile à réaliser dans la pratique, cette difficulté est causée par un manque de définition unique et précis d'un cluster dû à un manque d'informations préalables sur les distributions de données. Les principaux problèmes liés à la classification non supervisées des données multidimensionnelles sont les suivants :

- Des classes de formes complexes

La forme des clusters est un des problèmes majeurs dans la classification non supervisée. Les méthodes basées sur les centres des classes (par exemple le K-means [46] donnent de bons résultats pour des classes de forme convexes, voire sphérique ou ellipsoïdale, Figure 1.6.(a), (b). l'incapacité de mettre en évidence des classes de forme non convexe, figure 1.6.(c), représente le principal défaut de ces méthodes.

- Le nombre des classes

Le choix du nombre de classe pose probablement les plus grands défis en ce qui concerne la classification non supervisée des données. En l'absence d'informations à priori sur les données, les résultats de la classification sont validés par l'évaluation d'indices de validité vus précédemment, ceux-ci nous offrent une information sur la cohérence de la partition faite par une certaine méthode.

- La taille inégale des classes

Si la population des classes est très différente, ceci peut influencer les résultats de la classification. Des situations peuvent exister où une classe importante mais de faible population n'est pas mise en évidence parce que plusieurs classes importantes en nombre d'individus dominent le résultat de la classification.

- Le nombre d'attributs

L'augmentation du nombre d'attributs améliore la résolution spectrale des images multi variées. Ceci peut être vu comme un avantage car on dispose d'un plus d'information pour l'analyse, mais le traitement mathématique devient de plus en plus compliqué. Des méthodes de réduction du nombre des attributs doivent être mises en œuvre pour éviter les problèmes liées à la complexité des calculs mathématique ainsi que pour éviter de prendre en compte l'information redondante.

- Le bruit

L'acquisition des images est souvent accompagnée par la superposition d'un bruit sur l'information utile. Le bruit peut avoir des origines différentes : la sensibilité du capteur, des interférences ou des variations d'une autre nature. La présence du bruit favorise l'apparition de données aberrantes qui peuvent rendre les résultats très difficiles à interpréter, ou pire, donner des solutions inexactes. Des prétraitements pour supprimer le bruit sont souvent indispensables.[3]

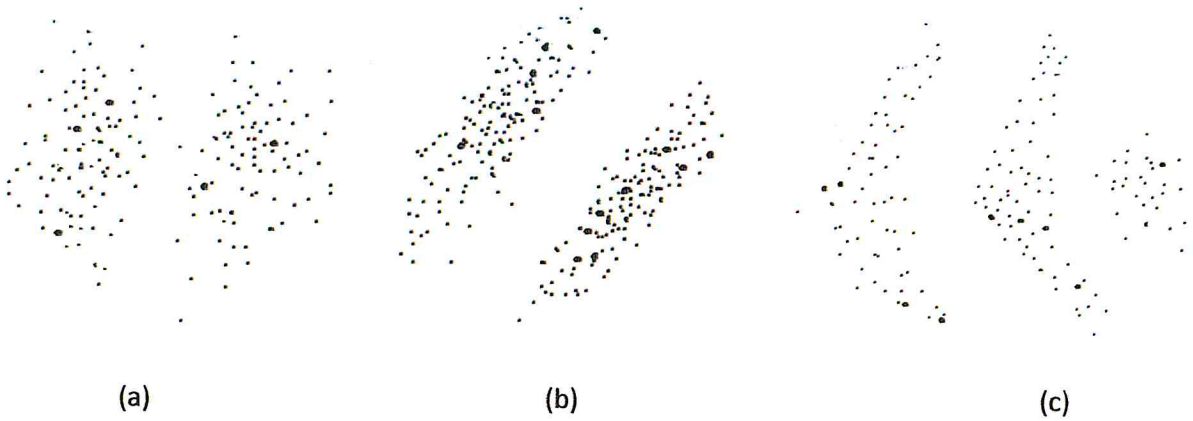


Figure 1.6. Représentation de trois types de bruits sur des informations

7. Conclusion

Au cour de ce chapitre nous avons défini notre ensemble de données en entré ainsi que les différents types de données qui le composent, nous avons aussi présenter la manière de calculer les différentes distances existantes. Enfin, nous nous sommes dirigés vers la présentation des familles de clustering pour en choisir celui qui nous intéresse. mais néanmoins, ce chapitre annonce que le clustering peut présenter des faiblesses quand il s'agit de données multidimensionnelles, et que l'analyse des clusters demeure aussi difficile que l'analyse des données sources, pour cette raison, nous proposerons dans le chapitre suivant, une étude concernant la représentation visuelle de données multidimensionnelles et essayer de l'adapter aux résultats de clustering pour une meilleure exploitation.

Chapitre II

Les techniques de visualisation

1. Introduction

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.

Les développements en informatique ont conféré une grande capacité de recueil et de génération de données mais également une grande puissance au niveau des techniques d'imageries. [4]

Au cour de ce chapitre nous nous intéressons à la visualisation comme étant une solution aux complexités des résultats du clustering, nous introduirons les types et modèles de visualisations et nous nous orienterons vers la visualisation des données multidimensionnelles.

2. Paradigme de l'exploration visuelle

Ce que Ben Shneiderman nomme l' « information seeking Mantra »-« Overview first, zoom and filter, and then details-on-demand »- est une exploration visuelle de données obéissant à un processus en trois phases :

- Vue d'ensemble,
- Zoom et filtrage,
- Détails à la demande.

D'abord, l'utilisateur a besoin de se faire une idée de l'ensemble de données par vue d'ensemble. Il identifie par la suite des structures intéressantes et il se focalise sur une ou plusieurs d'entre elles. Enfin, pour analyser ces structures, l'utilisateur cherche à accéder au détail de données. [4]

3. Objectifs de la visualisation

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.
- Gilles Balmisse dit encore que recourir aux technologies de la visualisation a un double objectif :

  - Communiquer efficacement des informations au travers d'une représentation graphique via des cartes cognitives.
  - Faciliter la découverte de connaissances grâce à une représentation graphique issue de l'analyse d'un corpus d'informations via des cartes sémantiques. [4]

4. Type de visualisation

D'après le dictionnaire CNRTL[1], la visualisation est la présentation visuelle sur un écran des résultats d'un traitement sous forme alphanumérique ou graphique. Card et al (1999) quant à eux définissent le terme de visualisation par l'utilisation, assistée par l'ordinateur, des représentations visuelles de données pour amplifier la cognition. Ces définitions montrent que la visualisation est une activité cognitive facilitée par une représentation graphique externe pour aider les utilisateurs et les analystes de construire une représentation mentale interne sur le monde [5]. Il existe trois catégories de la visualisation qui sont les suivantes :

- *La visualisation scientifique :* permet de comprendre les phénomènes physiques dans les données [6] et qui se base sur des modèles mathématiques.
- *La visualisation d'information :* vise à explorer les données et les informations sous forme graphique qui permet aussi d'extraire et d'identifier les tendances, les corrélations et les structures abstraites dans les données.
- *La visualisation de connaissances :* est utilisée pour désigner tout procédé permettant de présenter une structure de connaissance comme moyen pour évaluer soi-même des connaissances et aider à la compréhension et à la navigation [7].

Dans notre cadre d'étude, on se concentre sur la visualisation d'information que nous allons développer dans ce qui suit.

[1] D'après le Centre National de Ressources Textuelles et Lexicales (CNRTL) : www.cnrtl.fr/definition/visualisation. [En ligne] consulté le 06/02/2013.

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.

5.2. Éléments du processus de visualisation [11]

➢ **Les données brutes :**

En 1996, Shneiderman a proposé une taxonomie pour les conceptions de visualisation d'information construite sur le type et la tâche des données (TTT). Il distingue sept types de données : unidimensionnelles, bidimensionnelles, tridimensionnelles, temporelles, multidimensionnelles, arbres et réseaux. Dans ses travaux, D.Keim introduit les logiciels et les algorithmes comme de nouveaux types de données qui pourraient être visualisé.

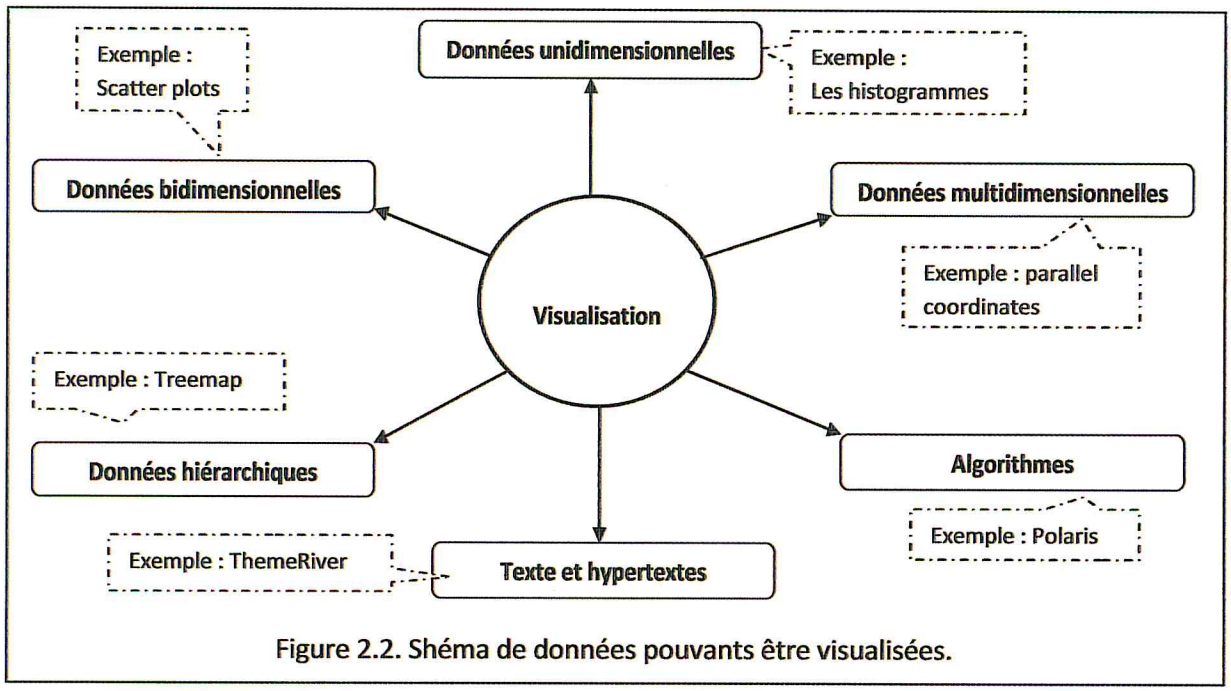


Figure 2.2. Shéma de données pouvants être visualisées.

- *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. [12]

5. Modèle de visualisation de l'information

Le processus de la visualisation de l'information est un sujet de recherche qui a été enrichie par les travaux de P.K Robertson et De Ferrari, 1994 ou encore Ed Chi qui a introduit le modèle « Data State Reference Model » en 2000, et aussi le travail de Card, Machinlay et Schneiderman, 1999 qui propose le modèle « Data Flow » [7], ce dernier est le modèle de référence le plus utilisé dans le domaine de la visualisation [11] car il est simple à mettre en œuvre et montre clairement les différentes étapes du processus depuis la transformation de données sources jusqu'à la visualisation.

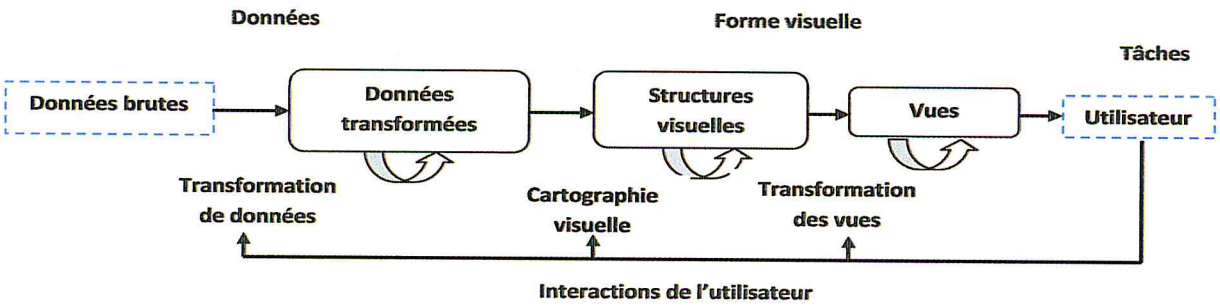


Figure 2.1. Processus de visualisation de l'information (Shneiderman) « Data Flow ».

5.1. Les étapes du processus de visualisation de l'information [11]

Transformation de données

Cette étape produit des données uniformes à partir de données sources brutes (structurées ou non structurées) et ceci après application de plusieurs opérations comme la sélections des attributs ou la réduction de dimensions, la projection, l'agrégation, l'échantillonnage, le résumé de données et la normalisation de données, et surtout quand les données sources sont multidimensionnelles.

La cartographie visuelle

La cartographie est l'étape de base du processus où les dimensions sont mises en correspondance avec les aspects visuels pour former des structures visuelles. Une des opérations les plus connues de cette étape est la génération d'ordre en attribuant les dimensions données à des axes de visualisation dans des ordres différents.

- *Données bidimensionnelles :* Les données bidimensionnelles ont deux dimensions distinctes. L'exemple typique de ce types de données est la représentation des 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. [12]

- *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. 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.[12]

- *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 [8].

- *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 le 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 [9].

- *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 » [10].

➢ **Les interactions de l'utilisateur**

La visualisation de l'information ne peut être traitée sans aborder l'interaction.

Cette dernière rend possible l'exploitation réelle des vues d'ensemble une fois produites. En effet, la perception est indissociable de l'action : c'est le couplage « action perception ». Ainsi l'être humain est plus habile à extraire des informations d'une interface s'il peut agir directement et activement sur cette interface que s'il reste passif. Daniel Keim (2002) distingue les qualificatifs « dynamiques » et « interactif » selon que les modifications apportées à la visualisation des données soient effectuées automatiquement ou manuellement (l'utilisateur final pouvant agir directement) :

- Projection dynamique : il s'agit de changer dynamiquement les projections afin d'explorer un ensemble de données multidimensionnelles.
- Filtrage interactif : il s'agit d'avoir, d'un part, la possibilité de diviser interactivement l'ensemble des données dans des segments et, d'autre part, de se concentrer sur les sous-ensembles intéressants. Ceci peut être fait en choisissant directement le sous-

ensemble désiré (browsing) ou en spécifiant des propriétés du sous-ensemble désiré (querying).

- Zoom interactif : il s'agit de partir d'une vue globale des données et de permettre l'affichage des détails selon différentes résolutions.
- Liens interactifs et brossage (interactive linking and brushing) pour les données multidimensionnelles : l'idée est de combiner des méthodes différentes de visualisation pour surmonter les imperfections des techniques simple. Les nuages de points (scatterplots) des projections différentes, par exemple, peuvent être combinés en colorant et en liant des sous-ensembles de points dans toutes les projections.[4]

Le tableau suivant propose une projection des interactions connues par rapport à l'interaction de l'utilisateur dans le processus de visualisation de l'information :

Étapes	Techniques
Transformation des données	Les requêtes dynamiques, Direct walk, Manipulation directe
Cartographie visuelle	Détails à la demande, Les lentilles magiques, Brushing
Transformation des vues	Mouvement de caméra, Le contrôle de point de vue

Tableau 2.1. Les différentes techniques selon les points d'interaction avec le processus [7]

➢ Techniques de visualisation

Dans la dernière décennie, un grand nombre de nouvelles techniques de visualisation d'informations ont été développés. D. Keim (2002) se concentre sur la conception de l'environnement visuel et propose une classification des techniques de visualisation qui prend en compte les développements récents en matière d'information de visualisation : l'affichage 2D/3D standard, affichage génétiquement transformé, les écrans à base d'icônes, les écrans à pixels denses et les écrans empilés. Nous nous intéresserons dans la suite de ce chapitre aux techniques de visualisation des données multidimensionnelles.

- Ben Shneiderman (1996) classifie la visualisation selon les « type des données » à représenter et les « tâches » possibles de l'utilisateur [11]
- La taxonomie de Daniel Keim (2002) qui s'appuie sur trois critères (techniques de visualisation, le type de données et le type d'interaction). [12]

- Wiss & Carr (1998) proposent une taxonomie selon trois aspects cognitifs : attention, abstraction et affordances [11]
- HURTER Christophe (2010) propose une taxonomie basé sur la représentation des données (valeur, relation, arborescente), la visualisation (Uniforme, non uniforme) et l'interaction. [7]

➢ **Tâches de l'utilisateur**

Selon Ben Shneiderman, les tâches interactives possibles par l'analyste sont :

- Avoir une vue de l'ensemble ou globale
- Zoomer
- Filtrer
- Détailler
- Voir les relations entre objets
- Avoir l'historique des actions pour le rejouer
- Extraire

6. Techniques de visualisation des données multidimensionnelles

Une visualisation est une représentation visuelle des données. Les données sont donc mappées en une certaine forme numérique et traduite en une représentation graphique.

Il existe de nombreuses visualisations et un bon nombre important de taxonomies (B.Shneiderman, 1996). Dans notre étude nous nous concentrons sur les tables de données multidimensionnelles (n dimensions avec $n > 3$). Les taxonomies suivantes concernent les représentations graphiques de données multidimensionnelles les plus connues :

- 2D and 3D Scatterplots (nuages de points à 2D et 3D)
- Matrix of scatterplots (matrice des nuages de points)
- Parallel coordinates
- Polar charts
- RadViz

Plusieurs techniques et des dérivées des techniques présentées existent, nous allons décrire dans la suite de ce chapitre les cinq techniques citées plus en haut.

6.1. Scatterplots (nuages de points)

Un nuage de points est une projection des données, en points, dans un espace deux ou trois dimensions. Représenté sur l'écran dans le format classique (X, Y) ou (X, Y, Z) (Figure 2.3), c'est la technique de visualisation de données multidimensionnelles la plus utilisée.

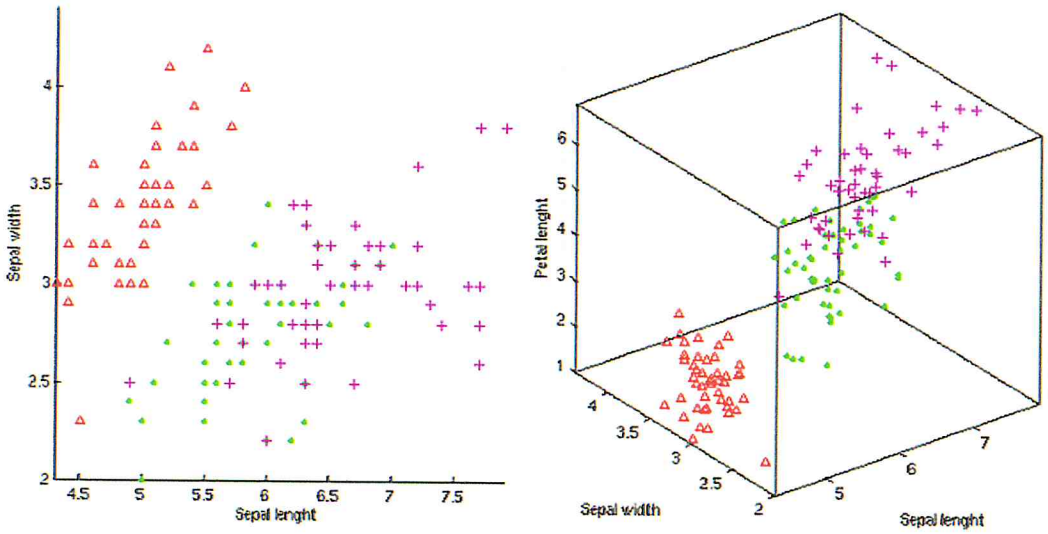


Figure 2.3. Nuages de points en 2D et 3D

De nombreuses applications ou transformations peuvent être appliquées sur les nuages de points. Les points affichés peuvent avoir de nombreux attributs tels que la couleur, la taille, la forme, la texture ou le mouvement. Pour interpréter l'interaction de projection 3D, il est nécessaire de résoudre les ambigüités, bien que d'autres techniques ont été utilisées (l'animation). En général, cette technique est liée à des écrans iconographiques et d'affichage de pixels.

6.2. Matrix of scatterplots (matrice de nuages de points)

Une matrice de nuages de points (Figure 2.4) est une matrice de nuages de points montrant toutes les paires de combinaisons de dimensions possibles. Pour les données de n dimensions, cela donne $\frac{n(n-1)}{2}$ nuages de points avec des échelles communes, bien que le plus souvent n^2 nuages de données sont affichés. Les nuages de points peuvent aussi être positionnés sous un format non ordonné (circulaire, hexagonal, etc.). On peut visuellement relier les traits d'un nuage de points avec des fonctionnalités d'un autre, ce qui augmente considérablement sa puissance.

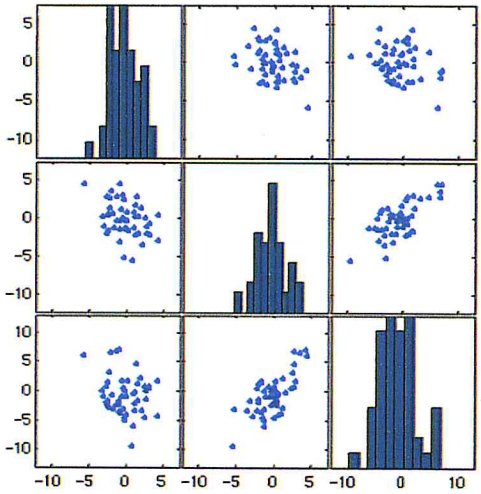


Figure 2.4. Matrice de nuages de points.

Cette technique a été utilisée bien avant sa publication [13] et [14].

Plusieurs variantes de la matrice de nuages de points ont depuis été développées : hyperslice [15], N-vision [16], hyperbox [17] pour n'en nommer que quelques-uns.

- Hyperslice est une matrice de panneaux où des tranches (slices) de fonctions multi variées sont présentées à un certain point spécifique.
- N-vision est similaire l'hyperslice, où la matrice est adaptée à l'exploration interactive des fonctions multi variées.

Hyperbox utilise la projection des paires identiques de données mais les projette sur des panneaux d'une boite à n dimensions. Chacun des panneaux possède une orientation différente, et les dimensions peuvent être coupées afin de montrer les histogrammes sur les panneaux selon la plage de la dimension ayant été coupée.

6.3. Parallel coordinates

Les parallel coordinates (Figure 2.5) utilisent des axes parallèles pour représenter les dimensions des données multidimensionnelles [18], [19]. Une ligne verticale est utilisée pour la projection de chacune des dimensions ou des attributs, les valeurs maximales et minimales de chacune des dimensions sont généralement mises à l'échelle à des limites supérieures et inferieures sur les lignes verticales. Une polyligne est composée de n-1 lignes à des valeurs dimensionnelles appropriées relie les axes pour représenter un point de n dimensions.

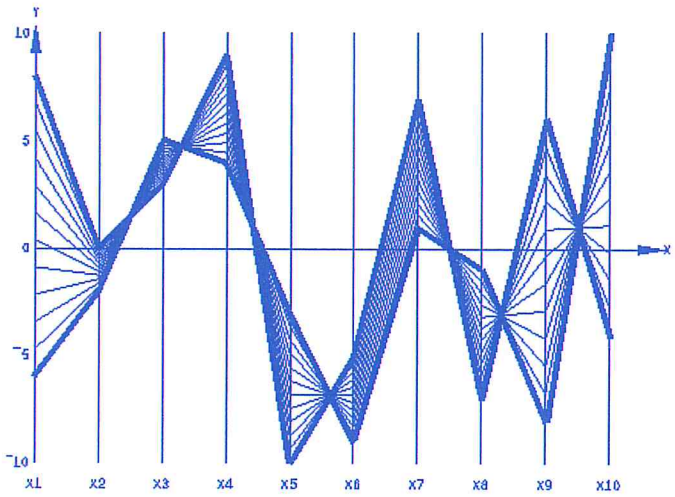


Figure 2.5. Parallel coordinates représentant des données multidimensionnelles.

6.4. Polar charts [20]

Le polar charts (Figure 2.6) est un graphique circulaire pour le tracé de coordonnées polaires. Ces coordonnées polaires mappent les données sur une surface 2D à l'aide de l'angle et le rayon, créant ainsi une version dite « wrap-around » littéralement « enrouler autour » du graphique linéaire. Le polar charts vient combler les limites du graphique linéaire, qui sont utilisés généralement pour représenter des valeurs uniques, ou par morceaux continus d'une dimension. Ce graphique peut être considéré comme la forme circulaire du parallel coordinates et peuvent donc réduire l'effet limitatif d'un grand nombre de dimensions. Toutefois, la taille des représentations des points de données dépend de la proximité du centre.

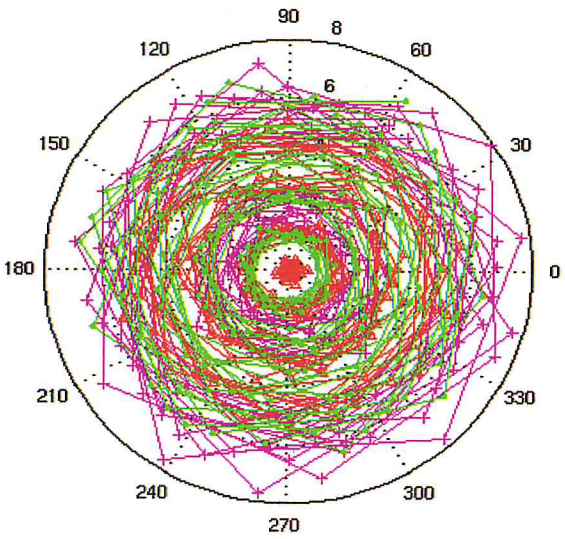


Figure 2.6. Polar charts.

6.5. RadViz

RadViz (Figure 2.7) est une technique d'affichage qui place les ancres de dimensions de données autour du périmètre d'un cercle [21]. Des constantes « élastiques » sont utilisées pour représenter des valeurs relationnelles entre les points de données, une extrémité de l'élastique est fixée à une ancre dimensionnelle, l'autre est fixée à un point de données. Les valeurs de chaque dimension sont généralement normalisées de 0 à 1. Chaque point de données est affiché à l'endroit où la somme de tous les élastiques est égale à zéro. La position d'un point de données dépend en grande partie de l'agencement des dimensions autour du cercle.

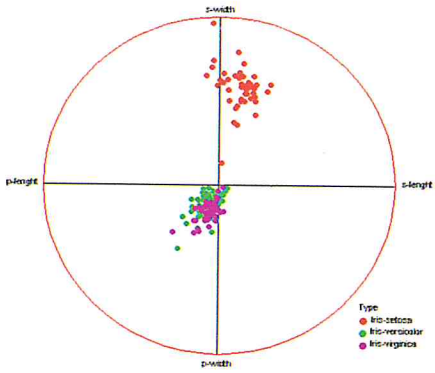


Figure 2.7. RadViz de points de données multidimensionnelles.

L'exploration les résultats de clustering de grands ensembles de données est un problème important mais difficile, les techniques de visualisation de l'information peuvent aider à résoudre ce problème. La prochaine étape de notre travail consiste à intégrer la technique de visualisation à la technique du regroupement choisi dans le but de faciliter la lecture de ses résultats.

7. Systèmes existants

Dans le domaine de la visualisation de données, il existe une multitude d'outils qui proposent de nombreuses techniques de visualisations à l'utilisateur. Dans bon nombre de ces systèmes existants, le processus commence par l'importation d'une table de données, suivie de la visualisation, l'étape du clustering n'est présente que dans des outils spécifiques aux études de clustering. Parmi les systèmes existants nous décrivons brièvement les suivants :

7.1. Spotfire

Par la société TIBCO, est une plate-forme logicielle qui permet aux clients d'analyser les données à l'aide de graphiques. Spotfire traite les données qui concernent le domaine du business intelligence. La dernière version (4.0) inclut l'intégration avec les « services

statistiques » et la capacité de développer des applications d'analyse dynamique qui s'exécutent sur le web à travers un client appelé TIBCO Spotfire Web Player.

7.2. Orange

Est un outil de visualisation de données open source, dédié aux experts comme aux non initiés. Il permet l'exploration de données grâce à la programmation visuelle ou par script Python. Il dispose par ailleurs de modules pour la bioinformatique et le text mining. Orange est aussi doté de fonctions pour l'analyse des données. [29]

7.3. XmdvTool

Est un logiciel du domaine public pour l'exploration visuelle interactive d'ensemble de données multidimensionnelles. Il est disponible sur toutes les grandes plates-formes. XmdvTool est développé en utilisant Qt et Eclipse CDT. Il prend en charge cinq méthodes pour afficher des données de forme plate et de données hiérarchiques en cluster (scatterplots, star glyphs, parallel coordinates, dimensional stacking, pixel-oriented display). [30]

7.4. GGobi

Est un programme de visualisation open source pour l'exploration de données multidimensionnelles. Il fournit des graphiques dynamiques et interactifs comme les tours, ainsi que des graphiques plus familiers tels que le nuage de points, barchart et le parallel coordinates. GGobi est entièrement documenté dans le livre « Interactive and Dynamic Graphics for Data Analysis ». [31]

8. Conclusion

Nous avons introduit dans ce chapitre les objectifs ainsi que le modèle de visualisation de l'information en s'appuyant sur des techniques de visualisation connues et utilisées dans de nombreux outils graphiques. L'idée maintenant est de projeter les résultats de clustering, multidimensionnelles, sur une des techniques qu'on a choisi : le parallel coordinates en raison de son concept basique et facile à interpréter. Dans le chapitre suivant, nous allons nous pencher sur le couplage entre clustering et visualisation de ses résultats.

Chapitre III

Clustering visuel

1. Introduction

L'analyse automatique appartient à une discipline issue d'une longue tradition et ayant de solides fondations théoriques. L'analyse automatique de données n'est pas centrée sur un seul domaine d'application et de ce fait, elle concerne des méthodologies générales. Cette analyse est faite sur les résultats de clustering automatique fait sans aucune information à priori. Cependant, les méthodes de clustering deviennent des « boites noires » entre les mains de l'utilisateur et la validation et l'interprétation des résultats deviennent très difficiles. Autre problème, les algorithmes produisent des résultats qui ne sont pas en réalité des solutions pertinentes car ils ne tiennent pas compte des connaissances d'experts spécialisés [13].

A toutes ces problématiques s'ajoutent les données massives sur les quelles s'exécutent les méthodes du clustering automatique et les résultats qui sont générés sont par conséquent massives eux aussi ce qui joue un rôle négatif dans la validation des clusters finaux pour toutes ces raisons l'analyse des données multidimensionnelles devient très importantes d'un coté, et très difficile d'un autre. Les techniques de visualisation de l'information contribueront à résoudre le problème.

Dans ce contexte, notre objectif consiste à améliorer le processus d'analyses exploratoire des données via l'utilisation des technologies de visualisation d'informations et des techniques de clustering pour une meilleure analyse de données.

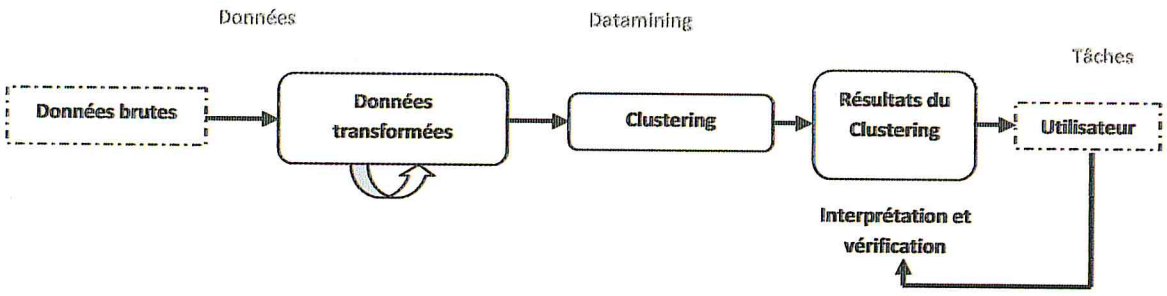


Figure 3.1. Processus du clustering de données et interprétation des résultats.

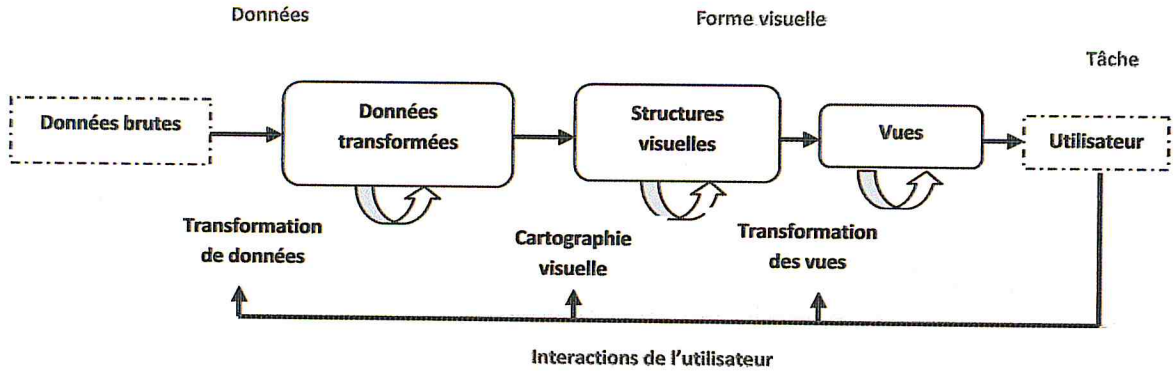


Figure 3.2. Processus de la visualisation de données.

2. Définition du clustering visuel

Le Clustering visuel de données est l'utilisation de techniques de la visualisation de données pour que les analystes puissent évaluer, surveiller et guider les entrées, les sorties et les processus de fouille de données [14].

Pour que l'exploration de données soit efficace, Ben Shneiderman introduit une démarche pour l'exploration visuelle basée sur des opérations de la recherche d'information qui sont : « Overview first, zoom and filter, details on demand » (1996). D.Keim, en 2006, a modifié cette démarche pour la spécialiser comme suit: « Analyse first, show the important, zoom, filter and analyse further, details on demand ». [7]

3. Processus de clustering visuel des données

L'analyse visuelle des résultats du clustering est un processus d'analyse visuelle qui combine des méthodes d'analyse automatiques (clustering) et la visualisation d'informations avec un couplage étroit à travers l'interaction humaine dans le but d'acquérir des connaissances à partir de données, la figure 3.3 présente un aperçu des étapes du processus de l'analyse visuelle du clustering.

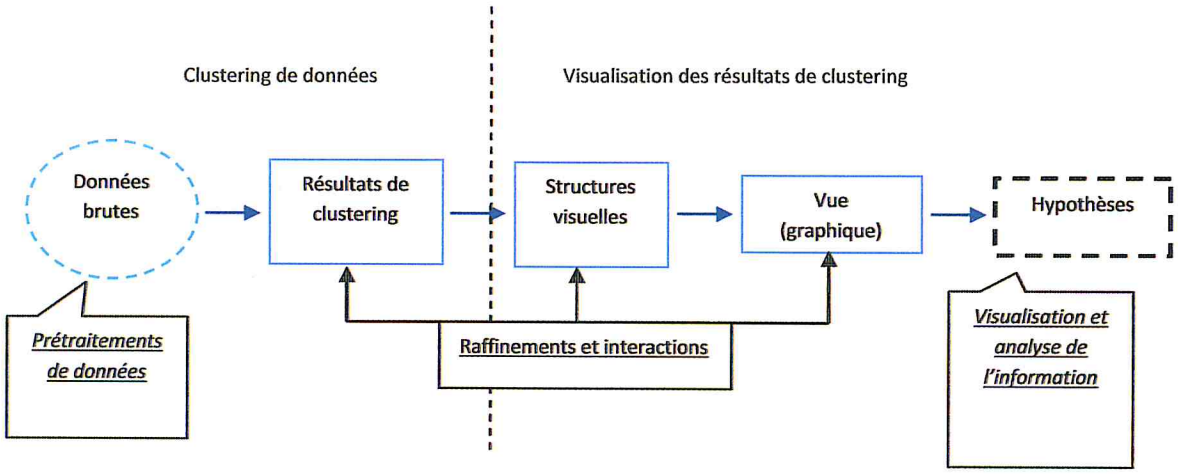


Figure 3.3. Processus de la visualisation des résultats de clustering de données [22]

Généralement, dans la majorité des applications existantes, les données hétérogènes et multidimensionnelles doivent être intégrées avant l'application de la visualisation ou des méthodes d'analyse automatique. Par conséquent, le plus souvent la première étape du processus consiste à prétraiter et transformer les données pour représenter les données de différentes manières afin d'aboutir à une exploration plus riche (comme indiqué dans la figure 3.3 : *prétraitement de données*). D'autres tâches appartiennent à l'étape de prétraitement de données et qui sont le nettoyage de données, la transformation de données, le regroupement de sources de données. Après la transformation de données, les méthodes de clustering sont appliquées pour générer des regroupements de données selon des critères spécifiés par l'utilisateur (choix du K pour l'algorithme du k-means, attributions de noms de partitions selon les cas, etc.) cette étape (clustering de données) fournit des informations supplémentaires aux données originales, on obtient alors un modèle ayant des données regroupées dans des clusters, les données multidimensionnelles se voient enrichies d'une autre information, à ce stade du processus, la visualisation du modèle vient alors pour évaluer les résultats du modèle généré. Le processus de l'analyse visuelle, et en plus de se placer entre les deux étapes : le clustering de données et la visualisation des résultats de ce dernier, vient apporter à l'utilisateur la possibilité d'améliorer continuellement tantôt le regroupement de données par l'interaction sur les paramètres de l'algorithme de clustering, tantôt sur le graphe représentant le résultat du clustering par interaction graphique (zoom, filtrage de dimensions, etc.) ces étapes de raffinement sont très importantes et il est recommandé à

l'utilisateur de les appliquer afin d'avoir de meilleurs résultats et d'éviter des résultats trompeurs [22].

Notre processus d'analyse visuelle vise à coupler étroitement les méthodes de clustering et les représentations visuelles interactives. D. Keim (2006) a présenté un guide selon lequel il est décrit la manière selon laquelle les données doivent être représentées à l'écran « Analyser en premier, afficher le plus important, zoomer, filtrer pour analyser de nouveau, détails à la demande » [7]. Dans le cadre de la représentation visuelle de résultats de clustering appartenant à la classification non supervisée, nous nous somme intéressé au parallel coordinates comme technique de visualisation.

4. Parallel coordinates

Les graphiques des coordonnées parallèles ont été inventés par Maurice d'Ocagne en 1885 [24]. Ils ont été redécouverts par Alfred Inselberg en 1959, utilisés depuis dans des systèmes de visualisations de données multidimensionnelles et connus sous le nom de parallel coordinates, il n'est pas rare de lire « diagrammes d'Inselberg » en référence au parallel coordinates. Le graphe du parallel coordinates est un dispositif géométrique qui affiche des points dans un espace de grande dimension, spécialement pour des données de plus de trois dimensions, ce type de graphe est une alternative aux « conventionnels » nuages de points. La représentation du parallel coordinates bénéficie des propriétés de dualité avec l'habituelle représentation cartésienne orthogonale. [26].

Son principe est simple, il utilise des axes parallèles équidistants (généralement à la verticale) représentant chacun une dimension l'ensemble de données et projette l'ensemble de données multidimensionnelles sur une surface à deux dimensions [27]. Les axes représentants les dimensions sont linéairement sur une échelle allant de la minimale à la valeur maximale de la dimension correspondante. De ce fait, chaque élément est représenté par une ligne polygonale coupant chaque axe à la valeur de la dimension correspondante [25].

L'intérêt de cette technique est immédiat, car il permet de visualiser simultanément un nombre de dimensions important. Il dessine des modèles des tendances et des corrélations. D. Keim a proposé une application sur les données boursières, ce qui permet de voir les tendances ou les anomalies. Le but de ce graphique est essentiellement exploratoire : il a pour but de faire découvrir un phénomène qui est difficilement détectable [4].

A ce stade de notre travail, nous avons défini le processus de la visualisation des résultats du clustering des données multidimensionnelles projetées sur un graphique interactif est la solution à notre problématique de ce fait nous allons proposer la description de notre analyse des besoins, ainsi que la conception de notre outil visuel.

Cette description est basée sur la structuration suivante :

- *Analyse des besoins :* dans laquelle nous présenterons la méthode ainsi que le cycle de vie de notre logiciel.
- *La conception :* dans cette partie nous décrirons les différents diagrammes utilisés pour la conception de notre outil.

Notre travail est décrit selon le langage de modélisation UML (Unified Modeling Language), suivant le processus UP (Unified Process).

5. Le langage de modélisation UML

UML se défini comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. [31]

UML est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration des modèles. Qualifier UML de « méthode objet » n'est donc pas tout à fait approprié. [32] pour cette raison, nous avons adopté le processus UP pour notre démarche.

6. Processus unifié (UP)

Le Processus Unifié est un processus de développement logiciel « itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques » :

- Itératif et incrémental : le projet est découpé en itérations de courte durée qui aident à mieux suivre l'avancement global. A la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.

- Centré sur l'architecture : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- Piloté par les risques : les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- Conduit par les cas d'utilisation : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés. [31]

7. Le cycle de vie

Le cycle de vie d'un logiciel est un ensemble séquentiel de phases, dont le nom et le nombre sont déterminés en fonction des besoins du projet, permettant généralement le développement d'un service ou d'un produit, en ce qui concerne notre projet nous avons suivi le modèle en cascade.

7.1. Modèle en cascade

Ce modèle est constitué d'une suite d'étapes qui ont pour but de réaliser un produit logiciel fini et testé. Le résultat de chaque étape est testé et on ne passe à l'étape suivante que lorsque l'étape actuelle est satisfaisante. Ce modèle est un cycle de vie linéaire, séquentiel, défini dans les années 70.

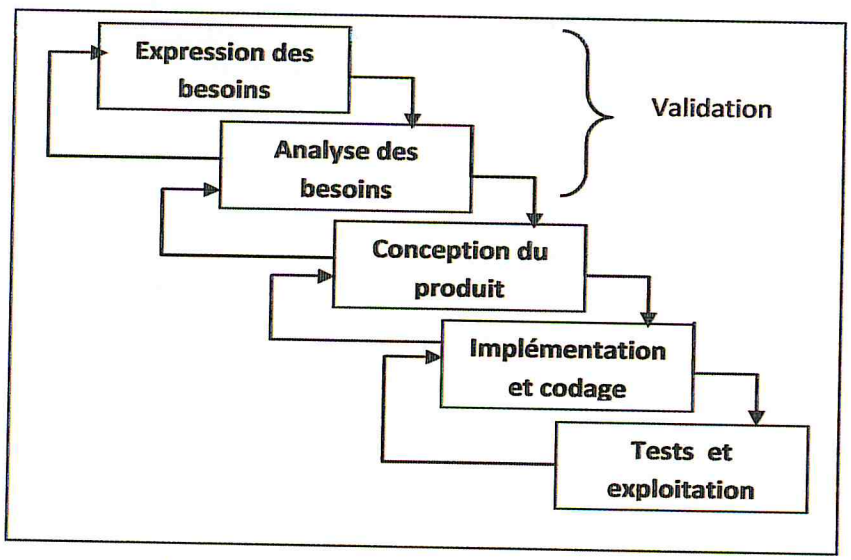


Figure 3.4. Cycle de vie selon le modèle en cascade.

7.1.1. Validation

A ce niveau, on parle de faisabilité, cette étape est représentée par les diagrammes suivants :

- Les diagrammes des cas d'utilisations
- Les diagrammes de séquence

Si la conception de ces diagrammes est faisable et validée on peut passer la seconde phase du modèle en cascade.

7.1.1.A. Diagramme des cas d'utilisations

- *Les acteurs*

L'acteur est, par définition le rôle joué par une personne qui interagit avec le système. Il est en principe extérieur au système, délimité par ses bornes.

L'acteur a un nom, qui le définit, ou qui précise son rôle dans la transaction décrite. Notre système est composé de deux types d'acteurs :

- *Utilisateur simple :* possède de simples connaissances dans le domaine de la visualisation des résultats du clustering, ou dans le type de distance etc. Il peut tout de même interagir avec notre outil graphique et l'exploiter au niveau qu'il souhaite.

- *L'expert :* peut utiliser notre système par des requêtes ciblées et spécifiques comme le choix de la distance, le choix des dimensions les plus stratégiques pour la visualisation finale etc.

- Les cas d'utilisations

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

Nous allons présenter notre diagramme des cas d'utilisations global, suivi par chaque cas d'utilisation détaillé.

➢ Cas d'utilisation global

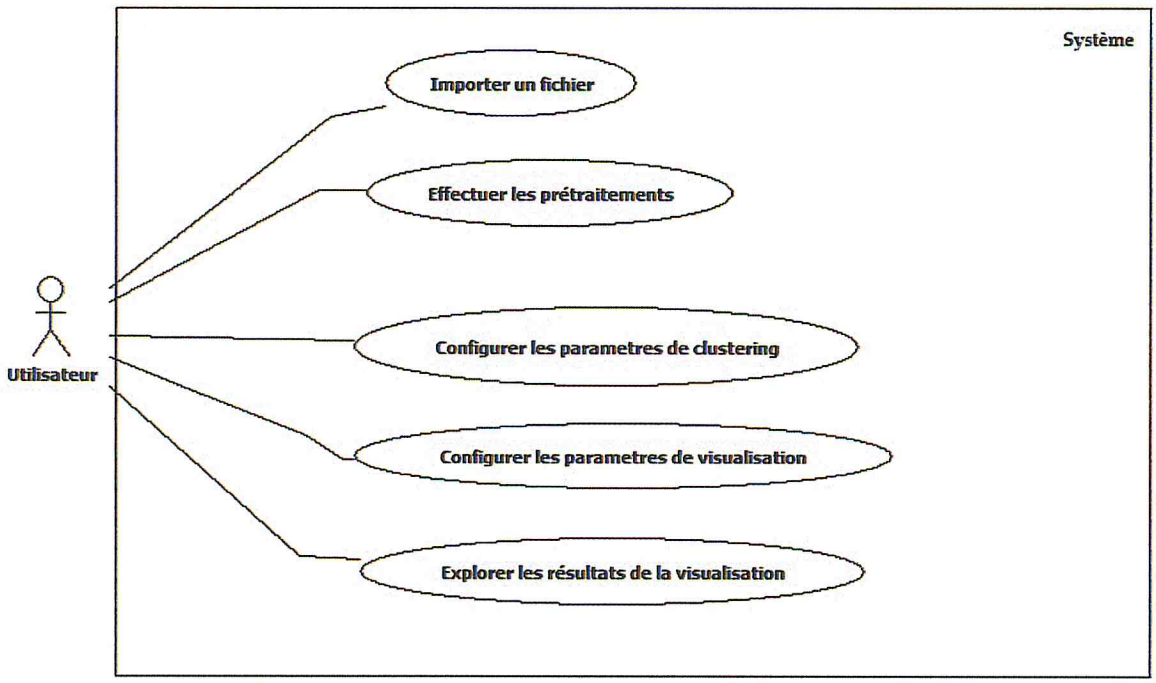


Figure 3.5. Cas d'utilisation global.

➢ Description du cas d'utilisation global :

Cas d'utilisation	Description
Importer un fichier	L'utilisateur spécifie le fichier à visualiser
Effectuer les prétraitements	L'utilisateur procède aux prétraitements nécessaires
Configurer les paramètres de clustering	L'utilisateur introduit les paramètres de clustering, de la distance et de l'ensemble de données
Configurer les paramètres de visualisation	L'utilisateur configure les paramètres concernant l'aspect visuel
Explorer visuellement les résultats du clustering	L'utilisateur interagit sur les résultats visuels

Tableau 3.1. Description du cas d'utilisation global.

➢ Détails des cas d'utilisation

- **Importation de fichier**

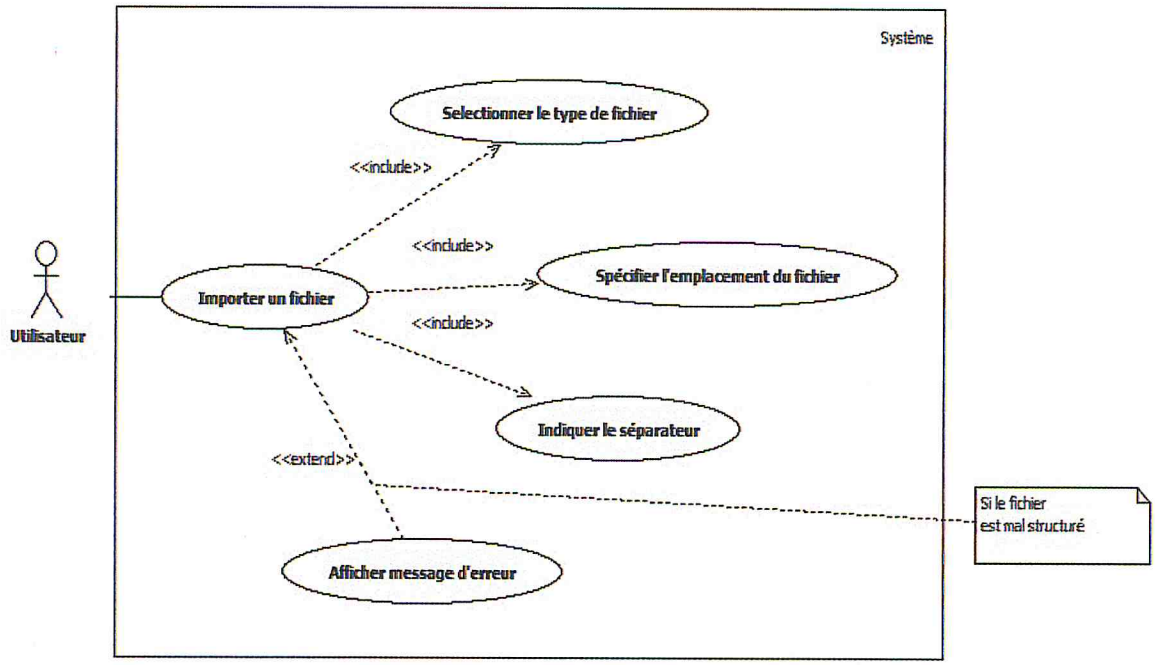


Figure 3.6. Cas d'utilisation de l'importation de fichier.

- **Description du cas d'utilisation de l'importation de fichier**

cas d'utilisation	Description
Importer un fichier	L'utilisateur lance le processus d'importation de fichier
Sélectionner le type de fichier	L'utilisateur spécifie dans un ensemble de types de fichiers l'extension du fichier voulu
Spécifier l'emplacement du fichier	Ici l'utilisateur sélectionne le fichier à visualiser
Indiquer le séparateur	L'utilisateur introduit le séparateur utilisé dans le fichier source pour séparer les dimensions
Afficher un message d'erreur	Dans le cas ou l'utilisateur sélectionne un fichier endommagé ou si le séparateur n'est pas le bon

Tableau 3.2. Description du cas d'utilisation de l'importation de fichier.

- **Prétraitements de données**

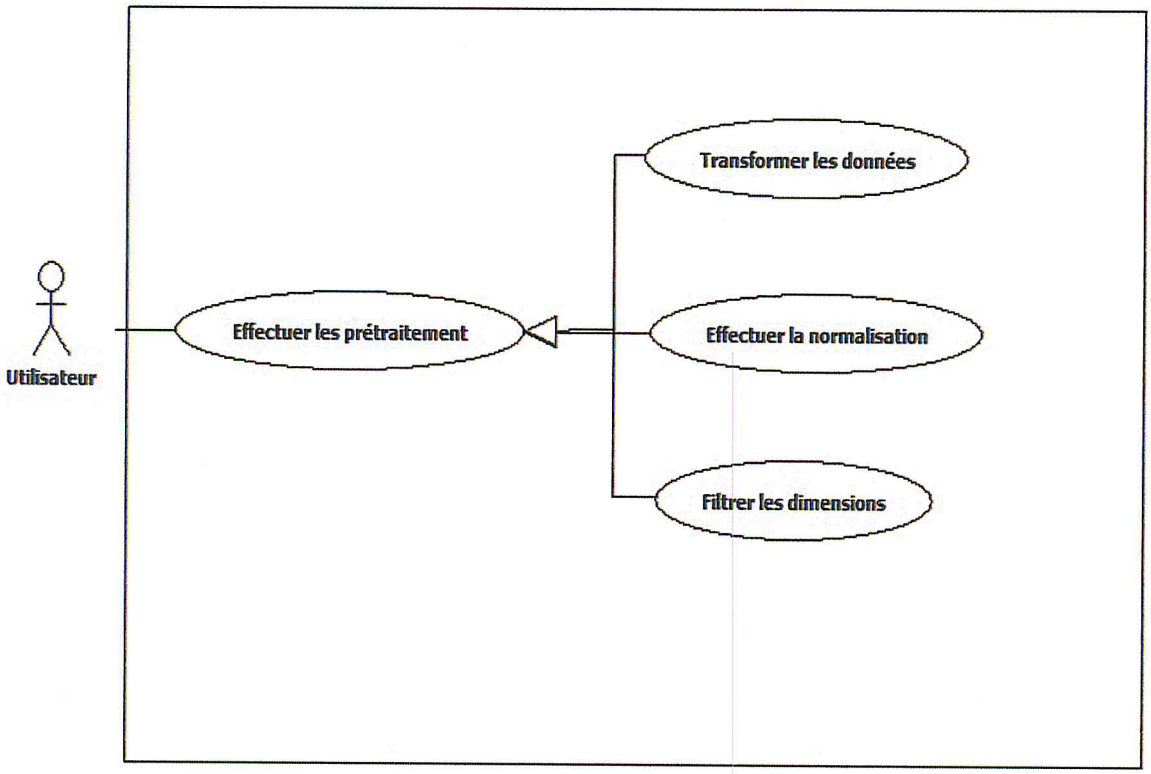


Figure 3.7. Cas d'utilisation des prétraitements de données.

- **Description du cas d'utilisation des prétraitements de données**

Cas d'utilisation	Description
Effectuer les prétraitements	Ici l'utilisateur choisi d'effectuer les prétraitements nécessaires sur les données afin de pouvoir effectuer les étapes suivantes
Transformer les données	L'utilisateur sélectionne des dimensions et effectue des transformations de types
Effectuer la normalisation	L'utilisateur sélectionne les dimensions et les normalise
Filtrer les dimensions	Ici l'utilisateur peut sélectionner qu'une partie des dimensions qui visualisera

Tableau 3.3. Description du cas d'utilisation des prétraitements de données.

- **Configurer les paramètres de clustering**

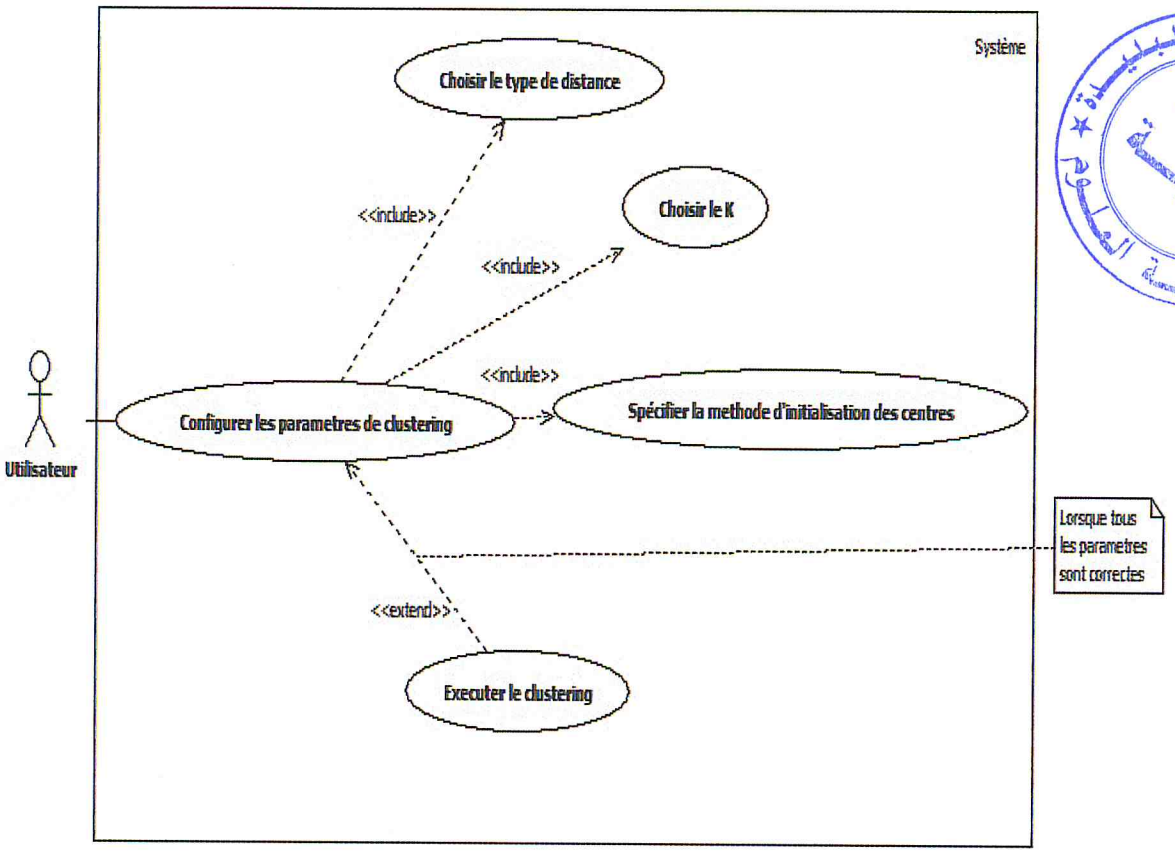


Figure 3.8. Cas d'utilisation configurer les paramètres de clustering.

- **Description du cas d'utilisation des configurations des paramètres de clustering**

Cas d'utilisation	Description
Configurer les paramètres de clustering	L'utilisateur doit paramétrer toutes les ressources nécessaires au clustering
Choisir le type de distance	Ici l'utilisateur choisi dans une liste une distance souhaitée
Choisir le K	L'utilisateur introduit le nombre des K clusters
Spécifier la méthode d'initialisation des centres	Une méthode d'initialisation des centres est paramétrée par l'utilisateur
Exécuter le clustering	L'utilisateur lance l'exécution de l'algorithme K means

Tableau 3.4. Description du cas d'utilisation des configurations des paramètres de clustering.

- **Configurer les paramètres de visualisation**

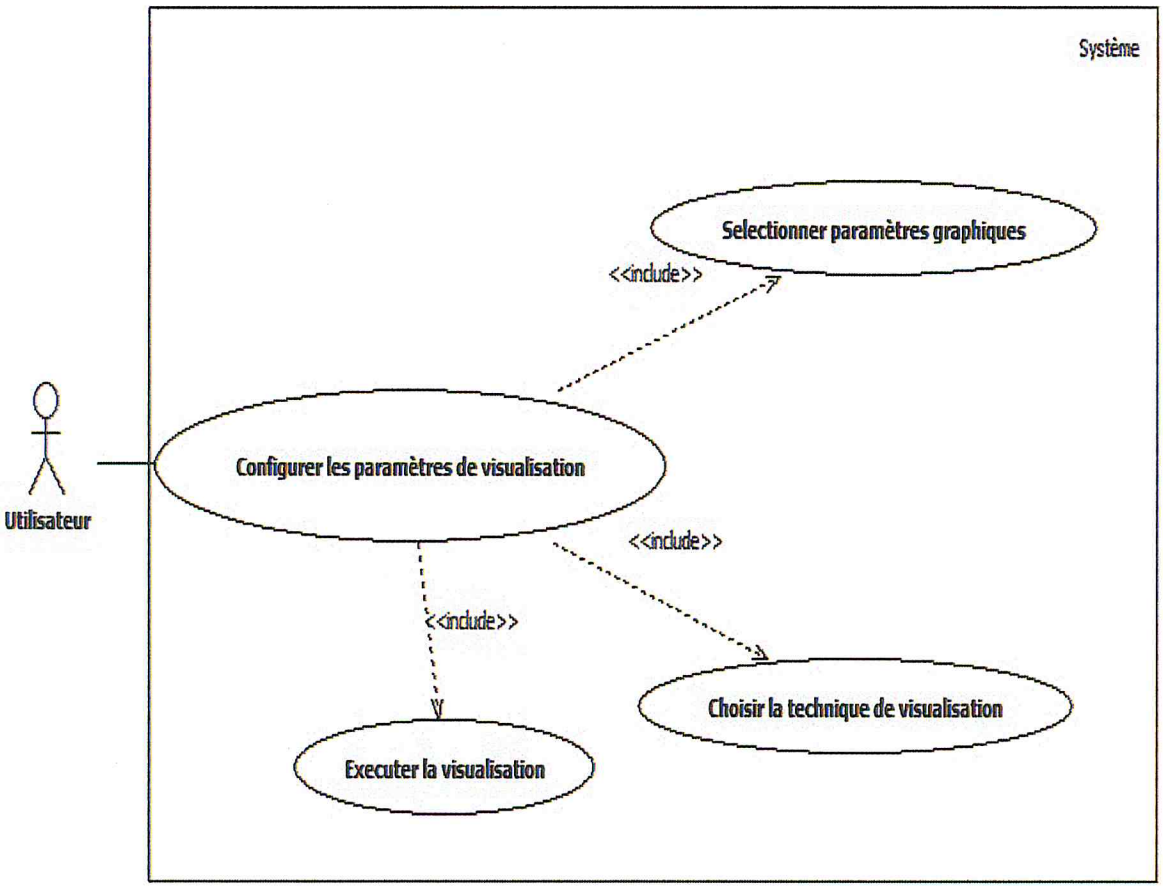


Figure 3.9. Cas d'utilisation configurer les paramètres de visualisation.

- **Description du cas d'utilisation configurer les paramètres de visualisation**

Cas d'utilisation	Description
Configurer les paramètres de visualisation	L'utilisateur doit paramétrer toutes les ressources nécessaires au graphe
Sélectionner les paramètres graphiques	L'utilisateur sélectionne le type de graphe
Choisir la technique de visualisation	Ici l'utilisateur choisi dans une liste de techniques proposées celle qu'il souhaite
Exécuter la visualisation	L'utilisateur lance le processus de visualisation

Tableau 3.5. Description du cas d'utilisation configurer les paramètres de visualisation.

- **Explorer les résultats de la visualisation**

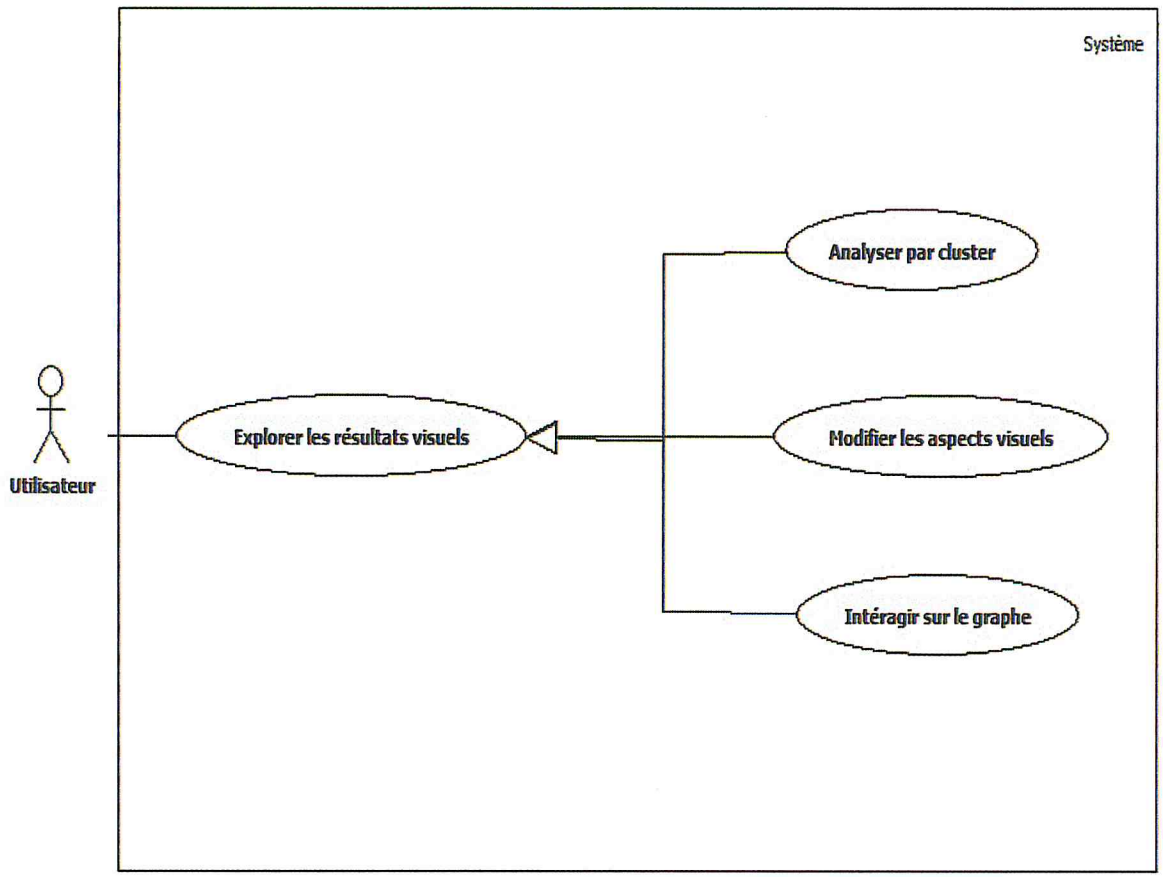


Figure 3.10. Cas d'utilisation explorer les résultats de la visualisation.

- **Description du cas d'utilisation de l'exploration des résultats de la visualisation**

Cas d'utilisation	Description
Explorer les résultats visuels	L'utilisateur analyse les résultats visuels grâce à l'interface interactive
Analyser par cluster	L'utilisateur choisi le/les clusters à visualiser
Modifier les aspects visuels	L'utilisateur change les composants visuels pour une meilleure analyse
Interagir sur le graphe	L'utilisateur interagit en exécutant les fonctions d'interactions proposées

Tableau 3.6. Description du cas d'utilisation de l'exploration des résultats de la visualisation.

6.1.1.B. Diagrammes de séquences

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Ainsi le temps est représenté explicitement par une dimension (la dimension verticale) et s'écoule de haut en bas. Nous présentons les différentes interactions avec notre système dans les diagrammes de séquences suivants

- **Diagramme de séquence de l'importation de fichier**

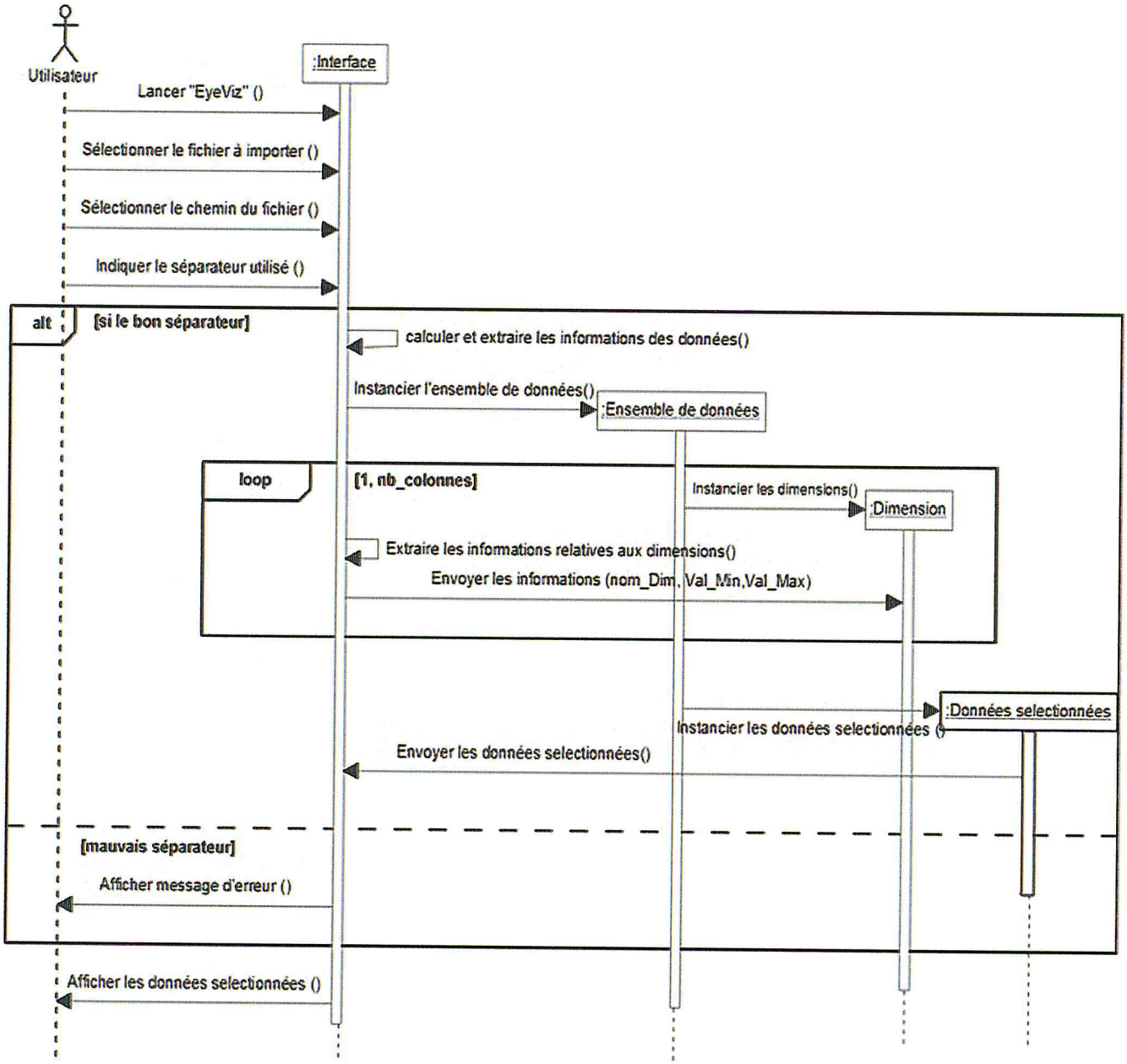


Figure 3.11. Diagramme de séquence de l'importation de fichier

- **Diagramme de séquence du prétraitement de données**

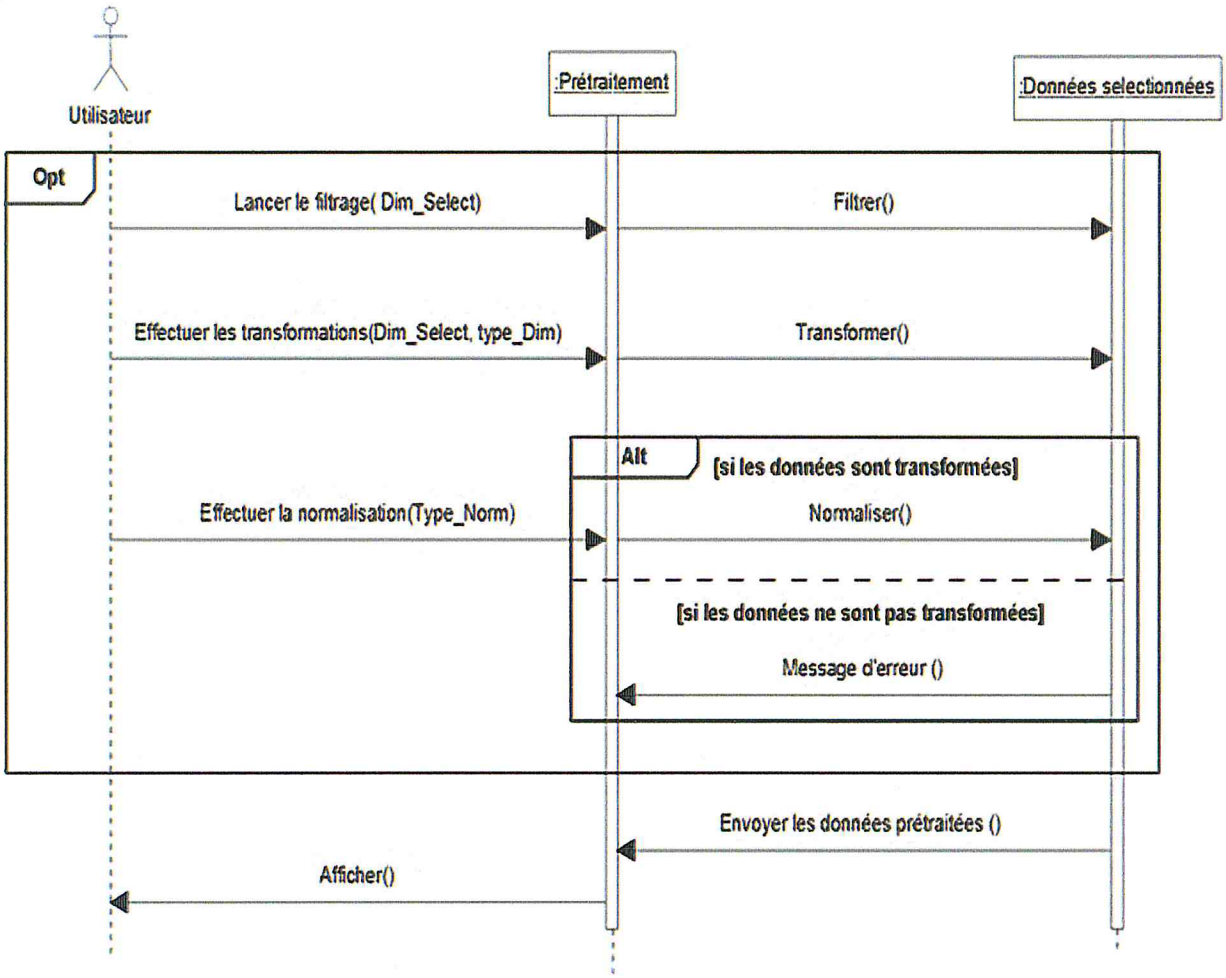


Figure 3.12. Diagramme de séquence des prétraitements de données

Le prétraitement de données est optionnel, l'utilisateur exécute les étapes du prétraitement indiquées dans la figure 3.12 dans le cas de données ayant des valeurs de types hétérogènes au départ. Toutes fois si l'ensemble de données importé présente des données prétraitées et normalisées, ces étapes n'auront plus d'utilité et les étapes de clustering peuvent débuter directement après l'importation de fichier.

- **Diagramme de séquence des paramètres de clustering**

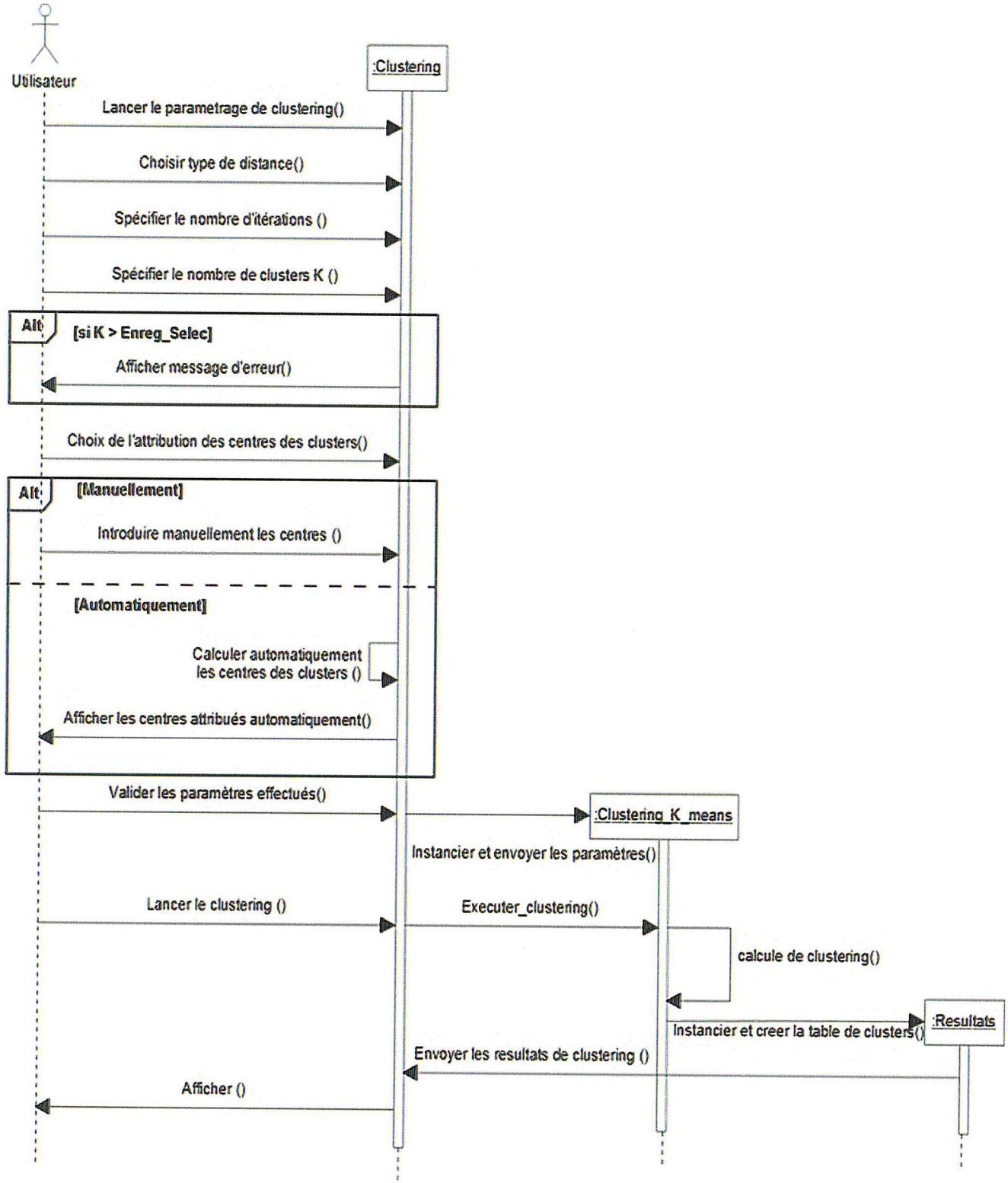


Figure 3.13. Diagramme de séquence du paramètre du clustering

- **Diagramme de séquence des paramètres de la visualisation**

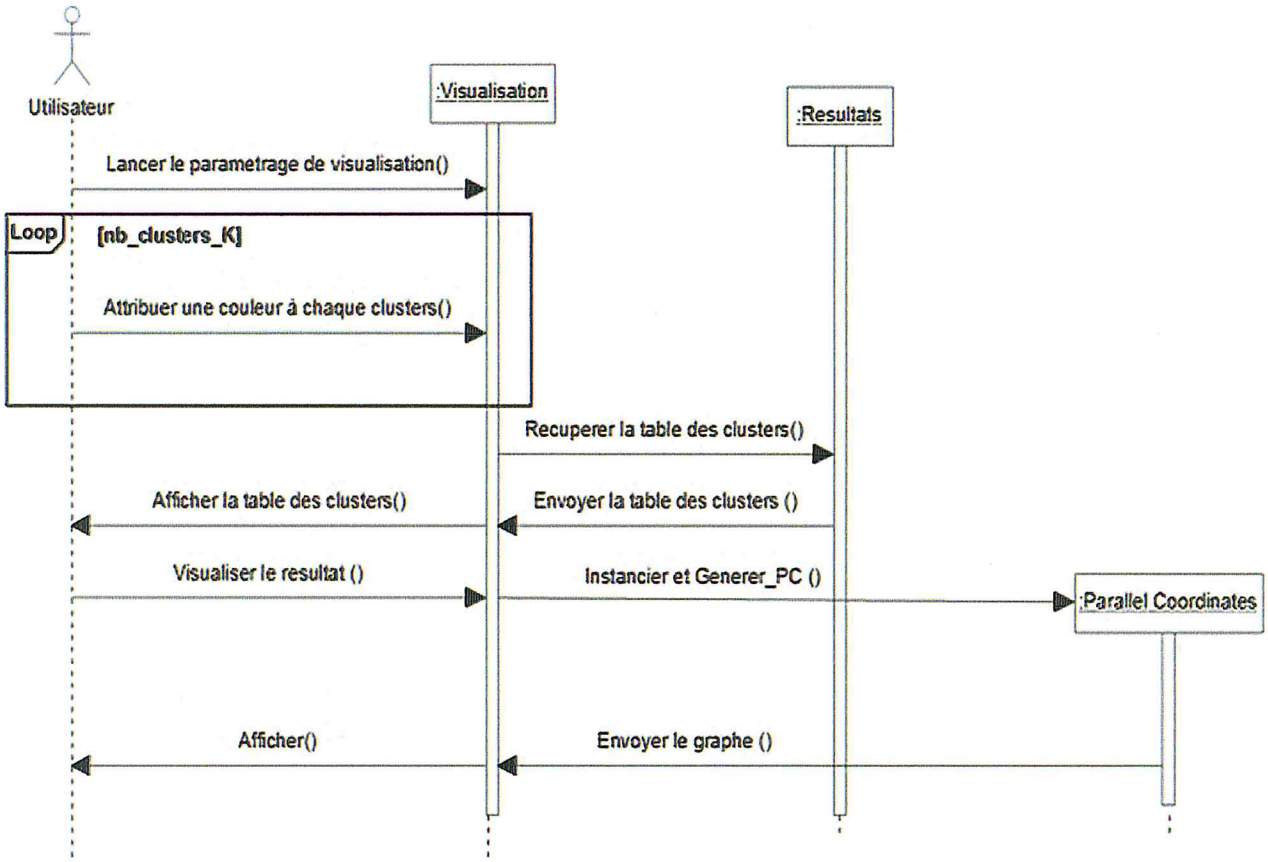


Figure 3.14. Diagramme de séquence des paramètres de la visualisation

- **Diagramme de séquence de l'exploration et des interactions possibles**

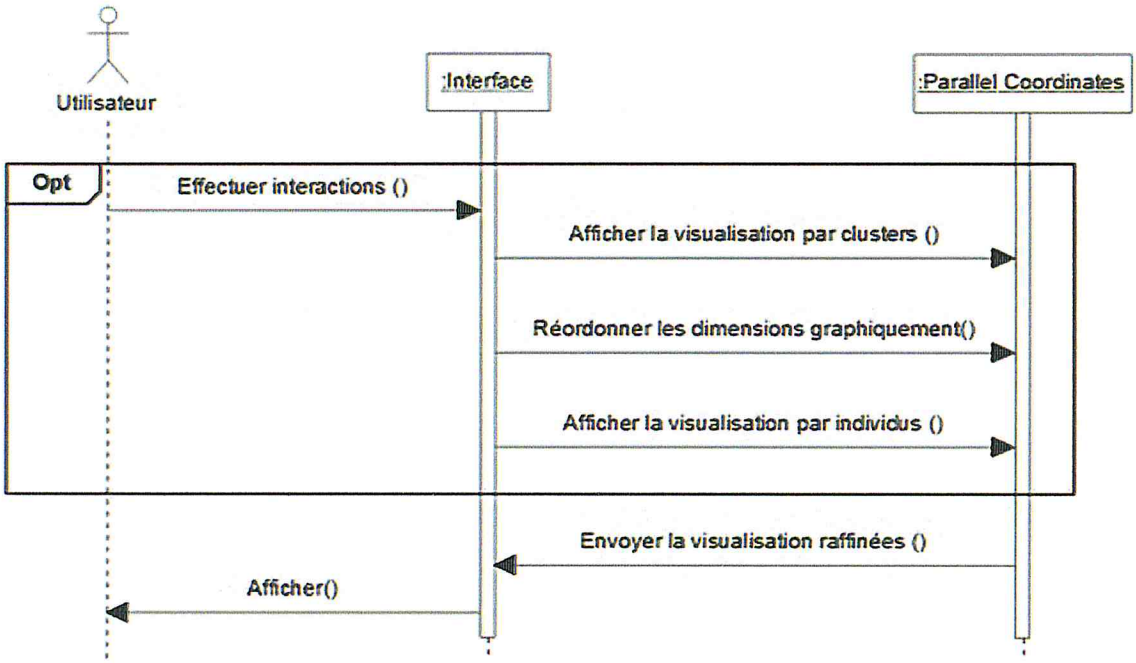


Figure 3.15. Diagramme de séquence de l'exploration visuelle

7.1.2 La conception du produit

Dans le cycle de vie d'un produit logiciel, la partie conception est décrite par le diagramme de classes UML. Dans cette partie du chapitre nous allons donc présenter notre diagramme de classes ainsi que les détails de chaque classe le constituant.

Le diagramme de classes

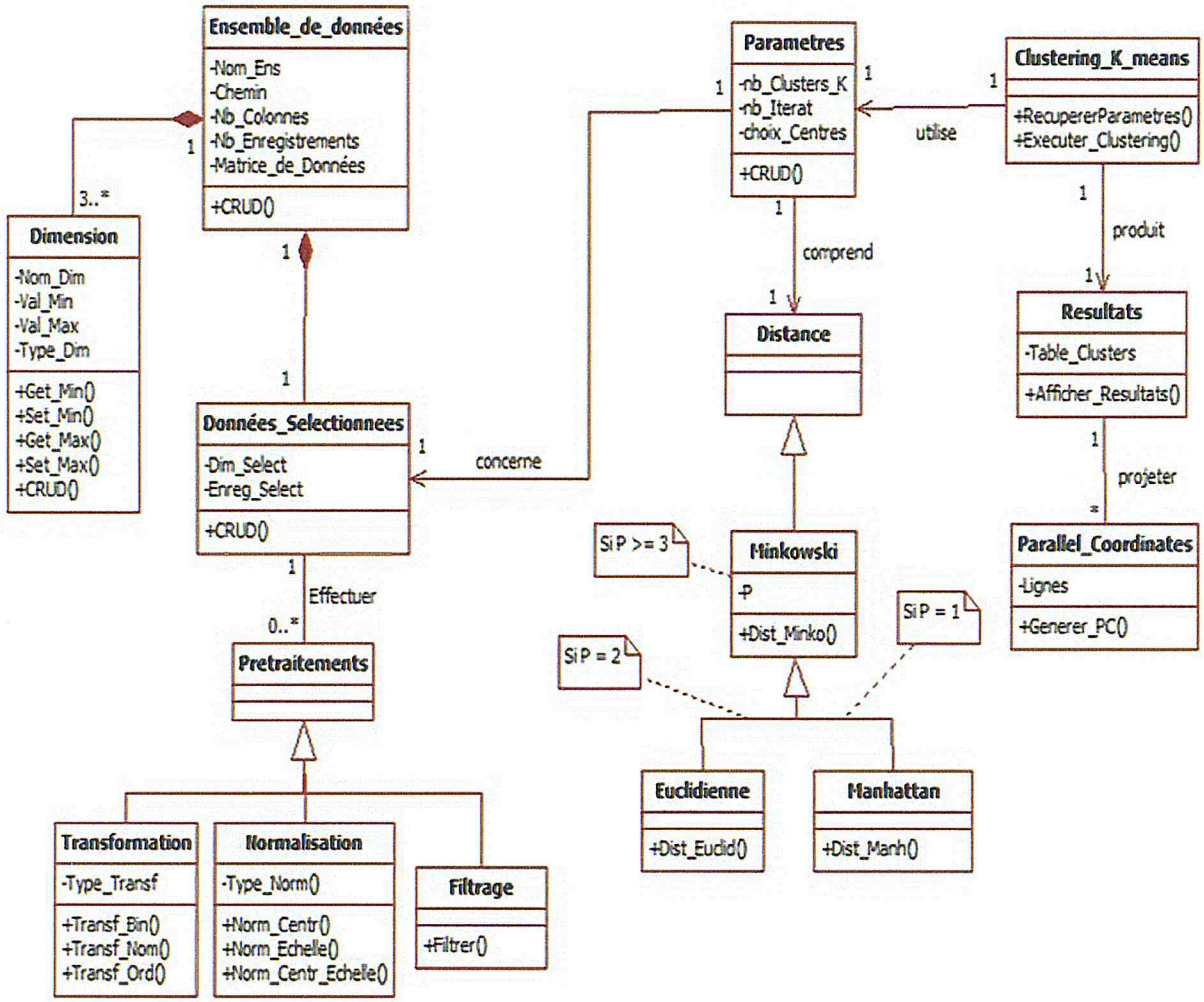


Figure 3.16. Diagramme de classes

Description du diagramme de classes

La classe	Détails
Ensemble_de_Données	Représente l'ensemble de données importé et contient les attributs basiques d'un ensemble de données.
Dimension	Cette classe représente l'ensemble de dimensions constituants l'ensemble de données initial. Cette classe dispose des opérations GetMin() et GetMax() qui seront utiles pour afficher les détails de chaque dimension.
Données Sélectionnées	Cette classe est un modèle de l'ensemble initial (une copie), c'est sur les « Données Sélectionnées » que le travail de prétraitement (optionnel) pourrait se faire.
Prétraitements	Cette classe est une classe abstraite. Elle est spécifiée par trois classes distinctes et qui concernent le prétraitement de données c'est les classes : transformations, normalisation et filtrage.
Distance	Classe abstraite spécifiée en la classe Minkowski qui elle-même contient une généralisation vers deux classes distinctes : la classe Euclidienne et la classe Manhattan. Ces classes contiennent des opérations de calcule de distance mais disposent toutes d'un attribut constant « p » qui diffère pour chaque distance créant ainsi la différence entre chaque type de distance.
Paramertres	Cette classe est composée d'attributs qui permettent de réaliser le clustering par la suite. Elle est aussi composée d'une classe Distance (elle-même deviendra attribut de la classe Parametres par la suite)
Clustering K_means	Cette classe récupère tous les paramètres du clustering K_means puisqu'elle en est composée, ces paramètres sont : « K », le paramétrage de distance choisie et l'initialisation des centres des clusters, etc. et exécute les itérations du K_means grâce à l'opération Executer_clustering ().
Resultats	Cette classe vient représenter les résultats du clustering par l'attribut Table_clusters.
Parallel coodinates	Cette classe dessine le graphique du parallel coordinates en fonction du nombre de dimensions par des axes verticaux, vient alors l'opération generer_PC () qui projettera les valeurs des enregistrements dans chaque dimensions dessinant ainsi plusieurs polylignes.

Tableau 3.7. Description des classes du diagramme de classes

8. Conclusion

Au cour de ce chapitre, nous avons introduit le clustering visuel de données multidimensionnelles comme étant la solution à notre problématique. Nous avons établi un couplage entre la visualisation de données multidimensionnelles (notre problématique) et la visualisation des résultats du clustering, la combinaison a été faite en se basant sur le processus proposé par Daniel Keim. Par la suite, nous avons présenté plus en détails la technique de visualisation choisie (parallel coordinates) et nous sommes passés à la conception de notre outil, le cycle de vie ainsi que le langage de modélisation choisi ont été détaillés par différents diagrammes à chaque niveau du cycle de vie répondant ainsi aux exigences du processus UP. Le cycle de vie en cascade sera finalisé dans le chapitre suivant, l'implémentation, tests et résultats de notre application.

Chapitre IV

Implémentation, tests et résultats

1. Introduction

Dans ce chapitre nous allons définir les outils de développement choisis pour l'implémentation de notre système. Ensuite, nous présentons notre application, les tests et les résultats.

2. Outil de développement et langage de programmation

Pour la réalisation de notre projet, nous avons utilisé le langage de programmation C++. Le C++ est un langage de programmation crée en 1983 par Bjarne Stroustrup. C++ est décrit comme suit [31] :

- Une amélioration du langage de programmation C
- Un langage de programmation qui supporte l'orienté objet mais ne l'impose pas (C++ au départ avait comme nom « C With Classes »)
- Un langage qui prend en charge la programmation générique
- C++ soutient l'abstraction des données

En plus de ces importantes caractéristiques, nous avons décidé de développer en C++ pour les raisons suivantes :

- Rapidité d'exécution : comparé à d'autres langages de programmation actuels, C++ permet une exécution rapide et ceci est lié à sa nature de langage compilé en code machine natif.
- Le large choix d'environnements de développement et d'API: notre choix s'est porté sur l'API/Framework « Qt » qui permet notamment la portabilité.

L'API Qt offre de nombreux composants graphiques (widgets) utiles à notre application, les interfaces graphiques « basiques » réalisées sur Qt sont élégantes, et compte tenu de nos besoins graphiques, notre choix s'est orienté vers ce Framework.

3. Présentation d'EyeViz

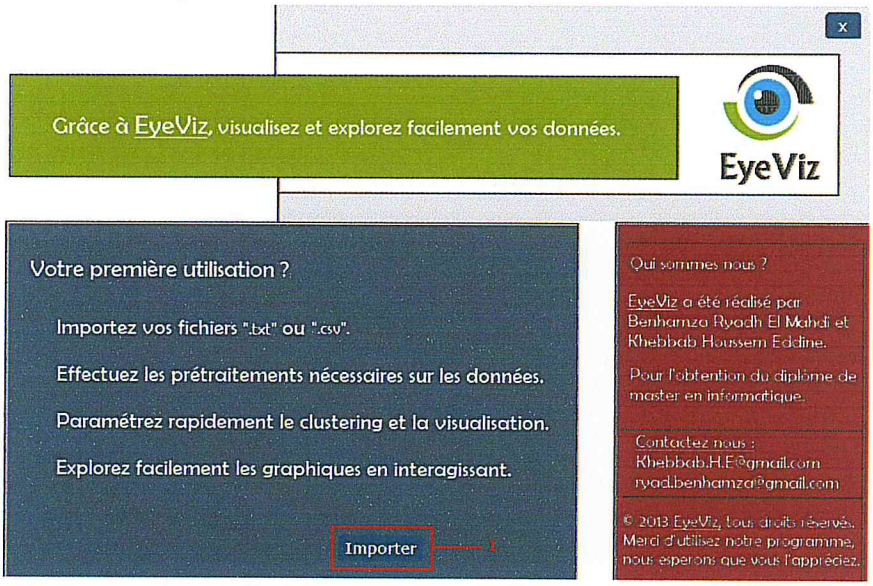


Figure 4.1. Interface d'accueil d'EyeViz.

Lors de l'exécution de notre application, une interface simple est affichée, contenant principalement les étapes qu'offre EyeViz résumées au nombre de quarte. Le bouton « Importer »(1) indique à l'utilisateur qu'il doit importer le fichier contenant l'ensemble de données à visualiser. L'interface principale d'EyeViz est voulue communicative.

3.1. Interface d'importation de fichier

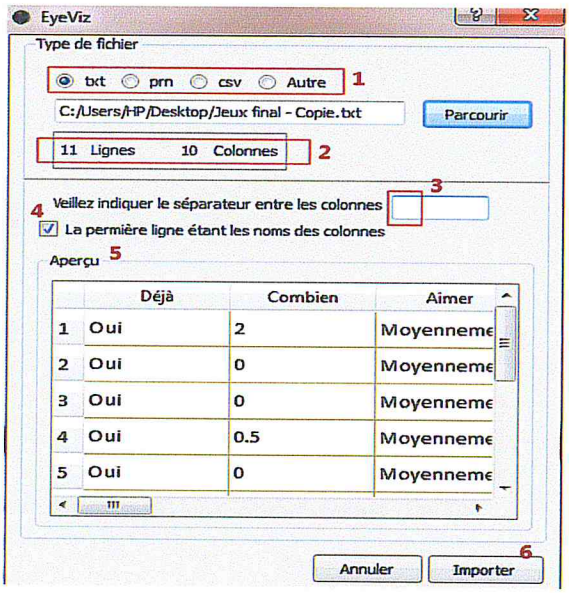


Figure 4.2. Interface de l'importation de fichier

A ce niveau, l'utilisateur doit indiquer le type de fichier à importer (1), une fois choisi et importé, des informations sur l'ensemble de données de ce fichier sont

automatiquement calculées puis affichées dans (2). L'étape suivante consiste à demander à l'utilisateur d'introduire le séparateur entre les colonnes de l'ensemble de données importé (3), pour notre exemple, le séparateur est un espace.

Le champ (4) est coché lorsque la première ligne de la table de données contient les noms de colonnes.

Une fois les étapes précédentes franchies, le champ (5) propose un aperçu de l'ensemble de données afin que l'utilisateur confirme son choix. Le bouton (6) propose à l'utilisateur d'importer le fichier sur l'interface des prétraitements.

Afin d'illustrer les fonctionnalités qu'offrent EyeViz, il est impératif d'établir des batteries de testes 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 du graphique du parallel coordinates final, nous montrerons aussi des différentes vues engendrées par des paramétrages distincts sur le même ensemble de données.

4. Tests et résultats

4.1. Présentation de l'ensemble de données

Pour nos testes effectués, nous avons utilisé un ensemble de données présenté lors d'une enquête réalisée par les chercheurs D. Nolan et T. Speed en 1994 sur une population qui ciblait 91 étudiants universitaires ayant accepté de participer à cette recherche [2].

Cette étude consistait à consulter un ensemble d'étudiants sur leurs opinions concernant les jeux vidéo ainsi que leurs préférences dans ce domaine. Pour tester EyeViz, nous utilisons un fichier contenant 11 individus parmi les 91 initiaux (Tableau 4.1) et les dimensions y sont au nombre de 9. [2]

	Déjà	Combien	Aimer	Lieu	Frequence	Occupé	Educatif	Sexe	Age	Note_Souhaitée
1	Oui	2	Moyennement	Ordi_dom	Hebdo	Non	Oui	Femme	19	A
2	Oui	0	Moyennement	Ordi_dom	Mensu	Non	Non	Femme	18	C
3	Oui	0	Moyennement	Arcade	Mensu	Non	Non	Homme	19	B
4	Oui	0.5	Moyennement	Ordi_dom	Mensu	Non	Oui	Femme	19	B
5	Oui	0	Moyennement	Ordi_dom	Semestr	Non	Oui	Femme	19	B
6	Oui	0	Moyennement	Console	Semestr	Non	Non	Homme	19	B
7	Oui	0	Non	Ordi_dom	Semestr	Non	Non	Homme	20	B
8	Oui	0	Moyennement	Ordi_dom	Semestr	Non	Non	Femme	19	B
9	Oui	2	Moyennement	Console	Quoti	Oui	Oui	Homme	19	A
10	Oui	0	Moyennement	Ordi_dom	Semestr	Non	Oui	Homme	19	A
11	Non	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	Homme	19	A

Tableau 4.1. Ensemble de données utilisé [2].

Le tableau suivant décrit l'ensemble de dimensions constituant l'ensemble de données utilisé dans nos testes (Tableau 4.2).

Dimension	Description
Déjà	Avez-vous déjà joué à des jeux vidéo.
Combien	Le temps passé à jouer à des jeux vidéo dans la semaine précédant l'enquête, en heure.
Aimer	Aimez-vous les jeux vidéo.
Lieu	Lieu ou vous jouez aux jeux vidéo.
Fréquence	A quelle fréquence jouez-vous à des jeux vidéo.
Occupé	Jouez-vous lorsque vous êtes occupés.
Educatif	Pensez-vous que les jeux aux quels vous jouez
Age	Quel âge avez-vous.
Note_Souhaitée	Quelle note souhaitez-vous avoir pour le prochain teste universitaire.

Tableau 4.2. Description des dimensions de l'ensemble de données utilisé.

Chaque dimension présente des valeurs différentes qui sont décrites comme suite :

Déjà : les valeurs de cette dimension sont au nombre de deux et qui sont oui ou non.

Combien : contient des valeurs réelles.

Aimer : peut contenir des valeurs nominales discrètes : oui, non ou moyennement.

Lieu : contient des valeurs nominales continues.

- Ordi_dom (ordinateur domestique) : indique que l'individu joue sur son ordinateur chez lui.
- Arcade : indique que l'individu joue dans des salles d'arcades.
- Console : indique que l'individu joue à des consoles de jeux.

Fréquence : contient les valeurs nominales discrètes suivantes :

- Semestr : indique que l'individu joue chaque semestre.
- Mensu : indique que l'individu joue chaque mois.
- Hebdo : indique que l'individu joue à une fréquence hebdomadaire.
- Quoti : indique que l'individu joue quotidiennement.

Occupé : contient des valeurs nominales au nombre de deux : oui ou non.

Educatif : contient des valeurs nominales au nombre deux: oui ou non.

Age : est une dimension contenant des valeurs numériques continues.

Note_Souhaitée : est une dimension qui contient des valeurs nominales ordinale représentant un système de notation universitaire (A, B, C, D, etc.).

Notons que l'individu 11 de l'ensemble de données (figure 4.1) présente la valeur (n.a) dans les dimensions : Combien, Aimer, Lieu, Fréquence, Occupé, Educatif. La valeur n.a est considérée comme étant une valeur manquante, le but étant de montrer qu'EyeViz traite le problème des valeurs manquantes.

Pour réaliser nos testes et montrer la capacité d'analyse et d'interprétation qu'offre EyeViz, nous choisissons de visualiser ce même ensemble de données avec des paramétrages de clustering ainsi que de visualisations différents afin de montrer la différences de graphes et ainsi la possibilité de réaliser des interprétations différentes.

4.1.1 L'étape de prétraitements

- **Transformation de données**

Figure 4.3. Interface de prétraitements

Cette interface offre les prétraitements optionnels tels que la transformation (bouton 2), la normalisation si toutes les valeurs sont transformées (bouton 3) et le filtrage (bouton 4). Le champ (5) montre l'emplacement du fichier utilisé. Cette étape est un passage obligé à chaque importation de fichier.

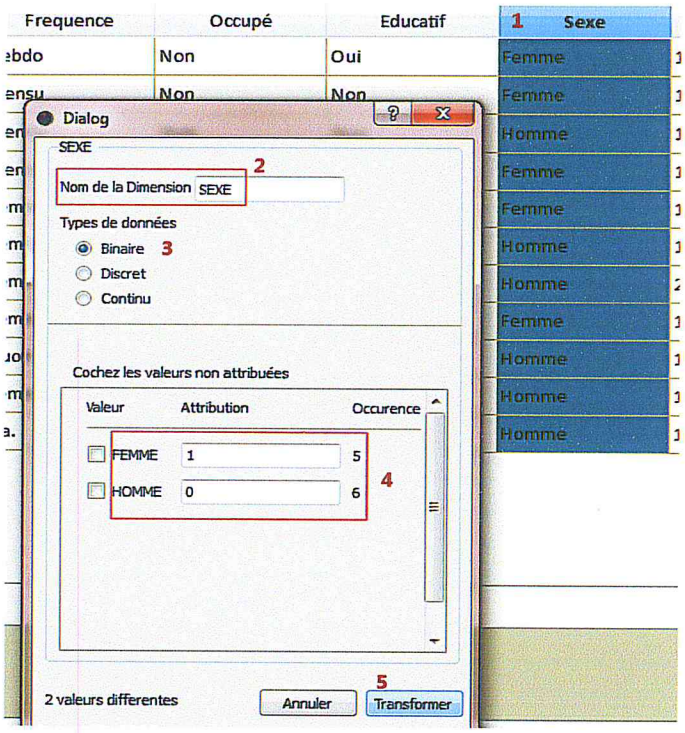


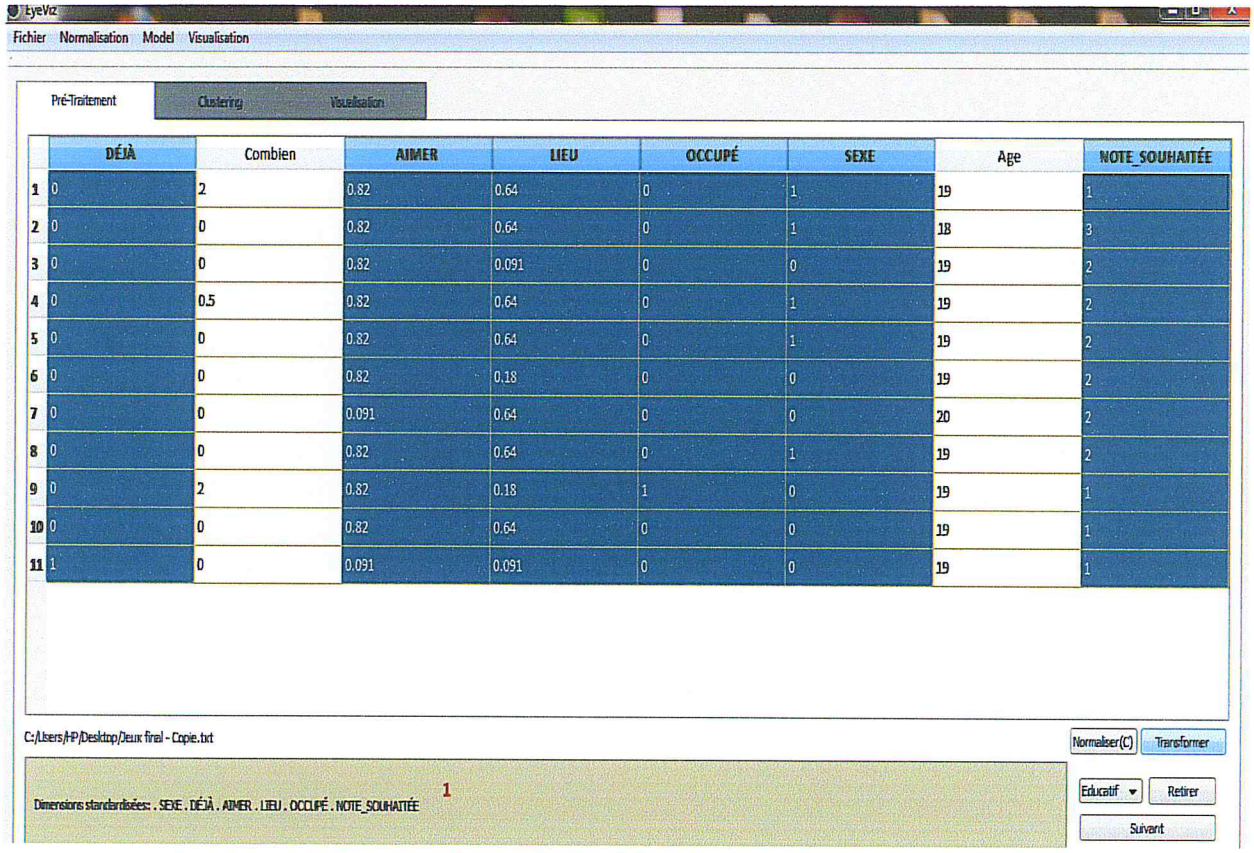
Figure 4.4. Un exemple d'une transformation de valeurs d'une dimension

Après la sélection de la dimension à transformer, l'utilisateur clique sur le bouton (2) de la figure 4.3, l'interface de transformation apparait. Dans la figure 4.4, la dimension sexe contient deux valeurs récurrentes (Homme et Femme) (1), notre système propose trois types de données cibles (3). L'utilisateur peut aussi changer le nom de la dimension (2). Dans le champ (4), notre système affiche le nombre d'occurrences des valeurs de la dimension choisie. Dans la figure 4.4 il s'agit de données de type binaire, EyeViz attribut automatiquement la valeur 0 à la valeur qui possède le plus grand nombre d'occurrences. Notons que l'utilisateur peut modifier les valeurs attribuées à sa guise. En cliquant sur le bouton (5), la dimension sélectionnée prend les nouvelles valeurs attribuées.

A ce niveau, l'utilisateur peut aussi attribuer des valeurs aux valeurs manquantes ou comme dans notre cas aux valeurs non attribuées (n.a).

La figure 4.5 montre l'ensemble de données choisi après les transformations jugées nécessaires à effectuer pour les étapes suivantes :

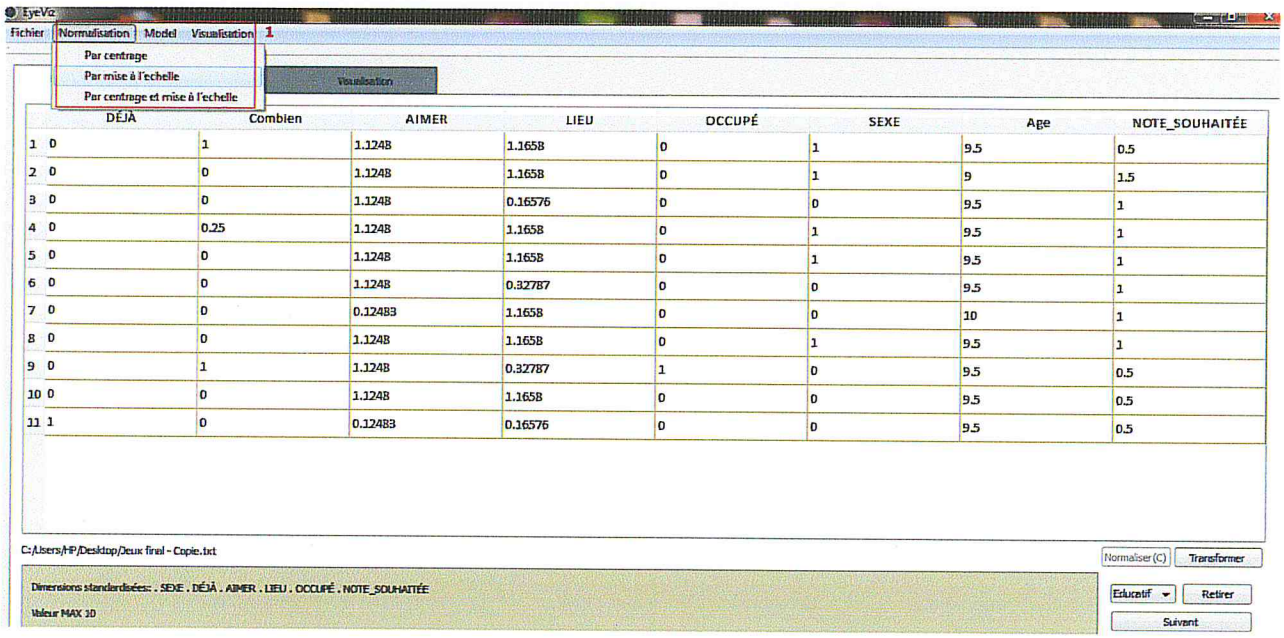
Le champ (1) montre les dimensions sur lesquelles l'utilisateur a effectué une transformation.



	DÉJÀ	Combien	AIMER	LIEU	OCCUPÉ	SEXE	Age	NOTE_SOUHAITÉE
1	0	2	0.82	0.64	0	1	19	1
2	0	0	0.82	0.64	0	1	18	3
3	0	0	0.82	0.091	0	0	19	2
4	0	0.5	0.82	0.64	0	1	19	2
5	0	0	0.82	0.64	0	1	19	2
6	0	0	0.82	0.18	0	0	19	2
7	0	0	0.091	0.64	0	0	20	2
8	0	0	0.82	0.64	0	1	19	2
9	0	2	0.82	0.18	1	0	19	1
10	0	0	0.82	0.64	0	0	19	1
11	1	0	0.091	0.091	0	0	19	1

Figure 4.5. L'ensemble de données après transformations des dimensions.

- **Normalisation**



	DÉJÀ	Combien	AIMER	LIEU	OCCUPÉ	SEXE	Age	NOTE_SOUHAITÉE
1	0	1	1.1248	1.1658	0	1	9.5	0.5
2	0	0	1.1248	1.1658	0	1	9	1.5
3	0	0	1.1248	0.16576	0	0	9.5	1
4	0	0.25	1.1248	1.1658	0	1	9.5	1
5	0	0	1.1248	1.1658	0	1	9.5	1
6	0	0	1.1248	0.32787	0	0	9.5	1
7	0	0	0.12483	1.1658	0	0	10	1
8	0	0	1.1248	1.1658	0	1	9.5	1
9	0	1	1.1248	0.32787	1	0	9.5	0.5
10	0	0	1.1248	1.1658	0	0	9.5	0.5
11	1	0	0.12483	0.16576	0	0	9.5	0.5

Figure 4.6. L'ensemble de données après la normalisation par mise à l'échelle.

Dans la figure 4.6. Le champ (1) représente les trois types de normalisations qu'EyeViz propose.

Les données brutes (après transformations) sont rarement utilisées dans l'analyse. La préparation des données pour le clustering requiert leur normalisation. [3]

Considérons que X^* soit la matrice de données initiale.

$$X^* = \begin{pmatrix} x_{11}^* & x_{12}^* & \dots & x_{1d}^* \\ x_{21}^* & x_{22}^* & & x_{2d}^* \\ x_{n1}^* & x_{n2}^* & & x_{nd}^* \end{pmatrix}$$

Pour chaque attribut de la matrice X^* , la moyenne m_i et la variance S_j^2 s'expriment par : [3]

$$m_j = \left(\frac{1}{n}\right)\sum_{i=1}^{n} x_{ij}^*$$

$$S_j^2 = \left(\frac{1}{n}\right)\sum_{i=1}^{n} (x_{ij}^* - m_j)^2$$

- Normalisation par centrage : consiste à extraire de chaque élément sa moyenne. Sa formule est la suivante [3] :

$$x_{ij} = x_{ij}^* - m_j$$

- Normalisation par mise à l'échelle : sa formule est la suivante [3] :

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

Dans la formule précédente, X_{*j} et X_{*j}^* représentent les vecteurs d'attributs de l'ensemble X respectivement X^*. $X_{*j\,max}^*$ et $X_{*j\,min}^*$ représentent les valeurs maximales et minimales observées de l'attribut X_{*j}^*.

Cette normalisation est recommandée lorsque l'ensemble de données contient des attributs binaires car ce type de normalisation garde les valeurs binaires.

- Normalisation par centrage et mise à l'échelle : sa loi est donnée par [3] :

$$x_{ij} = \frac{x_{ij}^* - m_j}{S_j}$$

4.1.2 L'étape de regroupement (Clustering)

Nous effectuons nos tests sur le même ensemble de données avec un paramétrage de clustering différent, nous verrons des résultats différents et nous les interpréterons par la suite.

Le Tableau 4.3 décrit les deux manières de paramétrages effectués sur l'ensemble de données.

	Test 1	**Test 2**
Nombre de clusters (K)	3	3
Type de distance	Manhattan	Euclidienne
Nombre d'itérations de l'algorithme K-means	2	2
Les centres initiaux	{1, 7, 11}	{4, 5, 9}

Tableau 4.3. Les paramètres de clustering des deux tests.

Le choix des nombres de clusters est le même pour pouvoir distinguer la différence dans la table des clusters ainsi que sur le graphique même avec le même nombre de K, les centres initiaux différent ce qui fait que les résultats différent à leur tour.

Nous présenterons l'interface de paramétrage de clustering effectué pour le test 1 suivie de celle du test 2 pour montrer les différences entres les tables de distances et de clusters des deux tests.

Les deux graphiques du parallel coordinates générés par les deux tests seront par la suite montrés pour illustrer la différence réelle ainsi que les différentes interprétations.

- **Résultats de clustering**

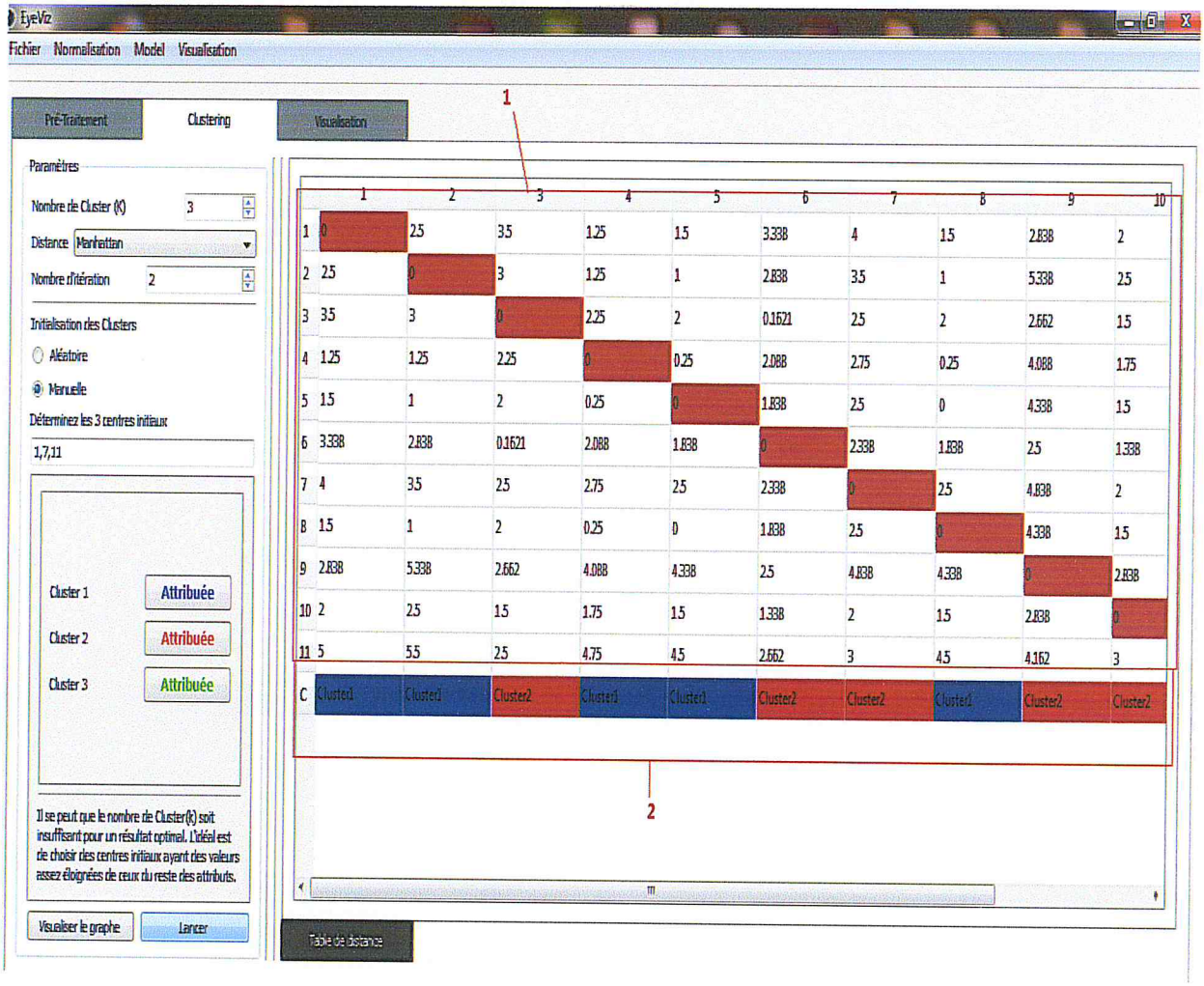


Figure 4.7. Résultats du clustering du premier test.

Dans cette fenêtre, l'utilisateur peut visionner la table de distance diagonale (champ (1)), et consulter en même temps la liste des clusters (champ (2)). Une fois cette étape confirmée, l'utilisateur clique sur le bouton « visualiser le graphe » pour consulter le parallel coordinates correspondant à ce paramétrage dans l'onglet « visualisation ».

Figure 4.8. Résultats du clustering du second test.

Dans le test 2, nous avons gardé le même ordre d'attribution des couleurs pour chaque cluster afin d'analyser rapidement les différences entres les deux tables de clusters, nous pouvons facilement comparer les deux listes de clusters et s'apercevoir des différences entre les deux.

Nous allons passer aux interfaces de visualisation des graphes des deux tests afin de les analyser.

4.1.3. Visualisation des résultats de clustering

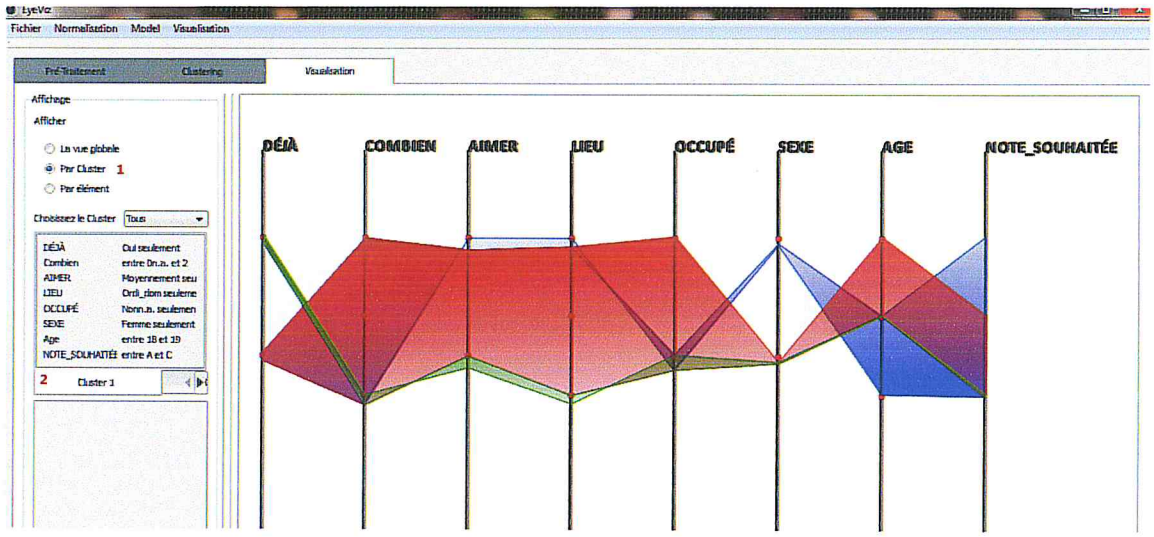


Figure 4.9. Visualisation des résultats du clustering du test 1 (par clusters).

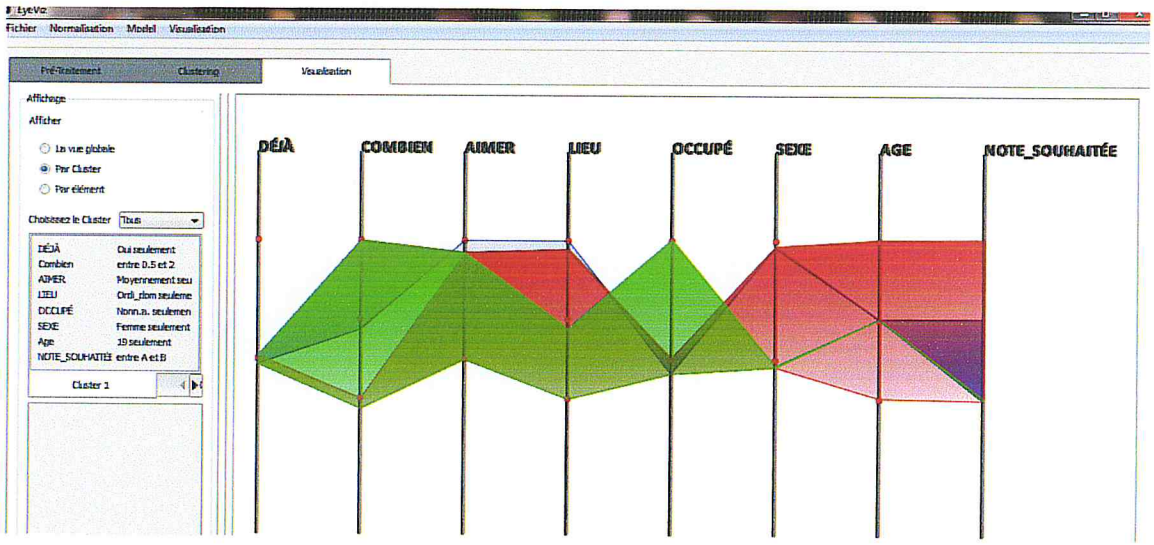


Figure 4.10. Visualisation des résultats du clustering du test 2 (par clusters).

L'interface de la visualisation offre la possibilité à l'utilisateur d'interagir sur le graphique en choisissant d'afficher les résultats par cluster (figure 4.9. (1)). Ou par individu, l'utilisateur pourra introduire l'élément souhaité et ne consulter que ce dernier, cette interface offre aussi une vue combinant ces deux vues.

Dans le champ (2) (figure 4.9) l'utilisateur consulte les détails des valeurs formant chaque cluster.

La figure 4.11 et la figure 4.12 illustrent la visualisation des deux tests par éléments pour montrer les différences entres les deux paramétrages.

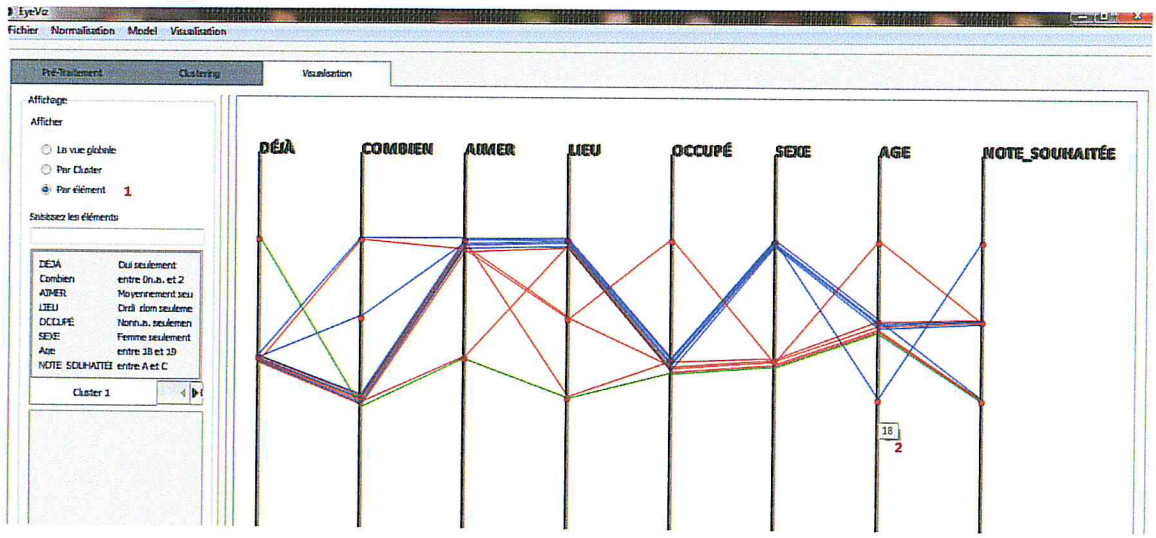


Figure 4.11. Visualisation des résultats du clustering du test 1 (par éléments).

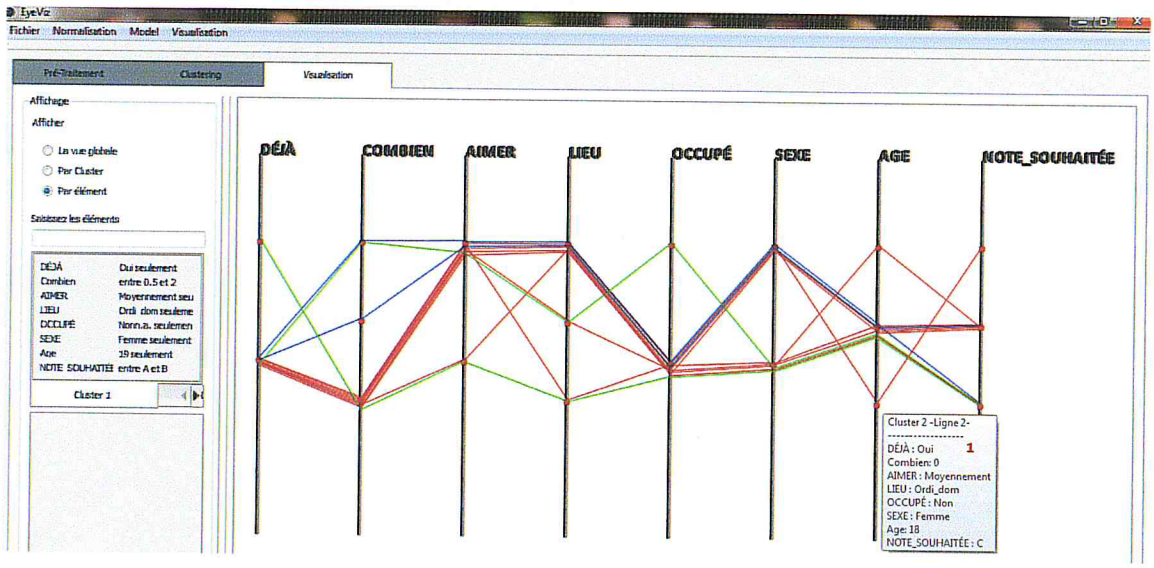


Figure 4.12. Visualisation des résultats du clustering du test 2 (par éléments).

Le champ sélectionné dans la figure 4.11 (1) est l'affichage des résultats de clustering par élément. Quant au champ (2) de cette même figure, il représente une valeur de la dimension « Age », les valeurs apparaissent dans les points leur correspondants lorsque l'utilisateur positionne le curseur dessus. Dans la figure 4.12, le champ (1) représente les informations d'un individu (l'individu 2), ces informations apparaissent sur toutes les lignes polygonales représentant chacune un élément lorsque l'utilisateur positionne le curseur sur l'élément souhaité.

Si on veut interpréter le test 1 par exemple, on pourrait dire que la figure 4.9 illustre les trois clusters et leurs couleurs respectives, l'analyse par cluster fournit surtout la plage de valeurs (la valeur maximale et la valeur minimale) des individus qui composent chaque cluster. L'étude dans ce cas est axée sur la dimension « sexe » dans le but de comprendre le comportement des deux sexes face aux jeux vidéo. Le cluster 1 représenté par la couleur bleu a pour centre l'individu 1 qui est une femme, quant au cluster 2 et le cluster 3 qui ont pour couleurs : rouge et vert, ont pour centres les individus 7 et 11, tous deux des hommes. A ce niveau, une comparaison avec le test 2 montre déjà une différence graphique (Figure 4.10). Les lignes polygonales présentées dans la figure 4.11 représentent les éléments de l'ensemble de données, les lignes bleues illustrent les « femmes », ces dernières jouent sur des ordinateurs domestique (dimension « Lieu ») et sont âgées presque toutes de 18 ans (dimension « Age »), toutes aiment moyennement les jeux vidéo (dimension « Aimer »). Trois d'entres elles n'ont pas joué la semaine précédant cette enquête (dimension « Combien ») et toutes ont déjà joué à des jeux vidéo (dimension « Déjà »). Le graphique montre aussi qu'aucune femme ne joue lorsqu'elle est occupée. Dans la figure 4.9 nous remarquons que le cluster 3 (vert) est en réalité composé que de son centre (l'individu 11) ce dernier se retrouve seul dans son cluster étant donnée qu'il est très différent des autres individus dans presque toutes les dimensions, on peut interpréter ceci en disant que l'individu 11 est une personne qui n'aime probablement pas les jeux vidéo.

5. Conclusion

Dans ce chapitre, nous avons présenté l'environnement ainsi que le langage de programmation utilisé pour la réalisation d'EyeViz. Ensuite, nous avons présenté l'interface d'accueil d'EyeViz et nous avons décrit la manière d'importer un fichier contenant l'ensemble de données, et de prétraiter les données avant de passer aux paramétrages de clustering et de la visualisation. Dans le but de démontrer une exécution complète, nous avons défini un ensemble de données et nous avons présenté toutes ses dimensions ainsi que les types de données le constituants. L'étape des tests et des résultats a conclut ce chapitre en illustrant la différence entre deux paramétrages de clustering et les graphiques du Parallel Coordinates correspondants à chaque test. Ces tests ont démontré que l'analyse est fiable et l'exploration est beaucoup plus simple et efficace grâce à EyeViz.

Conclusion générale

Conclusion générale

Le thème traité dans ce mémoire est la visualisation des résultats du clustering appliqué sur des ensembles de données multidimensionnelles et de types hétérogènes. De nombreuses approches et de travaux ont été réalisés dans le domaine de la visualisation de données hétérogènes, d'autres solutions existent dans le domaine du clustering et sa visualisation, mais notre approche concerne un couplage entre clustering de données multidimensionnelles et la visualisation des résultats de ce dernier, et ce dans le but de faciliter la lecture et l'interprétation des données sources souvent impossibles à analyser à cause de la tailles importantes des enregistrements et celles des dimensions les constituant.

Nous avons d'abord commencé par une étude détaillée de chacune des deux parties constituantes le clustering visuel, à savoir, le clustering de données vu au cour de notre cursus, et la partie traitant les techniques de visualisations de données. Dans chaque partie nous avons choisi une méthode en ayant une certaine correspondance logique, le clustering en utilisant l'algorithme K-means produit des résultats qui seront projetés sur le « Parallel Coordinates » qui, de par sa définition, propose de représenter les dimensions de données sources sous formes de lignes verticales parallèles équidistantes. Nous nous sommes dirigés dans notre étude vers les prétraitements possibles sur les données sources afin de les standardiser pour différents calculs nécessaires.

Après avoir déterminé la solution adoptée pour palier à la difficulté d'analyse des masses de données importantes, nous sommes passés à l'analyse des besoins et à sa conception. Dans cette partie du travail, nous avons enrichi notre solution par des interactions proposées à l'utilisateur afin d'affiner son analyse, notre but était de mettre l'utilisateur au centre du système pour qu'il interagisse avec lui de manière intuitive, mettant ainsi en harmonie les affordances des objets. Enfin, nous avons implémenté une application qui présente de manière explicite les grandes étapes du clustering visuel, chaque étape produit des résultats, ces derniers repris en entrées dans l'étape suivante jusqu'à génération d'un graphe interactif. Nos tests ont été réalisés sur différentes sources de données supportées par notre application.

Cette étude nous a permis d'une part de découvrir le domaine de la visualisation de données comme étant un secteur en plein expansion et de plus en plus intégré en raison de l'immensité des données stockées dans le monde, et de l'appliquer d'autre part sur des notions déjà connues et apprises au cours de nos années universitaire tel que le clustering. Cette étude a donc permis d'arriver à résoudre notre problématique.

Perspectives

L'objectif initial de notre étude étant atteint, et compte tenu de la multitude des méthodes de clustering d'une part, et des variantes de techniques de visualisation d'une autre, une de nos perspectives d'avenir est d'enrichir EyeViz d'un choix d'algorithmes de clustering et d'offrir à l'utilisateur une palette variée de techniques de visualisations. Nous projetons aussi d'ajouter et de mettre en œuvre des fonctions d'interactions sur les graphes proposés à l'utilisateur dans sa démarche d'exploration visuelle.

Bibliographie

Bibliographie

[1] S. Pinker. (1997). "How the mind works". USA: W.W. Norton & Company. 660 p.

[2] B. S. Everitt, S. Landau, M. Leese, D, Stahl. (2011). "Cluster Analysis": 5th edition. UK: A John Wiley and Sons, Ltd. 330 p.

[3] C. Lazar. (2008), Méthodes non supervisées pour l'analyse des données multivariées. Mémoire Doctorat Recherche : Génie Informatique, Automatique et Traitement du Signal. Reims : Université de Reims Champagne Ardenne, 136 p.

[4] S. Vidal. (2006), Visualisation de l'information : un panorama d'outils et de méthodes. Dossier de Synthèse. France : Centre National de la Recherche Scientifique, 38 p.

[5] R. Mazza. (2004), Introduction to Information Visualization. Publication. Suisse : Université de Lugano, 24 p.

[6] B. Zhu ET H. Chen. (2004). Information Visualization. In : Blaise Cronin. « Annual Review of Information Science and Technology ». USA, 139-177.

[7] C. HURTER. (2010), Caractérisation de visualisations et exploration interactive de grandes quantités de données multidimensionnelles. Mémoire Doctorat : Informatique. Toulouse : Université de Toulouse, 190 p.

[8] L. Nowell S. Havre, B. Hetzler and P. Whitney. (2002), "Themeriver: Visualizing thematic hanges in large document collections". IEEE transaction on Visualization and Computer Graphics (volume: 8, Issue: 1), 9-20.

[9] N. Lopez M. Kreuseler and H. Schumann. (2000), "A scalable framework for information visualization". IEEE Symposium On Information Visualzation, The IEEE Computer Society Technical Committee on Visualization and Graphics, 9-10 October 2000, Salt Lake City, Utah, 27-36.

[10] D. Tang C. Stolte and P. Hanrahan. (2002), "Polaris: A system for query, analysis and visualization of multi-dimensional relational databases", IEEE Transactions on Visualization and computer Graphics, 52-65.

[11] G. Jaeschke, P. Gupta, and M. Hemmje. (2005), Modeling Interactive, Three-Dimensional Information Visualizations. In: E.J. Neuhold Festschrift. From Integrated Publication and Information Systems to Virtual Information and Knowledge Environments. 197-206.

[12] D. A. Keim. (2002), "Information Visualization and Visual Data Mining", IEEE transactions on Visualization and Computer Graphics, 1-8.

[13] D. F. Andrews. (1972), "plots of high-dimensional data", Biometrics, Vol 29, 125-136.

[14] J. M. Chambers, W. S. Cleveland, B. Kleiner and P. A. Tukey (1976), "Graphical Methods for Data Analysis", New York : Chapman and Hall.

[15] J. J. Van Wijk and R. Van Liere. (1993), "HyperSlice", IEEE Visualization

Conference, San Jose, CA. 119-125.

[16] S. Feiner and C. Beshers. (1990), "Worlds Within Worlds : Methaphors For Exploring N-Dimensional Virtual Worlds", UIST '90 Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology, 76-83.

[17] B. Alpern. (1991), "Hyperbox", IEEE Visualization '91, San Diego, CA, 133-139, 418.

[18] A. Inselberg. (1985), "The Plane With Parallel Coordinates". In: Springer-Verlag. The Visual Computer, Vol 1, 69-91.

[19] A. Inselberg and B. Dimsdale. (1987), "Parallel Coordinates for Visualizing Multidimensional Geometry", In Computer Graphics International '87. Tokyo.

[20] G. Grinstein, M. Trutschl, U. Cvek. (2002), "High-Dimensional Visualizations". Institute for Visualization and Perception Research: University of Massachusetts Lowell and Anvil Information, Inc. 14 p.

[21] P. Hoffman and G. Grinstein. (1999), "Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations" NPIV '99 (Workshop on New Paradigm in Information Visualization and Manipulation).

[22] D. Keim, J. Kohlhammer, G. Ellid et F. Masmann. (2010), "Mastering the information age solving problems with visual analytics". Germany: Eurographics Association. 167 p.

[23] Ke-Bing Zhang. (2007), Visual Cluster Analysis in Data Mining. Mémoire Doctorat : Philosophie. Australie : Departement Of Comuting Division of Informatique and Communication Sciences Macquarie University, 61 p.

[24] d'Ocagne Maurice. (1885), "Coordonnées Parallèles et Axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles". Paris : Gauthier-Villards.

[25] E. J. Wegman et Q. Luo. (1996), "High Dimensional clustering using parallel coordinates and the grand tour", Center for Computational Statistics. George Mason University, Fairfax ,VA.

[26] A. Inselberg. (1997), "Multidimensional detective", IEEE Symposium on Information Visualization. Phoenix, AZ. 100-105.

[27] L'outil orange. Orange [en ligne]. (Page consultée le 10/05/2013). http://orange.biolab.si/.

[28] Site officiel de l'outil XmdvTool, XmdvTool [en ligne]. (Page consultée le 10/05/2013). http://davis.wpi.edu/xmdv/.

[29] Site officiel de l'outil GGobi, GGobi out of sight, out of mind [en ligne]. (Page consultée le 10/05/2013). www.ggobi.org.

[30] B, Stroustrup. (1985), " The C++ Programming Language". USA: Addison-Wesley Publishing Company. 911 p.

[31] P. Roques. (2008). " UML2 Modéliser une application web". France : Editions Eyrolles. 246p.

[32] J. Steffe. (2005). Cour UML : école nationale des ingénieurs des travaux agricoles de bordeaux. Département entreprise et système. Unité de formation informatique et génie des équipements.

[33] A. K. Jain, M. N. Murty et P. I. Flynn. (1999). "Data Clustering: A Review". ACM Computing Surveys, Vol. 31, Issue 3. 264-323.

[34] A. K. Jain et R. C. Dubes. (1988). "Algorithms for Clustering Data". USA : Prentice Hall advanced reference series, 320 p.

[35] V. Kumar. (2000). "An Introduction to Cluster Analysis for Data Mining". Raport technique. C. S. Dept. Université du Minnesota.

[36] W. Pedrycz. (2005). " Knowledge-Based Clustering: From Data to Information Granules". USA, Hoboken : Wiley & Sons. 336p.

[37] M. Halkidi, Y.Batistakis, M. Vazirgiannis. (2001). "On Clustering Validation Techniques". Information Systems Journal, Kluwer Pulishers, vol.17, n°2-3. 107-145.

[38] S.B. Kotsiantis, P. E. Pintelas. (2004). "Recent Advances in Clustering: A Brief Survey". WSEAS Transactions on Information Science and Applications, Vol 1, n°1. 73-81.

[39] G.Lance et W.Williams. (1967). "A general theory of classification sorting strategies". Computer Journal, n°9. 373-386.

[40] D. L. Davies et D. W. Bouldin. (1979). "A Cluster Separation Measure". IEEE Transactions On Pattern Analysis and Machine Intelligence. Vol 1, n°2. 224-227.

[41] J. C. Dunn. (1974). « Well separated clusters and optimal fuzzy partitions ». Cybernetics and Systems, n°4. 95-104.

[42] L. Hubert et J. Schula. (1976). "Quadriatic assignment as a general data analysis strategy". British Journal of Mathematical Psychology, n°29. 190-241.

[43] C. Frelicot. (1992). Un système adaptatif de diagnostic prédictif par reconnaissance de formes floue. Thèse de doctorat : reconnaissance des formes. Université Technologique de Compiègne. 162p.

[44] C. V. Rijsbergen. (1979). "Information Retrieval". Londre : Butterworths. 208p.

[45] J. Handl. (2003). And-based methods for tasks of clustering and topographic mapping. Thèse de Master : Informatique. Allemagne. Université d'Erlangen-Nürnberg.

[46] J. A. Hartigan et M. A. Worng. (1979). "A K-means clustering algorithm". Applied Statistics, Vol 28. 100-108.