UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES
DEPARTEMENT DES MATHEMATIQUES

Master Mathématique et Application au Calcul Scientifique (MACS)

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du Diplôme de Master Sciences et Techniques (MST)

Indice de validité en statistique décisionnelle

Réalisé par: **OUBBIH Omar**

Encadré par: **Pr. AMMOR Ouafae**

Soutenu le 16 juin 2016

Devant le jury composé de:

- Pr. AMMOR Ouafae — Faculté des Sciences et Techniques de Fès
- Pr. ELHILALI ALAOUI Ahmed — Faculté des Sciences et Techniques de Fès
- Pr. EZZAKI Fatima — Faculté des Sciences et Techniques de Fès
- Pr. HILALI Abdelmajid — Faculté des Sciences et Techniques de Fès

Année Universitaire 2015 / 2016

Université Sidi Mohammed Ben Abdellah
Faculté des Sciences et Techniques Fès
Département de Mathématiques

Master : Mathématiques et Applications au Calcul Scientifique (MACS)

Memoire du Projet de Fin d'Études

Indice de validité en statistique décisionnelle

Réalisé par :

- *OMAR OUBBIH*

Encadré par :

- *Mme. O. AMMOR*

Membres de Jury :

- *Mme.*

- *M.*

Année Universitaire : 2015/2016

Remerciement

Avant tout, je remercie Dieu le très haut qui m'a donné le courage et la volonté de réaliser ce modeste travail. Sans sa miséricorde, ce travail n'aura pas abouti.

Au terme de ce travail, je tiens à exprimer mon remerciement les plus chaleureux à *Mme. AMMOR Ouafae* **professeur à la Faculté des Sciences et Techniques de Fès pour son précieux, judicieux conseils, son encadrement, son disponibilité et son suivi durant toute la période du ce projet. J'aimerai lui exprimer ma sincère reconnaissance pour son effort et de m'avoir offert toutes les informations nécessaires à l'évolution et à la réalisation de ce projet.**

Je tiens à remercier également les membres du jury : *Mr. ELHILALI ALAOUI Ahmed*, *Mme. EZZAKI Fatima* **et** *Mr. HILALI Abdelmajid* **qui ont accepté d'évaluer notre travail.**

Mon vif remerciements s'adresse également à nos professeurs qui nous ont encouragés et nous ont aidés avec leurs remarques et observations durant la période de notre parcours universitaire.

Enfin, mon gratitude s'adresse à tous ceux et celles qui ont contribué, de près ou de loin, à la réalisation de ce modeste travail, qu'ils trouvent ici l'expression de nos remerciements les plus distingués.

Table des matières

INTRODUCTION GÉNÉRALE

La classification est une discipline relié de près et de loin à plusieurs domaines, elle a pour but d'identifier les classes aux quelles appartient les objets à partir d'un certains traits descriptifs. Son application a joué un rôle dans presque toutes les sciences et techniques qui font appel à la statistique multidimensionnelle. A titre d'exemple les sciences biologiques : botanique, zoologie, écologie qui utilisent le termes "taxionomie" pour désigner l'art de le classification. Ainsi que la reconnaissances des formes, imagerie, et segmentation ... etc. Cependant, il existe deux variantes de classification : supervisée et non supervisée :

Dans l'approche de la classification supervisée, les classes existent a priori, on dispose en donnée du problème d'un ensemble de classes et d'objets, chacun d'eux étant déjà placé dans une classe qui lui convient aux mieux. Le but de cette démarche est de pouvoir allouer une classe à un nouvel objet en restant le plus cohérent possible avec la structure initiale en classes.

Dans l'approche de la classification non supervisée les classes sont encore inexistantes. pour cette démarche, on dispose donc au départ d'un ensemble d'objet. L'idée consiste à découper l'ensemble des objets en groupes (clusters) de telle sorte que les caractéristiques d'objets dans un même cluster soient similaires et les caractéristiques des objets dans des clusters différents soient distinctes. Dans la suite de ce travail nous nous intéressons exclusivement à l'approche non supervisée, encore nommée clustering ou segmentation.

Les techniques de classifications floue non supervisée sont très utilisés. Cependant, la majorité des algorithmes de clustering souffrent du problème de détermination du nombre de clusters qui souvent laissé à l'utilisateur. A ce problème, plusieurs fonctions appelés

indices de validités ont été proposées.

Un indice de validité est une fonction qui mesure la qualité du résultat final d'un algorithme de clustering. Trois critères sont en général utilisés : Externe, interne et relatif, les deux premiers sont basés sur les méthodes statistiques et demandent beaucoup de temps de calcul. Les techniques basés sur le critère relatif fonctionne correctement dans le cas de classes compacts et sans chevauchement. cependant, plusieurs applications présentent différentes degrés de chevauchement, et l'application de ces algorithmes reste limitée.

Le but de ce projet de fin d'étude est d'une part montrer l'importance de cet indice de validité en statistique décisionnelle, et d'autre part faire une comparaison entre quelques indices de validités existent afin de donner les avantages et inconvénients de leurs utilisations et sélectionner un meilleur indice qui pourra nous donner le nombre optimal de cluster dans un jeu de données (on utilise pour cela les données d'iris). En appliquant cet indice sur un exemple de segmentation d'imagerie médicale de cette étude.

Chapitre 1

Outils et Technique de Bases

Ce chapitre présente essentiellement les concepts et techniques de base liées à notre étude. Dans un premier temps, on va décrire le principe fondamental de la classification non supervisée ou clustering. Dans la seconde partie, on distinguera aussi entre le clustering hiérarchique et le clustering par partition. On s'intéressera essentiellement aux méthodes de classification non supervisée k-means, PAM et la technique de clustering hiérarchique ascendante. On finira ce chapitre par une notion essentielle dans la théorie de l'information à savoir le principe d'entropie, qu'est souvent utilisé et fournit des résultats très intéressants pour la sélection du nombre optimal de clusters.

1.1 Principe fondamental du clustering

1.1.1 Définition

L'objectif d'une tâche de classification non supervisée : cluster [4], consiste à proposer une partition des objets en k sous ensemble, où le paramètre k est le nombre de regroupements attendus par l'utilisateur. une variation de cette tâche est de ne pas utiliser le nombre attendu de regroupements comme une donnée du problème. Dans ce cas, l'algorithme construit plusieurs partitions candidates et choisie la meilleure partition est celle qui optimise un critère de qualité des partitions.

Chaque cluster issu de ce processus doit vérifier les deux propriétés suivantes :

- La cohésion interne (les objets appartient à ce cluster soient les plus similaires possibles)
- L'isolation externe (les objets appartenant aux autres clusters soient les plus distincts possibles).

Le regroupement repose sur une mesure précise de la similarité/dis-similarité des objets que l'on veut regrouper. Cette mesure est appelée distance en métrique.

1.1.2 Quelques définitions

Étant donné un ensemble $O = \{o_1, ..., o_N\}$ de N objets correspondant chacun à un point d'un espace métrique à M dimensions dont les coordonnées sont notées par le vecteur $x_i = (x_{i1}, , x_{iN})$ pour l'objet o_i, on peut définir les notions suivantes :

Définition 1

Mesure de dissimilarité [4] : On appelle indice ou mesure de dissimilarité sur un ensemble O une application $d : O \times O \longrightarrow \mathbb{R}^+$ qui vérifie les propriétés suivantes pour tout couple $(x_1, x_2) \in O \times O$:

- $d(x_1, x_2) = d(x_2, x_1)$ (symétrie)
- $d(x_1, x_2) \geqslant 0$
- $d(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$ (séparabilité)

Autrement dit, moins les unités x_1 et x_2 se ressemblent, plus le score est élevé. Remarquons qu'une distance est une dissimilarité, puisque toute distance possède les deux

propriétés précédentes ainsi que l'inégalité triangulaire. Toutes les distances connues, en particulier la distance euclidienne, sont donc des exemples de dissimilarité.

A l'inverse, une autre possibilité consiste à mesurer la ressemblance entre les observations à l'aide d'une similarité :

Définition 2

On appelle métrique [4] sur un ensemble O , une application d : $d : O \times O \longrightarrow \mathbb{R}^+$ qui vérifie les propriétés suivantes pour tout couple $(x_1, x_2) \in O \times O$:

- $d(x_1, x_2) = d(x_2, x_1)$ (symétrie)
- $d(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$ (séparabilité)
- $d(x_1, x_2) \leqslant d(x_1, x_3) + d(x_2, x_3)$ (inégalité triangulaire)

Contrairement à la dissimilarité, plus les unités x_1 et x_2 se ressemblent plus le score est élevé. On peut citer comme exemple de similarité la valeur absolue du coefficient de corrélation :

$$|\rho(x_1, x_2)| = |\frac{\sum_{j=1}^{p}(x_{1j} - \overline{x}_1)(x_{2j} - \overline{x}_2)}{\sqrt{\sum_{j=1}^{p}(x_{1j} - \overline{x}_1)^2 \sum_{j=1}^{p}(x_{2j} - \overline{x}_2)^2}}|$$

Définition 3

Ultramétrique [4] : On appelle ultramétrique sur un ensemble O une application $d : O \times O \longrightarrow \mathbb{R}^+$ qui vérifie les propriétés suivantes pour tout couple $(x_1, x_2) \in O \times O$:

- $d(x_1, x_2) = d(x_2, x_1)$ (symétrie)
- $d(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$ (séparabilité)
- $d(x_1, x_2) \leqslant \max\{d(x_1, x_3), d(x_3, x_2)\}$ (inégalité ultramétrique)

où x_1, x_2 et x_3 sont des vecteurs.

L'indice de dissimilarité le plus communément utilisé est la métrique de Minkowski :

$$d_r(o_i, o_j) = d_r(x_i, x_j) = (\sum_{k=1}^{M} q_k |x_{ik} - x_{jk}|^r)^{1/r}$$

q_k est un facteur de pondération.

Suivant la valeur de r ($r \geqslant 1$) on obtient les mesures suivantes :

- $r = 1$: distance de Manhattan ;
- $r = 2$: distance Euclidienne ;
- $r = \infty : d_\infty(xi, xj) = \max_{1 \leqslant k \leqslant M} |x_{ik} - x_{jk}|$

Ces mesures sont souvent utilisées pour des données numériques. Dans le cas de données symboliques, d'autres distances doivent être utilisées. La plus connue étant la distance de Hamming, à l'origine définie pour mesurer la distance entre des chaînes binaires, et qui correspond au nombre de bits différents dans les deux chaînes pour une même position.

Définition 4

L'erreur quadratique est un critère des plus courants parmi les critères utilisés pour le partitionnement.

Considérons qu'une partition de O a été obtenue sous la forme de K classes $c_1, ..., c_K$ composées respectivement de $|c_1|, ..., |c_K|$ objets. Comme chaque objet appartient à une et une seule classe on a :

$$\sum_{i=1}^{k} |c_i| = N$$

Le centre de gravité g_i de la classe c_i est donné par :

$$g_i = \frac{1}{|c_i|} \sum_{j=1}^{|c_i|} x_j^{(i)}$$

Où $x_j^{(i)}$ est le j^{eme} point de la classe c_i et $|c_i|$ est le cardinal de c_i. L'erreur quadratique ε_i^2 sur la classe c_i est :

$$\varepsilon_i^2 = \sum_{j=1}^{|c_i|} d^2(x_j, g_i)$$

L'erreur quadratique (aussi appelée inertie intra-classe : I_W) de la partition est alors :

$$I_W = Eq^2 = \sum_{j=1}^{k} \varepsilon_j^2$$

Bien évidemment, ce critère ne peut servir à comparer des partitions ayant un nombre de classe différent. En effet, si $K = N$ alors $E_q = 0$ car $x_i = g_i \ \forall i \in 1, ..., N$ et l'erreur quadratique est minimale.

Le vecteur moyen g de l'ensemble O est donné par :

$$g = \frac{1}{N} \sum_{i=1}^{k} |c_i| g_i$$

Enfin, l'inertie interclasse, notée I_B, est donnée par :

$$I_B = \sum_{i=1}^{k} |c_i| d^2(g_i, g)$$

L'inertie totale I :

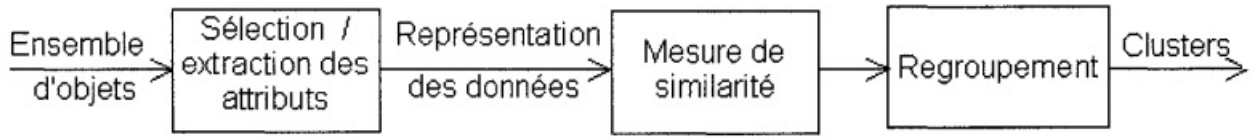
$$I = I_B + I_W$$

Est stable pour un nombre de classes K fixé.

1.1.3 Processus de clustering

Étant donné un ensemble d'objets $X = \{x_1, x_2, ..., x_n\}$ dans l'espace d'attributs $\mathbb{R}^d$ avec d : dimension de l'espace, n : le nombre d'objets. $x_i = \{x_{i1}, x_{i2}, ..., x_{id}\}$ représente le i^{eme} objets et x_{ij} la valeur du j^{eme} attribut pour la i^{eme} objet. Le but principale du clustering est de recherche des structures similaires dans l'espace d'objets $\mathbb{R}^d$. On constate que toutes les techniques de clustering suivent le même principe général qui consiste à maximiser la similarité des objets à l'intérieur d'un cluster, et minimiser la similarité des objets entre les clusters.

Les différentes étapes d'une tache de clustering sont [20] :

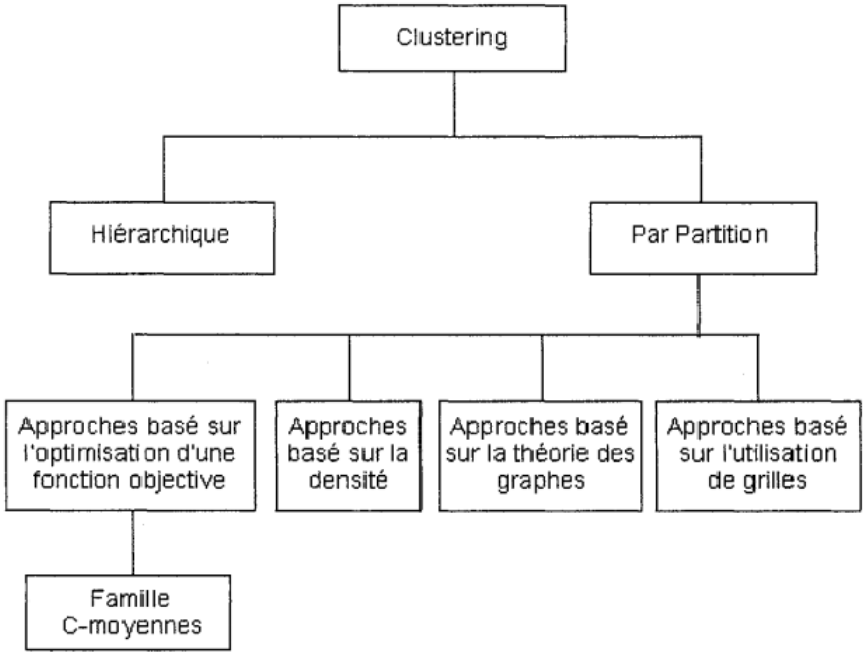


Telle que :

- La sélection/extraction des attributs correspond à l'utilisation d'une ou plusieurs transformations des attributs fournis en entrée afin de sélectionner le sous-ensemble le plus efficace à l'utilisateur pour le clustering.
- La représentation des données se réfère à la spécification du nombre de données, ainsi que la dimension et le type des disponibles pour l'algorithme de clustering.
- La mesure de similarité consiste à définir une métrique appropriée au domaine de données. La distance euclidienne est l'une des métriques les plus utilisées.
- Le regroupement consiste en la construction des groupes similaires, qui représente le résultat du processus de clustering.

1.2 Approches de clustering (Techniques) :

A travers le schéma suivant, on distingue deux grandes catégories des techniques de clustering [23,28] :



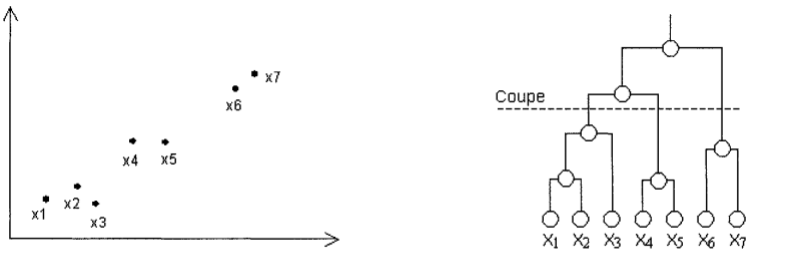
- Clustering hiérarchique : Dont le but est de former une hiérarchie de clusters, de telle sorte que chaque point où le cluster est progressivement "absorbé" par le cluster proche. Plus les clusters sont spécifiques à une similarité.
- Clustering par partitionnement : Dont le but est de former plusieurs partitions dans l'espace des objets, de telle sorte que chaque partition représente un cluster.

1.2.1 Clustering hiérarchique :

Différents algorithmes de clustering hiérarchique [6,7,28] ont été proposés dans la littérature. Toutes ces algorithmes partagent une caractéristique importante, ils ne produisent pas une seule partition mais une hiérarchie de partitions emboitées. Ici, un cluster est défini comme un noeud d'arbre auquel est associé l'ensemble des objets qui le composent, ainsi leurs caractéristiques.

Il existe deux catégories d'algorithmes hiérarchiques : les méthodes des ascendantes et les descendantes. Dans les méthodes des ascendantes ou agglomératives la partition initiale contient autant de clusters que d'objet ($c = n$). A chaque étape, on cherche un couple

(c_a, c_b) de cluster candidats à le fusion qui maximise (resp. minimise) une dissimilarités. On réitère ce processus jusqu'à n'obtenir qu'un cluster contenant tous les éléments. Afin de déterminer le nombre de clusters, on coupe la hiérarchie à un certain niveau de détail. La figure suivante illustre cette hiérarchie de partitions sous forme appelée dendogramme :



Un ensemble d'objets à classer *Dendogramme de la partition*

Principe du clustering hiérarchique .

L'un des avantages des techniques de clustering hiérarchique est de fournir via le dendogramme une interprétation naturelle du comportement de l'algorithme. A l'opposé, on est généralement confronté à un grand temps et surtout en espace. En effet, la création de la partition initiale avec un élément par cluster nécessite de calculer et de stocker les distances entre tous les couples de points. Cette complexité quadratique peut s'avérer critique pour les jeux de données de grande taille. Récemment, divers travaux ont été menés pour réduire la taille de la partition initiale.

Elle souffrent de l'effet de chaine : des clusters proches mais distincts peuvent êtres concaténés s'il existe une chaine de points qui les relie. On s'intéresse à la méthode de la classification ascendante, donnée par :

La classification ascendante hiérarchique (CAH) :

C'est une méthode de classification automatique utilisée en analyse des données, à partir d'un ensemble Ω de n individus, son but est de répartir ces individus dans un certain nombre de classes. De plus c'est une solution simple et populaire qui consiste à produire un dendrogramme pour voir si elle suggère un nombre particulier de clusters, qui est considéré quelque part le nombre optimal de clusters d'un ensemble d'individus. La méthode suppose qu'on dispose d'une mesure de dissimilarité entre les individus ; dans le cas de points situés dans un espace euclidien, on peut utiliser la distance comme mesure de dissimilarité. La dissimilarité entre des individus x et y sera notée dissim(x,y). La

classification ascendante hiérarchique est dite ascendante car elle part d'une situation où tous les individus sont seuls dans une classe, puis sont rassemblés en classes de plus en plus grandes. Le qualificatif "hiérarchique" vient du fait qu'il produit une hiérarchie H, l'ensemble des classes à toutes les étapes de l'algorithme, qui vérifie les propriétés suivantes :

- $\Omega \in H$: au sommet de la hiérarchie, lorsqu'on groupe de manière à obtenir une seule classe, tous les individus sont regroupés,
- $\forall \omega \in \Omega$, $\{\omega\} \in H$: en bas de la hiérarchie, tous les individus se trouvent seuls,
- $\forall (h, h') \in H^2$, $h \cap h' = \emptyset$ ou $h \subset h'$ ou $h' \subset h$

Algorithme :

- Principe : Initialement, soit n classes chaque individu forme une classe. On cherche à réduire le nombre de classes à $nbClasses < n$, ceci se fait itérativement. À chaque étape, on fusionne deux classes, réduisant ainsi le nombre de classes. Les deux classes choisies pour être fusionnées sont celles qui sont les plus "proches", en d'autres termes, celles dont la dissimilarité entre elles est minimale, cette valeur de dissimilarité est appelée indice d'agrégation. Comme on rassemble d'abord les individus les plus proches, la première itération a un indice d'agrégation faible, mais celui-ci va croître d'itération en itération.

- Mesure de dissimilarité inter-classe : La dissimilarité de deux classes $C_1 = \{x\}$, $C_2 = \{y\}$ contenant chacune un individu se définit simplement par la dissimilarité entre ces individus. $dissim(C_1, C_2) = dissim(x, y)$

Lorsque les classes ont plusieurs individus, il existe de multiples critères qui permettent de calculer la dissimilarité. Les plus simples sont les suivants :

- Le saut minimum retient le minimum des distances entre individus de C_1 et C_2 : $dissim(C_1, C_2) = \min\limits_{\substack{x \in C_1 \\ y \in C_2}}(dissim(x, y)) = \inf\limits_{x \in C_1} \inf\limits_{y \in C_2} d(x, y) = D(C_1, C_2)$
- Le saut maximum est la dissimilarité entre les individus de C_1 et C_2 les plus éloignés : $dissim(C_1, C_2) = \max\limits_{\substack{x \in C_1 \\ y \in C_2}}(dissim(x, y)) = \max\limits_{x \in C_1} \max\limits_{y \in C_2} d(x, y)$
- Le lien moyen consiste à calculer la moyenne des distances entre les individus de C_1 et C_2 : $dissim(C_1, C_2) = \frac{1}{|C_1|+|C_2|} \sum\limits_{\substack{x \in C_1 \\ y \in C_2}} (dissim(x, y))$
- La distance de Ward vise à maximiser l'inertie inter-classe :

$dissim(C_1, C_2) = \frac{n_1 * n_2}{n_1 + n_2} dissim(G_1, G_2)$ avec n_1 et n_2 les effectifs des deux classes, G_1 et G_2 leurs centres de gravité respectifs.

- Implémentation en pseudo-code :

Entrées :

- individus : liste d'individus
- nbClasses : nombre de classes que l'on veut finalement obtenir

Sortie :

- classes : liste de classes initialement vide, une classe est vue comme une liste d'individus.

Pour i=1 à individus.longueur Faire

classes.ajouter(nouvelle classe(individu[i])) ;

Fin Pour

Tant Que classes.longueur > nbClasses Faire

// Calcul des dissimilarités entre classes dans une matrice triangulaire supérieure

matDissim = nouvelle matrice(classes.longueur,classes.longueur) ;

Pour i=1 à classes.longueur Faire

Pour j=i+1 à classes.longueur Faire

matDissim[i][j] = dissim(classes[i],classes[j]) ;

Fin Pour

Fin Pour

// Recherche du minimum des dissimilarités

Soit (i,j) tel que : matDissim[i][j] = min(matDissim[k][l]) avec $1 \leqslant k \leqslant classes.longueur$ et $k + 1 \leqslant l \leqslant classes.longueur$;

// Fusion de classes[i] et classes[j]

Pour tout élément dans classes[j] Faire

classes[i].ajouter(élément) ;

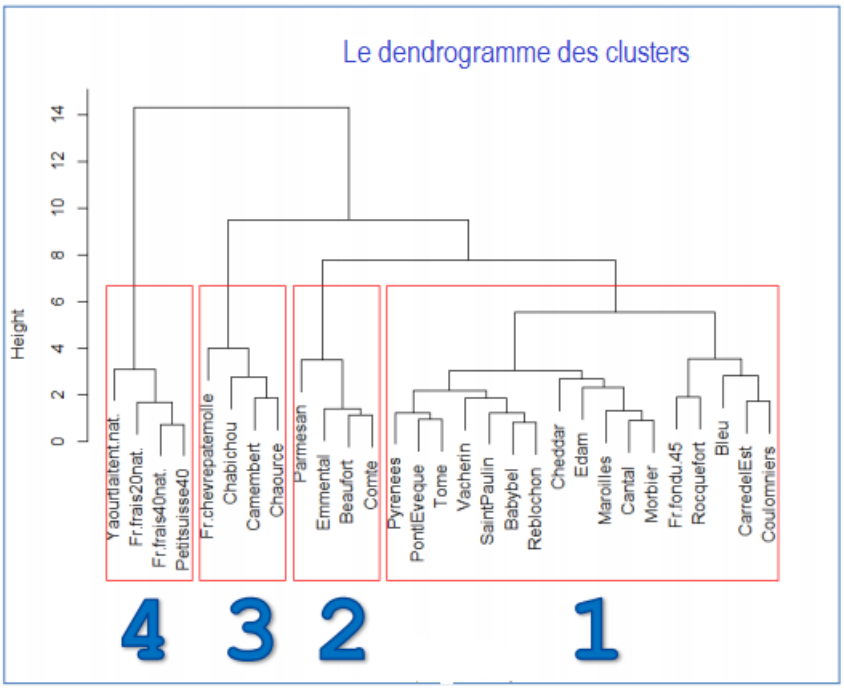
Fin pour

supprimer(classes[j]) ;

Fin Tant Que

Dendrogramme :

Soit un exemple simple de la méthode CAH :



Le dendrogramme nous suggère qu'ils existe quatre groupes, qui on peut le considérer comme un nombre optimale que on peut classer les fromages :

- Le 4^{eme} groupe est constitué de fromages frais.
- Le 3^{eme} de fromages à pâte molle.
- Le 2^{nd} de fromages « durs ».
- Le 1^{er} est un peu fourre-tout.

1.2.2 Clustering par partition

Contrairement au clustering hiérarchique, le clustering par partition [5,17,20,2Z8] a pour but de trouver une seul partition d'espace d'objet, de telle sorte qu'elle soit la plus pertinente pour la formation des clusters. Des algorithmes appartenant à cette catégorie sont présentés dans ce qui suit :

a)- Algorithmes basés sur la densité :

Le but ici est de chercher à former des clusters denses, de telle sorte que chaque cluster représente une région homogène de haute densité, entouré par des régions de faible densité. Pour cela, deux paramètres qui contrôlent la densité sont.

- Eps : Rayon maximum du voisinages,

– Minpts : Nombre minimum de points qui doivent être contenus dans ce voisinage.

Le voisinage d'un objet est définie comme suit : $V_{Eps} := \{x_j \in X / dist(x_i, x_j) \leq Eps\}$

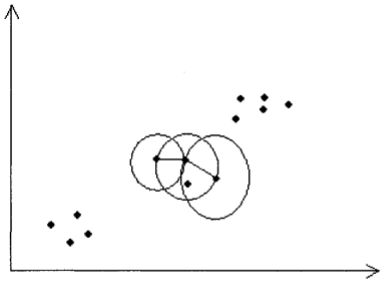
Le principe général est décrit comme suit :

(1)- Sélectionner aléatoirement un objet x_i selon une loi uniforme sur les objets.

(2)- Vérifier si son voisinage respecte le critère de densité ; c'est à dire s'il y a au moins Minpts points dans la sphère de centre x_i et de rayon Eps.

(3)- Si le critère de densité est respecté, intégrer les objets correspondant dans le cluster, et répéter le procédé avec ses objets.

(4)- Sinon, aller à 1 (la sélection aléatoire se fait sur les objets non encore classés).



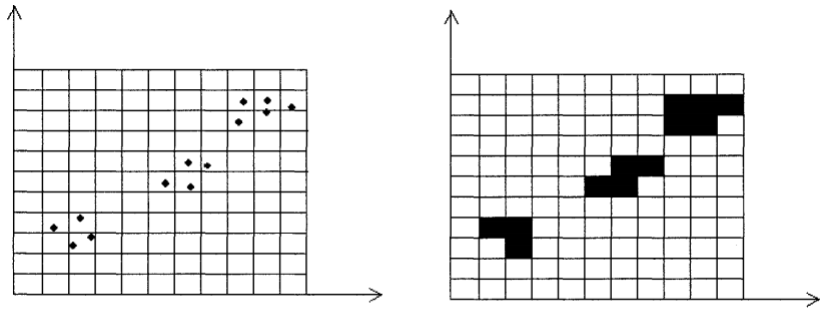
Exemple de clustering basé sur la densité

b)- Algorithmes basés sur les grilles :

Le principe de base de ces algorithmes est d'utiliser une grille pour diviser l'espace en un ensemble de cellules, ensuite identifier les ensembles de cellules denses connectées pour former les clusters. Ici, un cluster est vu comme un ensemble de cellules denses et connectées. Il existe deux méthodes pour identifier un cluster :

– Les méthodes qui calculent la densité de chaque cellule, puis fusionne les cellules pour que le résultat soit suffisamment dense et uniforme. La figure sous-dessus est une illustration graphique de ces méthodes.

– Les méthodes qui se basent sur la détection des limites du clusters. Le principe de base ici est la détection des limites entre les zones de haute densité et les zones de faible densité, ensuite la reconstitution des clusters à partir de ces limites.



Division de l 'espace à base d'une grille *le clustering associé*

Clustering basé sur les grilles

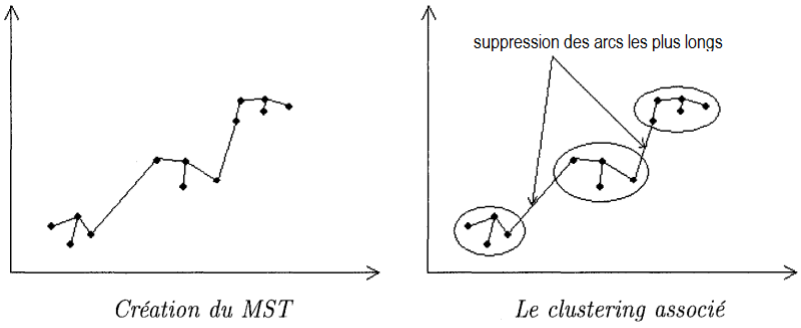
La majorité des algorithmes appartenant à cette catégorie souffrent d'une problématique importante, qui est le choix de la taille des cellules.

c)- Algorithmes basés sur la théorie des graphes :

Le principe de ces algorithmes est de rechercher des arcs à conserver, dans un graphe qui connecte entre eux afin de former des clusters. Ici un cluster est défini comme un ensemble de noeuds connectés dans un graphe.

Dont le principe général est décrit comme suite :

- construction d'un MST "Minimal Spanning Tree" de données, cela revient à définir un graphe connexe en joignant tous les objets de la base, dont la somme des valeurs des étiquettes associées aux arcs minimale.
- Suppression des arcs les plus longs pour la création des clusters.



Création du MST *Le clustering associé*

Clustering basé sur la théorie des graphes.

Une autre possibilité consiste à conserver les liens entre les objets séparés par une distance inférieure à un certain seuil, les clusters étant alors l'ensemble des objets connectés.

d)- Algorithmes basés sur la minimisation d'une fonction objective :

Parmi les différentes techniques les plus populaires et les plus utilisées en clustering, jusqu'ici, les algorithmes basés sur l'optimisation d'une fonction objective et spécialement ceux appartenant à la famille des K-means, représente l'une des techniques les plus populaires et les plus utilisées en clustering. Comme toutes les techniques de clustering, le but principal des algorithmes appartenant à cette famille est de diviser une partition en un ensemble de groupes partages les mêmes caractéristiques.

d. 1)- Méthode des K-means [Macqueen, 1967]

L'algorithme des K-means [11,25,32] est un outil de classification classique qui permet de répartir un ensemble de données en K classes homogènes. La plupart des images (photos, dessins vectoriels 2D, synthèses 3D, ...) vérifient localement des propriétés d'homogénéité notamment en terme d'intensité lumineuse. L'algorithme des K-means permet donc d'apporter une solution à la segmentation d'images.

Cette méthode fut longtemps utilisé sur les grands jeux de données en raison de sa rapidité. On s'intéresse tout d'abord à l'algorithme même, puis à ses propriétés.

Principe :

L'idée principale est de définir les k centroïdes arbitraires $\{c_1, c_2, ..., c_k\}$ (ou k est le nombre de clusters fixé a priori et chaque c_i représente le centre d'une classe), ces centroïdes doivent être placés dans des emplacements différents. Donc, le meilleur choix est de les placer le plus possible éloignés les uns des autres. La prochaine étape est de prendre chaque point appartenant à l'ensemble de données et l'associer au plus proche centroïde. C-à-d chaque classe P_i sera représentée par un ensemble d'individus les plus proches de son centre c_i.

Lorsqu'aucun point n'est en attente, la première étape est terminée et un groupage précoce est fait. À ce point nous avons besoin de recalculer les k nouveaux centroïdes m_i des groupes issus de l'étape précédente qui vont remplacer les c_i (g_j est le centre de gravité de la classe P_j, calculé en utilisant les nouvelles classes obtenues). Après, on réitère le processus jusqu'à atteindre un état de stabilité où aucune amélioration n'est plus possible, nous pouvons constater que les k centroïdes changent leur localisation par

étape jusqu'à plus de changements sont effectués. En d'autres termes les centroïdes ne bougent plus.

Description de l'algorithme :

L'algorithme k-means est l'algorithme de clustering le plus connu et le plus utilisé, du fait de sa simplicité de mise en oeuvre.

L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde. Le k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. K-means change les objets de la classe jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de classe compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre des classes. Sur le plan algorithmique, l'algorithme des K-means se résume de la manière suivante :

Agorithme : K-means

Dans la méthode des "K-means", le choix des centres initiaux s'effectue sur la base d'un tirage aléatoire sans remise de k individus à partir de la population à classifier. La partition des classes est modifiée avec chaque affectation d'un individu i de X (X ensemble de données). Les individus sont géométriquement représentés dans l'espace vectoriel $\mathbb{R}^P$ muni d'une distance notée d. L'algorithme de la méthode des "K-means" se déroule comme suit :

Étape 1 :

- On choisit par un tirage aléatoire sans remise k individus parmi n individus composant l'ensemble X. Ces k centres notés $\{c_1^0, c_2^0, ..., c_k^0\}$ sont provisoires.
- Chaque individu i de X est affecté à une classe et une seule. Chacune de ces classes est localisée par son centre. La procédure d'affectation est la suivante : i est affecté à la classe notée P_i^0 de centre c_i^0 si et seulement si $d(i, c_i^0) = \inf_{j \in \{1,2,...,k\}} \{d(i, c_i^j)\}$.

Après avoir affecté tous les individus on obtient k classes notées $\{P_1^0, P_2^0, ..., P_k^0\}$ de centres respectifs $\{c_1^0, c_2^0, ..., c_k^0\}$.

Étape 2 :

En considérant les k classes obtenues à l'étape 1, on calcule ses centres de gravité. On obtient donc k nouveaux centres notés $\{c_1^1, c_2^1, ..., c_k^1\}$. On utilise la même règle d'affectation qu'à l'étape 1, on obtient k nouveaux classes $\{P_1^1, P_2^1, ..., P_k^1\}$ de centres respectifs $\{c_1^1, c_2^1, ..., c_k^1\}$.

Étape h :

On détermine k nouveaux classes en calculant les centres de gravité des classes obtenues à l'étape (h - 1). La règle d'affectation reste la même qu'à l'étape précédente et on obtient par la suite une nouvelle typologie de l'ensemble X : $\{P_1^h, P_2^h, ..., P_k^h\}$ de centres respectifs $\{c_1^h, c_2^h, ..., c_k^h\}$.

Test d'arrêt de l'algorithme :

L'arrêt de l'algorithme de la méthode des "K-means" se fait :

- Lorsque deux itérations successives conduisent à une même partition.
- Lorsqu'on fixe un critère d'arrêt tel que le nombre maximal d'itérations.

Discussion :

Cette méthode est la plus populaire des méthodes de clustering, malgré ça, un de ses problèmes majors est qu'il tend à trouver des classes sphériques de même taille. Il est donc fréquemment faire appeler une heuristique en pratique, ce qui explique qu'elle est convergente et surtout avantageuse du point de vue calcul mais elle dépend essentiellement de la partition initiale (Des initialisations différentes peuvent mener à des clusters différents problèmes de minima "locaux") cela risque d'obtenir une partition qui ne soit pas optimale pourtant qu'elle donne surement une partition meilleure que la partition initiale. De plus, la définition de la classe se fait à partir de son centre, qui pourrait ne pas être un individu de l'ensemble à classer, d'où le risque d'obtenir des classes vides.

d. 2)- Méthode PAM : (Partitioning Around Medoids)

L'algorithme de PAM [8,16,32] est très similaire à l'algorithme K-means, surtout parce que les deux sont des algorithmes de partitionnement. Une taille ensemble du cluster est déterminée et une première série de centres de cluster est établi avant le début de cette technique de clustering. Contrairement à K-means, ces centres sont medoids (medoids sont semblables dans le concept à moyen ou centroïdes, mais medoids sont toujours membres de l'ensemble de données) et sont toujours des observations de l'ensemble de données. Une fois que chaque observation est attribuée au cluster médoïde le plus proche, les nouveaux medoids sont choisis afin de minimiser une somme de similitudes et réaffectation des observations à la médoïde la plus proche se produit. Ce processus se répète jusqu'à ce qu'il n'y a pas de nouvelles affectations. En d'autres termes, à la fois briser l'ensemble de données en groupes (clusters), et les deux travaux par en essayant de minimiser l'erreur, mais PAM travaille avec Medoids, qui sont une entité de l'ensemble de données qui représentent le groupe dans lequel il est inséré, et K-means travaille avec centroïdes, qui sont créés artificiellement entité qui représente son groupe.

Les partitions de l'algorithme PAM l'ensemble de données de n objets en k clusters, où à la fois l'ensemble de données et le nombre k est une entrée de l'algorithme. Cet algorithme fonctionne avec une matrice de dissemblance, dont le but est de réduire au minimum la dissemblance globale entre les représentants de chaque groupe et de ses membres. L'algorithme utilise le modèle suivant pour résoudre le problème :

$$F(x) = minimiser \sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij} d(i,j)$$

Sujet à :

(1.) $\sum_{i=1}^{n} z_{ij} = 1,\ j = 1, 2, ..., n$

(2.) $z_{ij} \leqslant y_i,\ i, j = 1, 2, ..., n$

(3.) $\sum_{i=1}^{n} = k$, k = nombre de clusters

(4.) $y_i, z_{ij} \in \{0, 1\}, \quad i = 1, 2, ..., n$

Où F(x) est la fonction principale de minimiser, d(i,j) est la mesure de la dissimilarité entre les entités i et j, et z_{ij} est une variable qui garantit que seule la différence entre les entités du même groupe sera calculée en la fonction principale. Les autres expressions sont contraintes qui ont les fonctions suivantes : (1.) veille à ce que chaque entité unique est

affecté à un cluster et un seul cluster, (2.) veille à ce que l'entité est affectée à son médoïde qui représente le cluster, (3.) assure qu'il y a exactement k grappes et (4.) permet aux variables de décision supposent seulement les valeurs de 0 ou 1.

L'algorithme de PAM peut travailler plus de deux types d'entrée, la première est la matrice représentant toutes les entités et les valeurs de ses variables, et la seconde est directement la matrice de dissemblance, dans ce dernier, l'utilisateur peut fournir la dissemblance directement comme une contribution à la algorithme, au lieu de la matrice de données représentant les entités. De toute façon l'algorithme atteint une solution au problème, dans une analyse générale, l'algorithme procède de cette façon :

Phase de construction :

1.) Choisissez k entités pour devenir les medoids, ou dans le cas où ces entités ont été fournies les utilisent comme les medoids ;

2.) Calculer la matrice de dissemblance si elle n'a pas été informé ;

3.) Attribuez chaque entité à son médoïde plus proche ;

Phase de Swap :

4.) Pour chaque recherche de cluster si l'une des entités du pôle abaisser le coefficient de dissimilarité moyenne, si elle le fait de sélectionner l'entité qui abaisse ce coefficient le plus que le médoïde pour ce groupe ;

5.) Si au moins une médoïde a changé, aller à (3), sinon mettre fin à l'algorithme.

L'algorithme de PAM travaille avec une matrice de dissemblance, et de calculer cette matrice l'algorithme peut utiliser deux métriques. Le premier est le euclidienne, qui sont la racine de la somme des carrés des différences, tandis que le second est la distance Manhattan qui sont la somme des distances absolues.

Agorithme : PAM

L'algorithme PAM se décompose en deux parties. Dans un premier temps l'initialisation et dans un deuxième temps la recherche des meilleurs médoïdes.

Première étape : Initialisation

1 : Le premier centre de gravité choisi est l'élément pour lequel la somme des distances par rapport à tous les autres éléments est la plus petite.

2 : On définit chaque nouveau centre c_i comme l'élément parmi tous les éléments non

médoïdes, qui maximise la fonction objective.

Deuxième étape : On recherche les meilleurs médoïdes

Dans PAM, chaque classe est représentée par l'un de ces médoïdes. L'algorithme de PAM repose sur le principe suivant : à partir d'un ensemble de k représentants $M_1, ..., M_k$, choisis parmi les n objets à classer :

1 : Sélectionner au hasard un représentant M_i et un autre objet (non représentant) O_j.

2 : Calculer la qualité de la nouvelle partition si les rôles de M_i et O_j sont inversés.

3 : Échanger M et O si la qualité est supérieure.

4 : Et retourner en 1, jusqu'à stabilité de la qualité de la partition.

Discussion :

La méthode PAM possède plusieurs avantages, dont sa robustesse en présence de valeurs extrêmes. C'est plus avancée par rapport aux méthodes antérieures. Par contre, PAM ne sera pas aussi efficace pour un grand nombre d'observations. De plus, les calculs sont très complexes, l'algorithme devient trop coûteux pour des valeurs de n et k importantes.

Même si elle ne permet pas de travailler avec de grandes quantités de données, l'algorithme PAM demeure un algorithme de référence de l'implémentation des k-medoïdes.

La question d'utiliser d'autres méthodes vient donc tout naturellement. Ces dernières visent à augmenter la vitesse de l'algorithme, mais en étant moins précises.

1.3 Entropie de Shannon :

La théorie statistique de l'information de C. Shannon [10] appelé souvent à tort théorie de l'information ou théorie mathématique de la communication, est souvent réduite est connue en SIC (Sciences de l'information et de la Communication) à travers le schéma du système général de la communication : source, émetteur, signal ... bruit. La théorie de Shannon est connue en statistique par sa célèbre formule de l'entropie. Cette théorie est important car elle est à la jonction de la théorie du signal de la statistique unidimensionnel

et bidimensionnel. Nous essaierons à travers de cette étude de concentrer sur la notion de l'entropie applicable en statistique plutôt en classification.

1) Une approche d'entropie :

L'entropie de Shannon, notée S ou S_n mesure la qualité d'information moyenne. Soient $i \in I$ et E_i une partition de l'ensemble E par le caractère I (tq : I est un caractère donné), où on note la distribution des fréquences par : $P_i = \frac{|E_i|}{|E|}$. La qualité d'information moyenne est donnée par :

$$S(I) = -\sum_i P_i Log(P_i)$$

S(I) est aussi appelée entropie de la partition de E définie par I.

Remarque :

On remarque que S(I) ne dépend pas du caractère I, ni même du type de ces modalités, mais uniquement de la distribution des fréquences P_i.

2) La démarche de shannon :

Voici rapidement les hypothèses que formule Shannon. Supposons que nous ayons un ensemble de n événements possibles dont les probabilités d'occurrence sont : $P_1, P_2, ..., P_n$. Comment trouver une mesure de l'incertitude du résultat, c'est à dire du nombre de choix possibles ? Les probabilités sont connues apriori et c'est tout ce que nous connaissons sur le future. Si tous les événements sont équiprobables il est raisonnable de considérer qu'il est souhaitable que l'incertitude soit maximal.
Shannon impose à cette mesure S trois conditions :

- S est une fonction continue des P_i
- Si tous les P_i sont égaux. Alors, S est une fonction monotone croissante de n.
- Si un choix se décompose en deux choix successifs le S original devra être la somme pondérée des valeurs individuelles.

Shannon montre que la seule fonction S satisfaisante aux trois hypothèses ci-dessous est de la forme :

$$S = k\sum_{i=1}^{n} P_i Log(P_i)$$

Conclusion :

Dans ce chapitre, on a présenté les concepts et quelques techniques de base utilisés pour la détermination de l'indice de validité, et une vue globale du principe fondamentale de clustering, aussi les différentes techniques de clustering utilisées. Dans tous les approches, les algorithmes souffrent du problème du choix du bon nombre de clusters qui est souvent laissé à l'utilisateur.

Dans ce contexte, il est utile de présenter concrètement des outils permettant d'estimer le bon nombre de clusters dans un ensemble de données. Ces outils appelés indices de validité qui sont présentés dans le chapitre trois.

Chapitre 2

Clustering en statistique décisionnelle :

Le clustering a été utilisées dans plusieurs domaines, allant de l'ingénierie (apprentissage automatique, intelligence artificielle, reconnaissance des formes, génie mécanique, génie électrique), l'informatique, sciences médicales et vie, sciences de la terre, sciences sociales, et l'économie. Cette diversité reflète la position importante du regroupement dans la recherche scientifique. D'autre part, cette diversité peut être une source de confusion, en raison des terminologies et objectifs différents. Les algorithmes de clustering ont été développés pour résoudre des problèmes particuliers, dans des domaines spécifiques, et ils sont généralement basés sur des hypothèses et des suppositions sur l'ensemble de données à traiter. Ces suppositions affectent inévitablement les performances de ces algorithmes dans d'autres problèmes qui ne satisfont pas ces hypothèses. Par exemple, l'algorithme K-means basé sur la distance euclidienne et, par conséquent, il tend à générer des clusters hyper sphériques. De plus, le clustering joue un rôle important dans toutes les sciences et techniques qui font appel à la statistique décisionnelle qui n'est qu'un outil d'aide à la décision, permet de décrire synthétiquement des données, et de tester les hypothèses relatives à une situation, et plus généralement de traiter toutes sortes de données qu'elles soient qualitatives ou quantitatives.

2.1 Le problème de décision :

Dans tous les domaines, de l'expérimentation scientifique à la vie quotidienne en particulier dans le clustering, on est amené à prendre des décisions sur une activité

risquée au vu de résultats d'expériences ou d'observation de phénomènes dans un contexte incertain.

Par exemple :

- Informatique : au vu des résultats des tests d'un nouveau système informatique, on doit décider si ce système est suffisamment fiable et performant pour être mis en vente.
- Essais thérapeutiques : décider si un nouveau traitement médical est meilleur qu'un ancien au vu du résultat de son expérimentation sur des malades.
- Finance : au vu du marché, décider si on doit ou pas se lancer dans une opération financière donnée.
- Justice : décider si l'accusé est innocent ou coupable à partir des informations acquises pendant le procès.

Dans chaque cas, le problème de décision consiste à trancher, au vu d'observations, entre une hypothèse appelée hypothèse nulle, notée H_0, et une autre hypothèse dite hypothèse alternative, notée H_1. En général, on suppose qu'une et une seule de ces deux hypothèses est vraie. Un test d'hypothèses est une procédure qui permet de choisir entre ces deux hypothèses.

Les conséquences des mauvaises décisions peuvent être d'importances diverses :

- Informatique : si on conclut à tort que le système n'est pas assez fiable et performant, on engagera des dépenses inutiles pour le tester et l'analyser et on risque de se faire souffler le marché par la concurrence ; si on décide à tort qu'il est suffisamment fiable et performant, on va mettre en vente un produit qui ne satisfera pas la clientèle, ce qui peut couter cher en image de marque comme en cout de maintenance.
- Essais thérapeutiques : on peut adopter un nouveau traitement moins efficace, voire pire que l'ancien, ou se priver d'un nouveau traitement plus efficace que l'ancien.
- Finance : si on décide à tort que l'on peut lancer l'opération, on risque de perdre Beaucoup d'argent ; si on décide à tort de ne pas lancer l'opération, on peut se priver d'un bénéfice important.
- Justice : on peut condamner un innocent ou acquitter un coupable.

2.2 La prise de décision en clustering :

Le clustering est un processus qui organise des objets en groupes dont les membres se ressemblent "d'une certaine manière". Ces techniques descriptives peuvent être mises en oeuvre sans savoir quelles seront les clusters obtenues, ni même quel est le nombre pertinent des clusters.

Il est donc impossible d'être totalement neutre. Dans ce sens en fait appel à la prise de décision [32] qui nous permet de prendre une décision sur ce nombre.

En clustering, on est amené à prendre des décisions sur le nombre de clusters au vu des approches et des résultats des méthodes de clustering dans le contexte de choisir le nombre optimal des clusters. Comme toute approche de clustering la qualité de ses résultats doit être validée. Valider les résultats de clustering implique immédiatement la recherche du bon nombre de clusters. Par le « bonne nombre de clusters » on veut dire que chacun de ces clusters doivent faire apparaitre une structure sous-jacente aux données et ainsi permettre de faciliter leur interprétation. Chaque cluster doit avoir une signification pour l'expert du domaine.

Domaines d'applications de clustering :

Dans cette partie, nous décrivons quelques domaines d'applications où le clustering a été utilisé comme une étape essentielle. Citons par exemple :

- Analyse des données qui peuvent provenir des images satellites équipement médical, imagerie médicale, systèmes d'informations géographiques, afin d'extraire les caractéristiques nécessaires pour accélérer le processus d'exploitation de ces données.
- Le clustering est fort utile également dans les sciences de l'homme : psychologie (classer les individus selon leur type de personnalités), sociologie, linguistique, archéologie, histoire.
- Médecine : Localisation de tumeurs dans le cerveau
  - Nuage de points du cerveau fournis par le neurologue.

- Identification des points définissant une tumeur.

– Et dans les techniques décrives comme :

- Les enquêtes d'opinion : comme planification de villes identification de groupes d'habitations suivant le type d'habitation, valeur, localisation géographique ...
- Marketing : segmentation du marché en découvrant des groupes de clients distincts à partir de bases de données d'achats.
- Assurance : identification de groupes d'assurés distincts associés à un nombre important de déclarations.
- Environnement : identification des zones terrestres similaires (en termes d'utilisation) dans une base de données d'observation de la terre.

Chapitre 3

Indices de validité

Dans ce chapitre, on va présenter quelques indices de validité existants dans la littérature pour déterminer le nombre de clusters dans la classification floue non supervisée. Dans un premier temps, on va décrire le principe fondamental des indices de validité. Dans ce contexte, une stratégie d'implémentation de différents indices de validité pour la recherche du bon nombre de clusters sera présenté. Sur un état de l'art des indices de validité la classification non supervisée.

La dernière partie de ce chapitre sera consacré aux indices de validité principaux existants dans la littérature.

3.1 Introduction :

L'objectif d'une tache de classification non supervisée consiste à proposer une partition des objets en k sous ensembles où k est le nombre de regroupements attendus par l'utilisateur. Une variation de cette tache est de ne pas utiliser le nombre attendu de regroupements comme une donnée de problème. Dans ce cas, l'algorithme construit plusieurs partitions candidates et choisit la meilleure partition qui est celle qui optimise un critère de qualité des partitions.

3.1.1 Définition :

Les résultats de classification obtenus fortement du nombre de classes fixé. Il est donc primordial de choisir le nombre exact de classes pour espérer avoir une bonne qualité de classification. Ceci n'est pas toujours simple, surtout en présence de chevauchement. Plusieurs approches ont été proposées sur ce sujet pour différentes applications.

Cependant, pour les mêmes données on peut obtenir des résultats différents selon le nombre de classes bien séparées, les algorithmes de classification retrouvent le même nombre de clusters en général. Le problème se pose dans le cas de chevauchement de classes : rares sont les algorithmes qui arrivent à détecter le nombre réel de classes, et ils deviennent invalides pour un degré de chevauchement relativement fort. Le processus d'évaluations des résultats des algorithmes de classification est appelé indice de validité des clusters.

3.1.2 Position de problème des indices de validité :

La recherche des structures similaires dans un ensemble de données X, dont on n'a pas d'information disponible sur les éventuels de regroupement qui le composent, rend la tache d'un algorithme de clustering de plus en plus difficile. En effet, dans les algorithmes de clustering par partition, le choix du nombre de clusters est généralement laissé à l'utilisateur. cependant, une question cruciale émerge : "combien existe-t-il de clusters dans un ensemble de données ?", mais également "existe-t-il une structure en cluster ?".
Considérons l'exemple illustré dans les figures suivantes :

Figure : 1

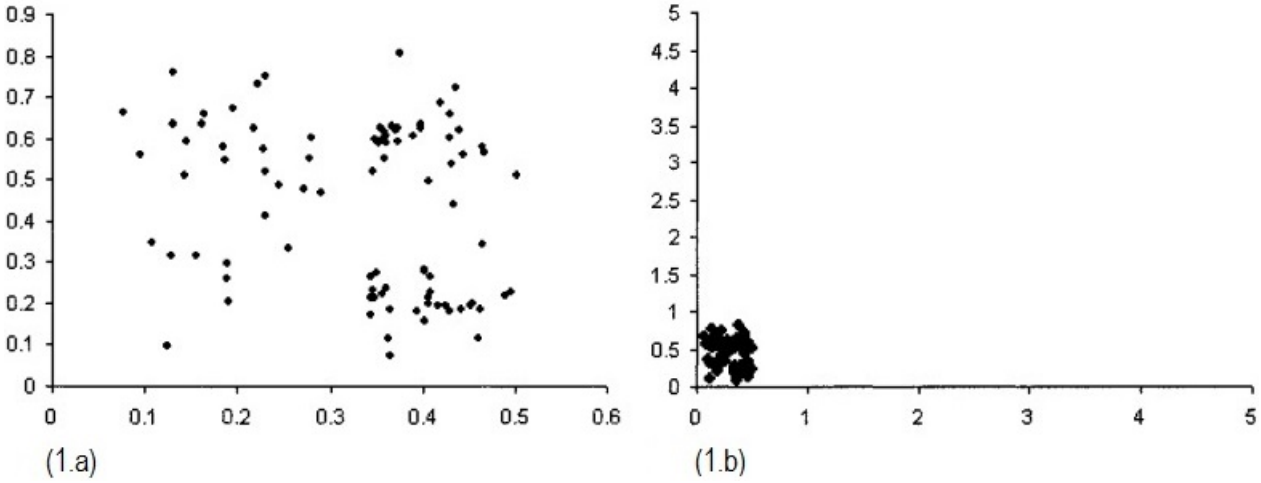
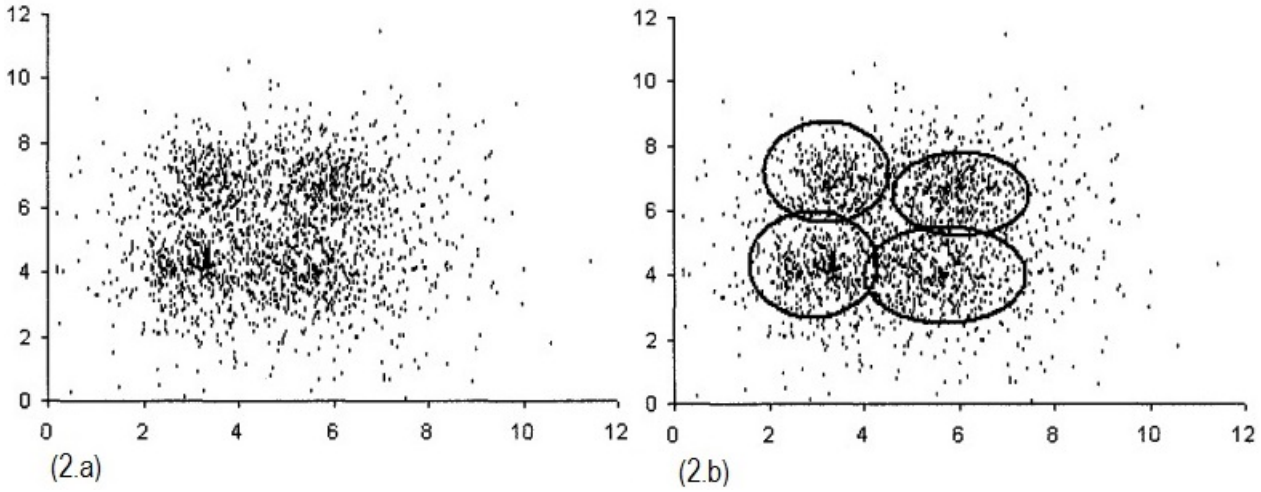


Figure : 2



Ces figures sont un exemple de difficulté rencontrée par un utilisateur pour déterminer s'il existe une structure en cluster ou non.

remarque : On remarque que les données sont visibles à niveaux d'échelle différents.

Visuellement, on a l'impression qu'il existe 3 clusters dans la figure $(1.a)$, alors que dans la figure $(1.b)$ on s'aperçoit qu'il y a un seul cluster. Dans la figure $(2.a)$ il est difficile pour un utilisateur de déterminer le nombre de clusters qui composent cet ensemble de données. Alors que dans la figure $(2.b)$ on constate que la réelle répartition de cet ensemble de données dans les figures sont représentées dans l'espace $\mathbb{R}^2$. La difficulté sera beaucoup plus élevé dans l'espace $\mathbb{R}^d$ ou $d > 3$ (on n'a pas un moyen de visualisation).

Le mode de fonctionnement non supervisé, représente la caractéristique principale d'un algorithme de clustering. Comme toute approche non supervisée, la qualité de ses

résultats doit être validée. Valider le résultat d'un algorithme de clustering implique immédiatement la recherche du bon nombre de clusters. Par "le bon nombre de clusters" on veut dire que chacun de ces clusters doivent faire apparaitre une structure sous-jacente aux données et ainsi permettre de faciliter leur interprétation. Chaque cluster doit avoir une signification pour l'expert du domaine.

Typiquement deux situations se présentent :

- Trop de clusters : cette situations se peut entrainer une grande confusion car certains clusters sont "artificiels", c'est à dire qu'ils ne représentent aucun réalité du domaine concerné.
- Pas assez de clusters : cet autre cas peut cacher des aspects importants présents dans les données. Par exemple de patients en deux groupes : les patients sains et malades. Mais il peut plus intéressant le médecin d'utiliser une structure en trois clusters faisant ressortir les patients sains, malades et à risque.

Face à ce problème, l'approche la plus utilisée consiste à :

1 - Exécuter les algorithmes de clustering avec différents nombres de clusters.

2 - Évaluer leurs résultats, et ce à partir d'une comparaison entre ces dernières.

L'évaluation des résultats est basé essentiellement sur l'utilisation des indices de validité.

Plus formellement, un indice de validité est une fonction qui mesure la qualité du résultat final d'un algorithme de clustering. Afin de trouver le nombre de clusters qui optimise (la plus petite ou la plus grande valeur) l'indice de validité en question, nous utilisons un processus itératif qui consiste à exécuter un algorithme de clustering avec différents nombres de clusters. Cependant, un problème crucial émerge, il concerne le choix d'un intervalle dont lequel le processus itératif de recherche doit être exécuté. Dans ce contexte, si nous faisons l'hypothèse initiale que chaque élément de l'ensemble de données X constitue un cluster, nous aurons un problème de taille énorme et un processus de recherche très compliqué. En plus de ça, dans la majorité des cas réels, le nombre de clusters est nettement inférieur aux nombre d'objets : $c << n$. L'hypothèse de limiter la recherche sur un intervalle bien défini, semble plus raisonnable et mieux adapté à ce genre de situation.

Généralement, le processus de recherche s'effectue entre $[C_{min}, C_{max}]$ avec :

- C_{min} : le nombre minimum de clusters

– C_{max} : le nombre maximum de clusters.

Dans la majorité des cas : $C_{min} \geqslant 1$, alors que le choix du C_{max} : il n'y a aucune règle formelle, quelques auteurs proposent de choisir $C_{max} = \sqrt{n}$.

3.2 Critères de validité

Dans le contexte de la classification automatique, il est naturel de s'interroger sur la validité de la partition obtenue. Les groupes découverts correspondent-ils à nos connaissances à priori ? Correspondent-ils vraiment à l'ensemble d'objets dont on dispose ? De deux classifications, laquelle est la plus pertinente ?

Ces différentes questions permettent de distinguer trois catégories de critères :

- **Les critères externes :** permettent de répondre à la première question et de mesurer l'adéquation entre une partition et les connaissances à priori dont on dispose.
- **Les critères internes :** quantifient l'adéquation entre une partition et l'idée subjective que l'on se fait d'une "bonne" classification. Ainsi, les propriétés les plus communément recherchées sont la compacité et la séparabilité des groupes découverts.
- **Les critères relatifs :** s'intéressent à la troisième question et à défaut de donner une appréciation absolue de la validité d'une partition, ils permettent d'ordonner plusieurs classifications et d'en choisir "une meilleure".

Les deux premiers critères sont basés sur des méthodes statistiques et demandent beaucoup de temps de calcul. Beaucoup des techniques sont basées sur le critère relatif [1,3].

Dans la partie suivante nous nous intéressons qu'aux indices les plus utilisés dans la littérature.

3.3 Historique des indices de validité :

Malgré toutes ces améliorations l'observateur humain reste limité et il ne peut pas évaluer la qualité de la classification surtout pour définir le nombre exact des classes. Pour

ces raisons une étape de validation a été introduite pour aider à fixer le nombre optimal de classes. Les calculs des indices de validités [2,12,24] sont effectues pour chaque méthode et pour chaque partition. Une étape de fusionnement est aussi ajoutée pour corriger quelque faute de classification.

En pratique, on distingue deux types de partition : dure "Hard" et floue "Fuzzy", d'où la nécessite d'utilisation des indices de validité conformes à ces différents types de partitions.

Il y a deux types d'indices de validité :

- Les indices dédiées à évaluer la validité des partitions "dures".
- Les indices dédiées à évaluer la validité des partitions floues, ce qui constituera l'objectif de cette partie.

Dans la suite de ce chapitre, on va donner quelques indices de validité principaux existent dans la littérature.

3.3.1 Les indices de Bezdek :

Ces deux indices utilisent les propriétés des degrés d'appartenance U_{ik} pour évaluer une partition.

V_{PC} : "Partition Coefficient" et V_{PE} : "Partition Entropy" sont les premiers indices de validité dédiés à la classification floue non supervisée proposés par Bezdek [20].

- Partition Coefficient V_{PC} (1974)

$$V_{PC} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} U_{ik}^2$$

Sous la contrainte que $\forall i$:

$$U_{ik} \in [0,1], \quad \sum_{i=1}^{n} U_{ik}^2 = 1$$

On a donc

$$\frac{1}{n} \leqslant V_{PC} \leqslant 1$$

Si la valeur de V_{PC} tend vers son maximum 1, pour un certain nombre de clusters c, on aura une partition qui est constituée de clusters bien séparés. Si la partition en question

ne contient aucune structure de clusters, V_{PC} atteint sa valeur minimale $\frac{1}{n}$. Il est claire que le nombre de clusters c qui maximise V_{PC} indique le nombre optimale de clusters.

- **Partition Entropy** V_{PE} (1975)

$$V_{PE} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c} U_{ik}^2 log_a(U_{ik})$$

Ou : $a \in (1, \infty)$ représente la base logarithmique.
La valeur de V_{PE} est comprise entre :

$$0 \leqslant V_{PE} \leqslant log_a(c)$$

Bezdek a démontré la relation qui lie les deux indices V_{PE} et V_{PC}, elle est définit comme suite :

$$0 \leqslant 1 - V_{PC} \leqslant V_{PE}$$

C'est a dire :

$$V_{PC} = 1 \Leftrightarrow V_{PE} = 0 \qquad (*)$$

A partir de (*) on peut déduit que si V_{PE} tend vers son minimum 0, on aura une partition constituée de clusters bien séparés.

Alors que s'il atteint sa valeur maximale $log_a(U_{ik})$ la partition en question n'a aucune structure de clusters. Il est clair que le nombre de clusters qui minimise V_{PE} indique le nombre optimale de clusters.

3.3.2 Les indices basés sur la compactness et la séparation :

L'évaluation du résultats d'un algorithme de clustering par l'intermédiaire des indices de validité appartenant à cette catégorie est basée essentiellement sur deux facteurs :

- La cohésion interne ou "compactness" : Pour une partition aussi pertinente que possible les uns des autres afin de former des structures compactes. L'idée ici est de maximiser la similarité entre les objets de même cluster.
- L'isolation externe ou séparation : l'objectif ici est de maximiser la distance entre les points représentant les clusters (un cluster est représenté par son prototype).

Parmi les indices appartenant à cette catégorie citons :

- **L'indice de Wemmert et Gançarski V_{WG} :**

L'indice Wemmert et Gançarski considèrent à la fois la compacité et la séparabilité des groupes et s'appuient sur le rapport entre deux distances [26] : la distance d'un objet au centre de son groupe et la distance minimale au centre d'un autre groupe. Il se définit ainsi pour un groupe :

$$V_{WG}(c_i) = \max\{0; 1 - \frac{1}{N_i} \sum_{x \in c_i} \frac{||x - w_i||}{\min_{j \neq i}\{||x - w_j||\}}\}$$

Avec : N_i est le nombre des éléments de la classes c_i, et w_i est les centre de la classe c_i.

Et la valeur de cet indice pour une partition correspond à la moyenne pondérée de l'indice de chacun des groupes :

$$V_{WG} = \frac{1}{N} \sum_{i=1}^{k} N_i \times V_{WG}(c_i)$$

- **L'indice de Davies-Bouldin V_{BD} (1979)**

L'indice de Davies-Bouldin [9] tient compte à la fois de la compacité et de la séparabilité des groupes. La valeur de cet indice est d'autant plus faible que les groupes sont compacts et bien séparés.

L'expression de cet indice est la suivante :

$$V_{BD} = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i}\{D_{ij}\}$$

Avec :

$$D_{ij} = \frac{\overline{d}_i + \overline{d}_j}{d_{ij}}$$

Ou :

$\overline{d}_i$ est la distance moyenne des membres du cluster i vers son centre g_i :

$$\overline{d}_i = \frac{\sum_l ||x_l - g_i||_d}{N_i}$$

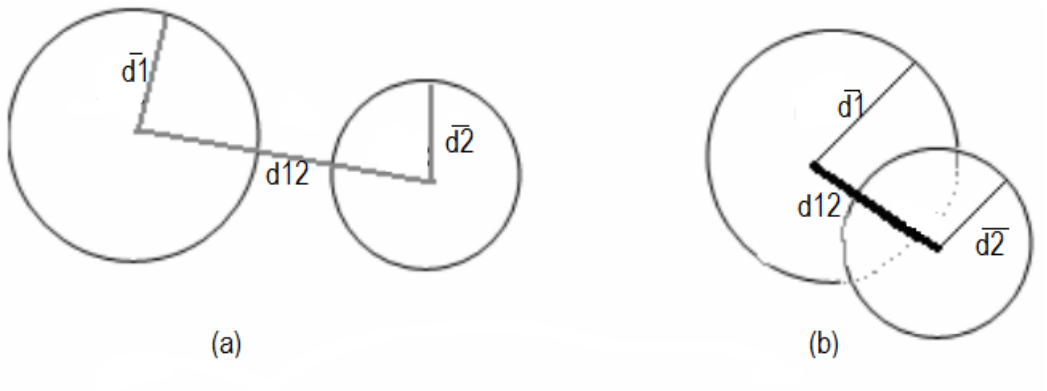
N_i : le nombre des éléments dans la classe i.

d_{ij} est la distance entre le centre du cluster i g_i et celui de j g_j

$$d_{ij} = ||g_i - g_j||_d$$

Et K le nombre de cluster à former.

- Si $\overline{d}_1 + \overline{d}_2 < d_{12}$ donc le couple $(1, 2)$ est bien séparé. le cas de la figure (a).
- Si $\overline{d}_1 + \overline{d}_2 > d_{12}$ donc le couple $(1, 2)$ n'est pas séparé. le cas de la figure (b).



- L'indice de Dunn : [1973]

L'indice de Dunn [15] est basé sur l'identification de clusters compacts et bien séparés. Il est défini par le rapport entre la plus petite dissimilarité inter-classe (i.e. entre deux individus de deux classes différentes) et la plus grande dissimilarité intra-classe (i.e. entre deux individus de la même classe). la valeur de cet indice est plus élevée que les groupes sont compacts et bien séparés.

$C = \{c_1, c_2, ..., c_k\}$ où k est le nombre des classes. L'indice de Dunn prend la forme suivante :

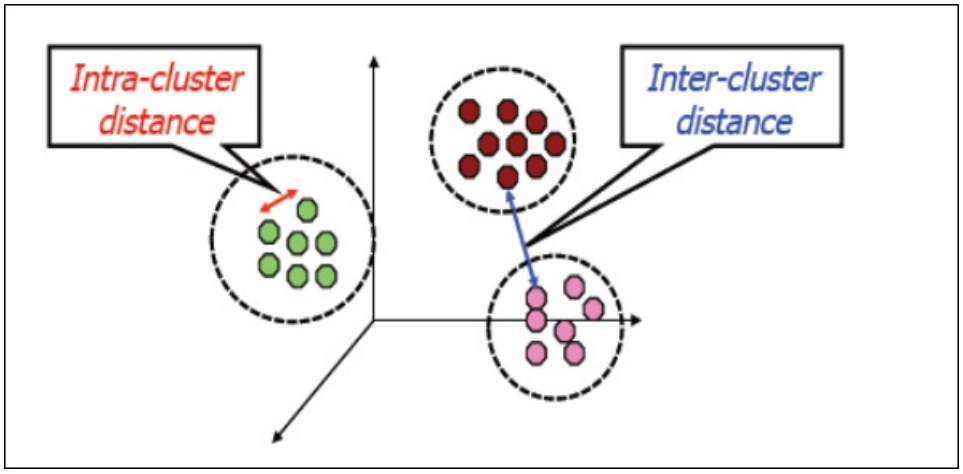
$$I(C) = \frac{\min\limits_{1 \leqslant i < j \leqslant k} \{(\delta(c_i, c_j)\}}{\max\limits_{1 \leqslant l \leqslant k} \{\Delta(c_l)\}}$$

Ou :

$$\delta(c_i, c_j) = \min_{\substack{x \in c_i \\ y \in c_j}} (d(x, y))$$

Et :

$$\Delta(c_l) = \max_{x, y \in c_l} (x, y)$$



L'objectif principal de cet indice est de maximiser la dissimilarité inter-classe et de minimiser la dissimilarité intra-classe. L'objectif est donc de maximiser l'indice.

3.3.3 L'indice basé sur la densité inter et intra classe :

Cet indice est basé comme tous les autres indices [27] sur deux aspects, noter V_{CDbw}. Le premier est la compacité des classes et le deuxième est la séparation des classes. La densité des classes a été ajoutée par cet indice. La densité inter et intra classe est calculée pour évaluer la compacité et la séparabilité des classes. La notation détaillée de cet indice est la suivante :

La déviation standard est donnée :

$$Stdev(i) = \sqrt{\frac{\sum_{k=1}^{n_i}(x_k - m_i)^2}{n_i - 1}}$$

Où n_i est le nombre des éléments associés à la classe i et m_i : la moyenne de la classe i.

La moyenne de la déviation standard :

$$Stdev = \sqrt{\frac{\sum_{i=1}^{K} ||Stdev(i)||^2}{K}}$$

- La densité intra classe est calculer par :

$$intra_{den}(K) = \frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{n_i} density(w_{ij}) \qquad K > 1$$

Avec :

$$density(w_{ij}) = \sum_{l=1}^{n_i} f(x_l, w_{ij})$$

Où : x_l est un élément de la classe i, et w_{ij} : la représentation de la classe i

Tel que :

$$f(x_l, w_{ij}) = \begin{cases} 1 & si : ||x_l - x_{ij}|| \leqslant Stdev \\ 0 & autre\ cas \end{cases}$$

- La densité inter classe est calculer par :

$$inter_{den}(K) = \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \frac{||clos_{rep}(i) - clos_{rep(j)}||}{||Stdev(i)|| + ||Stdev(j)||} \times density(u_{ij}) \quad K > 1$$

Avec :

$$density(u_{ij}) = \sum_{k=1}^{n_i + n_j} f(x_k, u_{ij})$$

Tel que :

$$f(x_l, w_{ij}) = \begin{cases} 1 & si : ||x_k - u_{ij}|| \leqslant \dfrac{||Stdev(i) + Stdev(j)||}{2} \\ 0 & autre\ cas \end{cases}$$

Où $clos_{rep}$: l'objet de la classe i le plus proche à les autres classes. Et u_{ij} est le milieu entre les deux points $clos_{rep}$(i) et $clos_{rep}$(j).

La définition de la valeur de séparation des classes est :

$$sep(K) = \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \frac{||close_{rep}(i) - close_{rep}(j)||}{1 + inter_{den(K)}} \quad K > 1$$

Donc l'indice global est donné :

$$V_{CDbw}(K) = intra_{den}(K) \times sep(K)$$

3.3.4 L'indice de validité basé sur le maximum d'entropie

Dans la section précédente on a revu les principaux indices de validité dédiés à la classification floue non supervisée. Parmi les difficultés par ces indices citons :

- La présence d'une grande variation dans la structure géométrique des clusters. Plus particulièrement pour ce qui est de la forme, la densité et l'orientation.
- Recouvrement entre les clusters.

On trouve des indices basés sur le nombre des classes et où la donnée elle-même est basée sur le principe du maximum d'entropie. L'indice proposé par l'article [21] nommé V_{MEP} est déduit d'une combinaison originale entre des méthodes d'analyse des données et le critère du maximum d'entropie. Ses performances sont montrées à travers un ensemble d'exemples simulés et réels. La procédure est complètement automatique dans le sens qu'elle ne nécessite aucun paramètre de réglage.

Considérons un ensemble de données avec k clusters $C_1, C_2, ..., C_k$ et leurs centres respectifs $g_1, g_2, ..., g_k$. On définit les probabilités P_{ij} comme le lien entre le point i de sa classe C_j (j est obtenu préalablement par l'algorithme de K-means) et son centre g_j. Les points i qui n'appartiennent pas à la classe C_j, ne possèdent aucun lien avec g_j ; c'est à dire $P_{ij} = 0$. Et on a : $\sum_{i \in C_j} P_{ij} = 1$, pour j=1,...,k.

L'indice de validité s'énonce comme suit :

$$V_{MEP} = S = \frac{1}{k} \sum_{j=1}^{k} S_j + ln(k)$$

Où :

S_j représente l'entropie à la classe j.

$$S_j = -\sum_{i \in C_j} P_{ij} ln(P_{ij})$$

Les coefficients P_{ij} est les probabilités que un élément $i \in C_j$, donné par :

$$P_{ij} = \frac{exp[-k\|x_i - g_j\|^2]}{\sum_{i \in C_j} [-k\|x_i - g_j\|^2]}$$

Chapitre 4

Description des méthodes étudiés :

4.1 Problématique :

Plusieurs travaux ont traité ce problème dans la littérature [Davies-Bouldin, Dunn, ELBOW, Silhouette, Écart-statistique,...]. Certaines méthodes de classification automatique sont non paramétriques et ne nécessitent pas un nombre de clusters connu à l'avance. Le nombre de clusters dans ces méthodes est construit itérativement au cours de la phase d'apprentissage de l'algorithme. Le nombre de cluster est induit comme une variable dans la fonction objective à optimiser. D'autres méthodes donnent une estimation du nombre de clusters, par exemple dans la classification spectrale le nombre de clusters est déterminer à partir de noyaux. Le nombre de valeurs propres les plus significatives détermine le nombre de clusters. D'autres méthodes sont basées sur le principe de densité pour déterminer les régions les plus denses qui vont former les différentes classes. De plus, d'autres méthodes basé sur la théorie de l'information, explorent l'espace de toutes les distributions de probabilité possibles des données pour trouver un qui maximise l'entropie soumise à des conditions supplémentaires sur la base de l'information préalable sur des clusters.

Une question fondamentale est : Si les données sont clusterables, alors comment choisir le bon nombre de grappes attendues (k) ?

Plusieurs méthodes ont été proposées pour la détermination du nombre optimal de clusters afin de mesurer les similitudes entre les grappes.

Une solution simple et populaire consiste à inspecter le dendrogramme produit en

utilisant la classification hiérarchique pour voir si elle suggère un nombre particulier de clusters. Malheureusement, cette approche est encore une fois subjective. Dans cette partie, je vais décrire trois méthodes relatives à la détermination du nombre optimal de clusters en utilisant les méthodes de classification k-means, PAM et hiérarchique. Ces méthodes comprennent des méthodes directes et méthodes d'essai statistiques.

Les méthodes directes consistent à optimiser un critère tel que la somme de carrés au sein de la grappe ou la largeur de la moyenne de silhouette. Les méthodes correspondantes sont appelées méthodes de ELBOW et de silhouette, respectivement. Les méthodes d'essai statistique consiste à comparer des preuves contre l'hypothèse nulle, qui est la méthodes d'écart statistique.

Nous nous décrivons ces trois méthodes les plus populaires [30] :

- La méthode de ELBOW,
- La méthode de silhouette,
- La méthode d'écart statistique.

Dans la section 5 (partie 1) on fera une comparaison de ces trois méthodes, en utilisant la base des données d'iris.

4.2 La méthode de ELBOW

4.2.1 Concept :

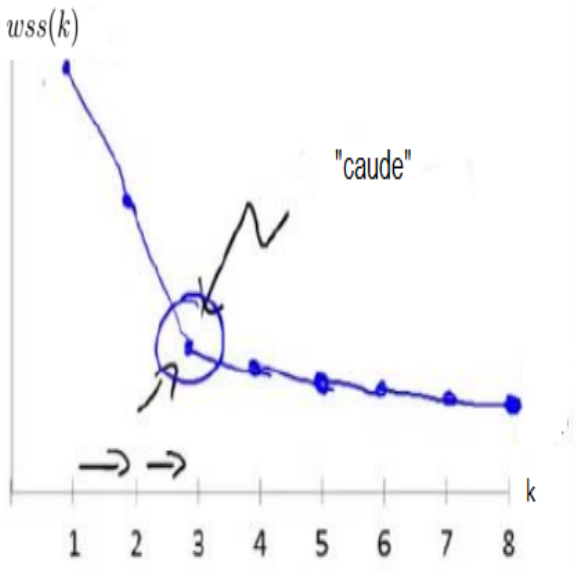
La méthode la plus ancienne pour déterminer le nombre optimal de grappes dans un ensemble de données est appelée la méthode du ELBOW [30]. Cette méthode examine le pourcentage de la variance expliquée en fonction du nombre de grappes : On doit choisir un certain nombre de grappes de sorte que l'ajout d'un autre groupe ne donne pas beaucoup mieux la modélisation des données. Plus précisément, si l'on trace le pourcentage de la variance expliquée par les pôles contre le nombre de grappes, les premières grappes va ajouter beaucoup d'informations (expliquer beaucoup de variance), mais à un certain point le gain marginal va baisser, ce qui donne un angle graphique. Le nombre de grappes est choisi à ce stade, d'où le "critère de coude". Ce "coude" ne peut pas toujours être identifié sans ambiguïté. Le pourcentage de variance expliquée est le rapport de la variance entre les groupes à la variance totale, aussi connu comme un F-test.

La somme moyenne des carrés interne est la distance moyenne entre les points à l'intérieur d'une grappe. Mathématiquement,

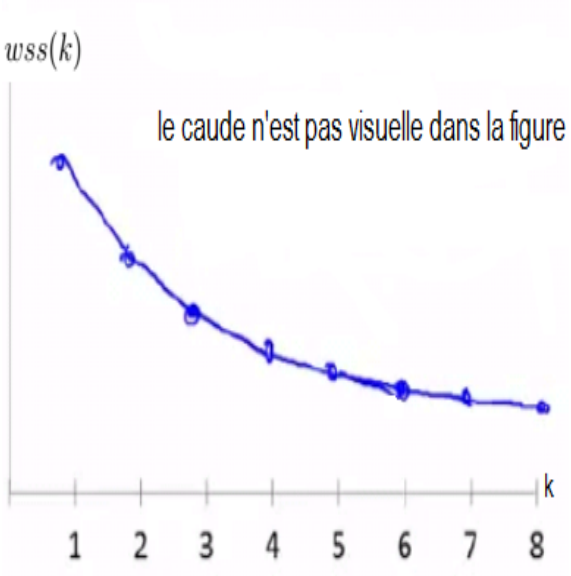
$$wss(k) = \sum_{l=1}^{k} \frac{1}{n_i} D_l$$

$$D_l = \sum_{i=1}^{n_l - 1} \sum_{j=i}^{n_l} ||d_i - d_j||_2$$

Comme montre la figure au-dessous, la distorsion $wss(k)$ descend rapidement avec k croissant de 1 à 2, et de 2 à 3, puis $wss(k)$ atteindre un coude à k = 3, puis la distorsion descend considérablement après. Par conséquence ce coude de cette courbe il ressemble peut-être on prend trois groupes comme un bon nombre de clusters.



Mais le problème avec la méthode du ELBOW est que : Ce "coude" ne peut pas toujours être identifié sans ambiguïté. Parfois, il n'y a pas du coude, ou plusieurs coudes comme le montre la figure suivante :



4.2.2 Algorithme :

Le nombre optimal des classes peut être défini comme suit :

a) Comput l'algorithme de classification (par exemple, k-means) pour différentes valeurs de k. Par exemple, en k variant de 1 à 10 groupements

b) Pour chaque k, calculer la somme des carrés ($wss(k)$) totale au sein du chaque cluster k

c) Tracer la courbe de $wss(k)$ en fonction du nombre de pôles k.

d) L'emplacement d'un coude (genou) dans la parcelle est généralement considéré comme un indicateur du nombre approprié de grappes.

4.3 La méthode de Silhouette

4.3.1 Concept :

Un certain nombre d'approches utilisent des indices comparant les distances intra-cluster avec les distances entre les clusters inter-cluster : la plus différence mieux l'ajustement ; beaucoup d'entre eux sont mentionnés dans Milligan et Cooper [22].

Un coefficient bien équilibré, est la largeur de la silhouette, qui a montré de bonnes performances dans des expériences, a été introduite par Kaufman et Rousseeuw [30]. Le concept de la technique de Silhouette calcule la largeur de la silhouette de chaque

échantillon, la largeur moyenne de la silhouette de chaque classe et de la largeur moyenne de la silhouette pour toutes les données. Cette approche est basée sur la comparaison de sa compacité et de sa séparation. La largeur moyenne de la silhouette est également utilisé pour décider combien de nombre des classes sélectionnées est bon.
Pour construire les silhouettes S(i), on utilise la formule suivante :

$$S(i) = \frac{b_i - a_i}{max(a_i, b_i)}$$

Tel que : a_i est la distance moyenne entre i et toutes les autres entités du groupe auquel appartient i, donné par :

$$a_i = \frac{1}{n(c_i)} \sum_{l \in c_i} dist(i, l)$$

Et b_i est le minimum des distances moyennes entre i et toutes les entités dans l'autre cluster. s'écrit par :

$$b_i = \min_{c_j \in P' \setminus c_i} \sum_{l \in c_j} \frac{dist(i, l)}{n(c_j)}$$

Avec $P' = \{c_1, ..., c_k\}$ est l'ensemble de k partitions, $n(c_i)$ est le cardinal de cluster c_i et la distance (euclidienne, Manhattan,...) entre les observations i et l qui est représentée par dist(i,l).

Largeur de silhouette : On appelle largeur de silhouette de la partition réel :

$$S = \frac{1}{n} \sum_{i=1}^{n} S(i).$$

On a alors l'interprétation suivante :

Valeur de S	Nature de la structure
$\in]0.51, 1]$	Forte
$\in]0.31, 0.50]$	Raisonnable
$\in [0, 0.30[$	Faible
$\in [-1, 0[$	Inexistante

Les valeurs de la largeur de la silhouette se situent dans la plage allant de -1 à 1. Si la valeur de la largeur de la silhouette d'une entité est à peu près nulle, cela signifie que

l'entité ne pouvait être attribuée à un autre cluster aussi bien. Si la valeur de la largeur de la silhouette est proche de -1, cela signifie que l'entité est mal classée. Si toutes les valeurs de largeur de silhouette sont proches de 1, cela signifie que l'ensemble est bien regroupé.

Un cluster peut être caractérisée par la largeur moyenne de la silhouette d'entités individuelles. La plus grande largeur moyenne de silhouette, sur différents k indique le meilleur nombre de clusters.

4.3.2 Algorithme :

L'algorithme est similaire à la méthode de ELBOW et peut être calculée comme suit :

a) Comput algorithme de classification (par exemple, k-means) pour différentes valeurs de k. Par exemple, en k variant de 1 à 10 groupements,

b) Pour chaque k, calculer la largeur silhouette moyenne des observations,

c) Tracer la courbe de la largeur silhouette moyenne des observations en fonction du nombre de pôles k,

d) L'emplacement du maximum est considéré comme le nombre approprié de grappes.

4.4 La méthode d'écart statistique :

4.4.1 Concept :

Dans les méthodes de clustering le nombre de grappes est soit un paramètre direct, ou peut être commandé par d'autres paramètres du procédé. Estimation du bon nombre de grappes est un problème important dans le choix de la méthode de clustering ainsi que lors de la validation du résultat. La méthode d'écart statistique [13] est l'une des techniques les plus populaires pour déterminer le nombre optimal de clusters. L'idée de cette technique est qu'elle compare le total de la variation au sein d'intra-cluster pour différentes valeurs de k avec leurs valeurs attendues dans la distribution de référence nulle des données, à savoir une distribution sans regroupement évident. L'approche peut être appliquée à toute méthode de classification (K-means, PAM et hiérarchique,...). La variation d'intra-classe totale pour un cluster k est donné par la somme de carrés totale ($wss(k)$).

L'ensemble de données de référence est généré à l'aide de simulations de Monte Carlo le processus d'échantillonnage (la méthode des simulations de Monte-Carlo, désigne une

famille de méthodes algorithmiques visant à calculer une valeur numérique approchée en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes.). Autrement dit, pour chaque variable (x_i) dans l'ensemble de données, nous calculons sa gamme [min(x_i), max(x_j)] et générer des valeurs pour les n points uniformément depuis cet intervalle.

Pour les données observées et des données de référence, la variation d'intra-cluster totale est calculée en utilisant différentes valeurs de k. La statistique d'écart pour un k donné est défini comme suit :

$$Gap_n(k) = E_n^*\{log(wss_k)\} - log(wss_k)$$

Où E_n^* désigne l'espérance sous un échantillon de taille n de la distribution de référence. E_n^* est définie en générant B copies des ensembles de données de référence et en calculant la moyenne $log(wss_k^*)$. La statistique de l'écart mesure l'écart de la valeur wss_k observée à partir de sa valeur attendue sous l'hypothèse nulle.

L'estimation des grappes optimales $\widehat{k}$ sera la valeur qui maximise $Gap_n(k)$ (à savoir, qui donne le plus grand écart statistique). Cela signifie que la structure de regroupement est éloignée de la répartition uniforme de points.

L'écart-type (sd_k) de $log(w_k^*)$ est également calculée afin de définir l'erreur standard (s_k) de la simulation comme suit :

$$s_k = sd_k \times \sqrt{1 + \frac{1}{B}}$$

Enfin, une approche plus robuste est de choisir le nombre optimal de groupes k comme le plus petit nombre tel que :

$$Gap(k) \geqslant Gap(k+1) - s_{k+1}$$

Autrement dit, nous choisissons la plus petite valeur de k telle que la statistique de l'écart se situe dans un écart-type de l'écart à k + 1.

4.4.2 Algorithme :

L'algorithme comprend les étapes suivantes :

a) Regroupez les données observées, variant le nombre de grappes de $k = 1, ..., k_{max}$, et calculer la wss_k correspondante.

b) Générer des ensembles de données de référence B et grouper chacun d'eux avec un nombre de pôles variable $k = 1, ..., k_{max}$. Calculer l'écart statistique estimé :

$$Gap(k) = \frac{1}{B} \sum_{b=1}^{B} log(wss^*_{kb}) - log(wss_{kb})$$

c) Soit $\overline{w} = \frac{1}{B} \sum_b log(wss^*_{kb})$, calculer la déviation standard :
$sd(k) = \sqrt{\frac{1}{B} \sum_b (log(wss^*_{kb} - \overline{wss})^2}$ et définie : $s_k = sd_k * \sqrt{1 + \frac{1}{B}}$

d) Choisir le nombre des clusters est la plus petit valeur k tel que :

$$Gap(k) \geqslant Gap(k+1) - s_{k+1}$$

Chapitre 5

Comparaison et Application :

5.1 Partie 1 : Comparaison entre les trois méthodes étudiés

Introduction :

Dans cette première partie, on va essayer de comparer entre les trois méthodes ont décrit dans le chapitre précèdent :

- ELBOW
- Largeur de silhouette
- L'écart statistique

En combinant ces trois méthodes avec les algorithmes de la classification : K-means, PAM et la classification hiérarchique. Afin de sélectionner une meilleur entre eux, pour l'appliquer dans la seconde partie de ce chapitre à un exemple d'application de clustering qui est la segmentation d'image médicale.

Pour cela, on utilise le jeu de données d'Iris qui est présenté dans l'Annexe A.

Tous ses résultats sont manipulés et donnés par le logiciel R. Une petite historique sur ce logiciel est présenté dans l'Annexe B.

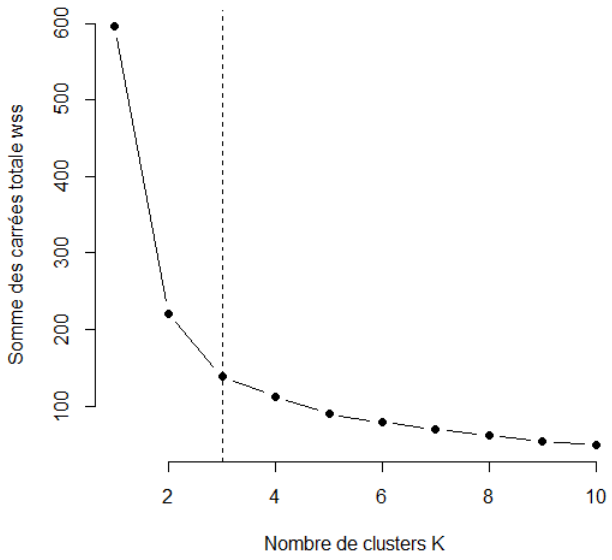
5.1.1 Résultats de la méthode de ELBOW :

a) K-means

On travaille sur les données d'iris décrit au début de ce chapitre, en appliquant la méthode de la classification k-means décrite en avant afin de déterminer le nombre optimal de clusters, en exécutant les instructions suivantes :

```
library(factoextra)
library(cluster)
#### Les données
data(iris)
#### Retirer la colonne des espèces (5) et l'échelle des données
iris.scaled <- scale(iris[, -5])
data <- iris.scaled
set.seed(123)
####################\\\ 1- La méthode de ELBOW ////#############
########%%%%% 1-1 La méthode de Elbow en utilisant k-means: %%%%%
## Calculer et tracer wss pour k = 2 à k = 10
k.max <- 10 # Nombre Maximal des clusters
wss <- sapply(1:k.max,
  function(k){kmeans(data, k, nstart=10 )$tot.withinss})
plot(1:k.max, wss,type="b", pch = 19, frame = FALSE,
     xlab="Nombre de clusters K",
     ylab="Somme des carrées totale wss")
abline(v = 3, lty =2)
```

on aura le résultat suivant :



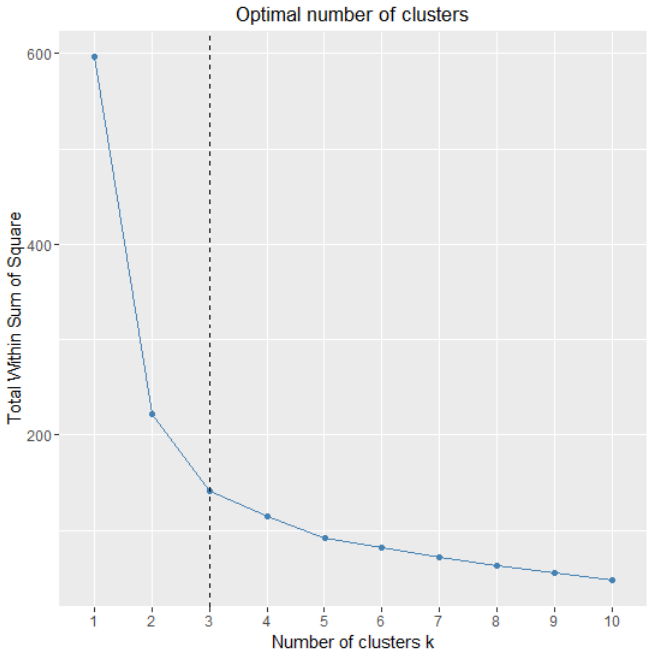
D'après la figure au-dessus, on observe que le nombre optimal de clusters est $k = 3$.

b) PAM

On travaille toujours sur les données d'iris, en appliquant cette fois-ci la méthode de la classification PAM décrite dans le premier chapitre. La détermination de nombre optimal de clusters se fait en exécutant les instructions suivantes :

```
############ 1-2 La méthode de Elbow en utilisant PAM: %%%%%%%%%%
fviz_nbclust(data, pam, method = "wss",k.max = 10) +
   geom_vline(xintercept = 3, linetype = 2)
```

On trouvera le résultat suivant :



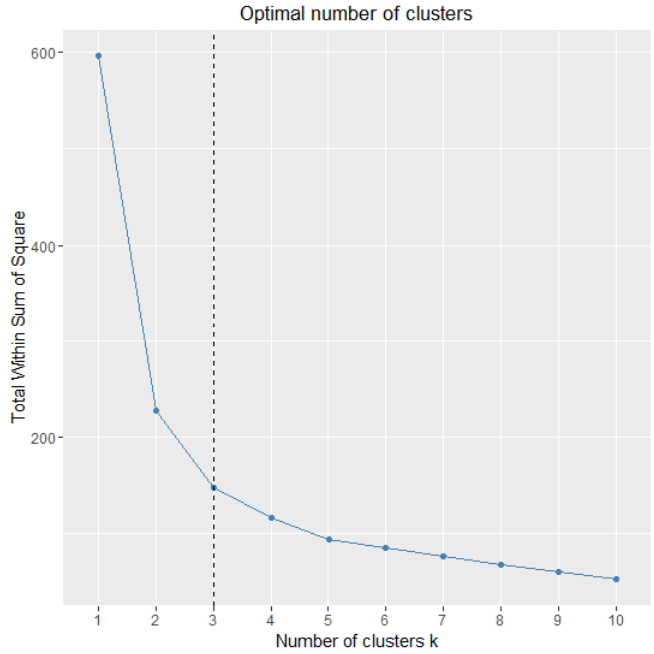
D'après la figure au-dessus, on observe bien que le nombre optimal de clusters est $k = 3$.

c) La classification hiérarchique

On travaille toujours sur les données d'iris, en appliquant cette fois-ci la méthode de la classification hiérarchique décrite dans le premier chapitre, en exécutant les instructions suivantes :

```
########### 1-3 La méthode de Elbow en utilisant hierarchical: %%%
fviz_nbclust(data, hcut, method = "wss",k.max = 10) +
    geom_vline(xintercept = 3, linetype = 2)
```

Après l'exécution de cette méthode on a le résultat suivant :



D'après la figure au-dessus, on déduit que le nombre optimal de cluster est toujours $k = 3$.

Discussion :

Cette méthode fut longtemps utilisé sur les grands jeu de données, en raison de sa rapidité qui sera démontrer par la suite quand on va l'appliquer en segmentation d'image (exemple d'application qui demande beaucoup de calculs). Le meilleur choix se fait quand on combinera cette méthode avec l'algorithme K-Means par contre au méthodes de classification hiérarchique et PAM qui sont limités lorsqu'on travail avec ce jeu de données, ce choix vient de sa simplicité de mise en ouvre, et que cet algorithme c'est le seul qui marche bien avec des données de grands tailles au point de vue des calculs.

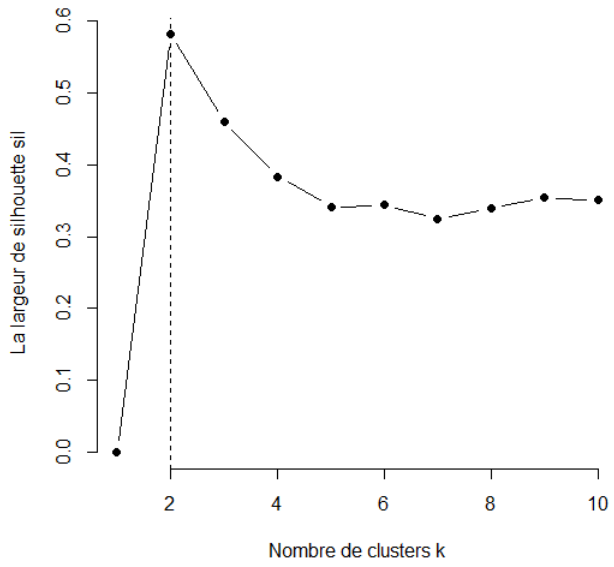
5.1.2 Résultats de la méthode de Silhouette :

a) K-means

Maintenant on va essayer d'appliquer la méthode de Silhouette en utilisent la méthode de classification k-means, en tournent les instructions suivantes :

```
#############\\\\\ 2- La méthode de Silhouette //////##############
# les memes données que la méthode d'ELBOW
########%%%%% 2-1 La méthode de silhouette en utilisant k-means %%
k.max <- 10
sil <- rep(0, k.max)
## Calculer la largeur moyenne de la silhouette pour k = 2 à k = 10
for(i in 2:k.max){km.res <- kmeans(data, centers = i, nstart = 25)
ss <- silhouette(km.res$cluster, dist(data))
sil[i] <- mean(ss[, 3])
}
#### Tracer la largeur moyenne de la silhouette
#### en fonction nombre de clusters
plot(1:k.max, sil, type = "b", pch = 19,
   frame = FALSE, xlab = "Nombre de clusters k",
     ylab = "La largeur de silhouette sil")
abline(v = which.max(sil), lty = 2)
```

On aura le résultat suivant :



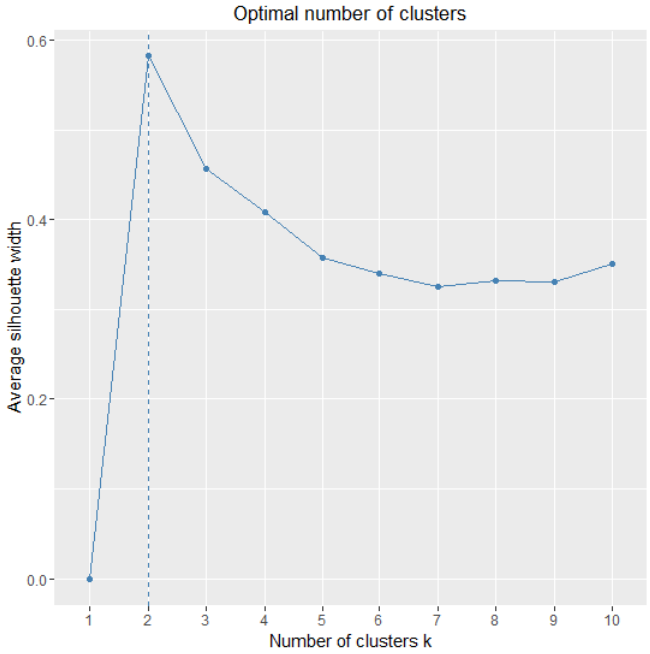
Le nombre de clusters que suggère cette méthode est donné par $k = 2$, qui est différent complètement à ce qu'on a trouvé dans la méthode de ELBOW. Par conséquence, cette méthode est peut-être considérer moins robuste que la méthode de ELBOW.

b) PAM

En utilisant la méthode de classification PAM :

```
###%%% 2-2 la largeur moyenne de la silhouette en utilisant PAM %%%##
require(cluster)
fviz_nbclust(data, pam, method = "silhouette",k.max = 10)
```

On aura le résultat suivant :



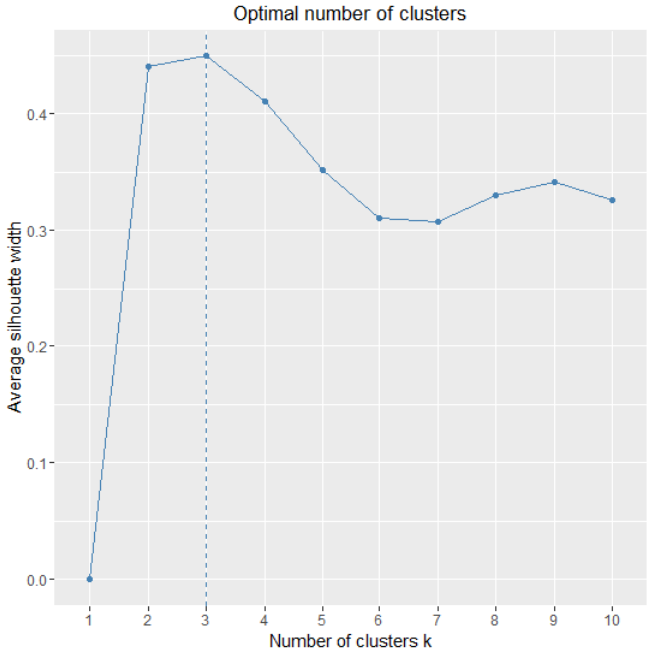
Même résultat que de la méthode k-meams $k = 2$.

c) La classification hiérarchique

On travaille toujours avec les données d'iris, en appliquant cette fois-ci la méthode de la classification hiérarchique, la détermination du nombre optimal de clusters se fait par l'exécution des instructions suivantes :

```
##%% 2-3 la largeur moyenne de la silhouette en utilisant hiérarchique
require(cluster)
fviz_nbclust(data, hcut, method = "silhouette",hc_method = "complete",k.max = 10)
```

On aura le résultat suivant :



Le vrai nombre optimal de cluster est donné par cette méthode $k = 3$. On déduit que la méthode de silhouette quand on a la combiné avec les deux méthodes : k-means et PAM on a trouvé que le nombre de clusters est $k = 2$, mais quand on a la combiné avec la méthode de la classification hiérarchique on a trouvé $k = 3$. Ces différents résultats nous conduisons à faire une décision sur ce nombre.

Pour confirmer le quel de ses résultats donne le nombre optimal de clusters, nous essayons de tourner par suite la méthode de la statistique d'écart, afin de déterminer le nombre optimal de clusters.

Discussion :

A travers les résultats obtenus sur cette méthode on constate que :

- **Cette méthode échoue complètement si on a la combiner avec les deux meilleurs algorithmes K-Means et PAM d'estimer le nombre exact de clusters.**
- **La performance de cette méthode diminue, il est important de noter qu'il y a une différence significative des résultats obtenus par rapport à la méthode de ELBOW.**

5.1.3 Résultats de la méthode de la statistique d'écart :

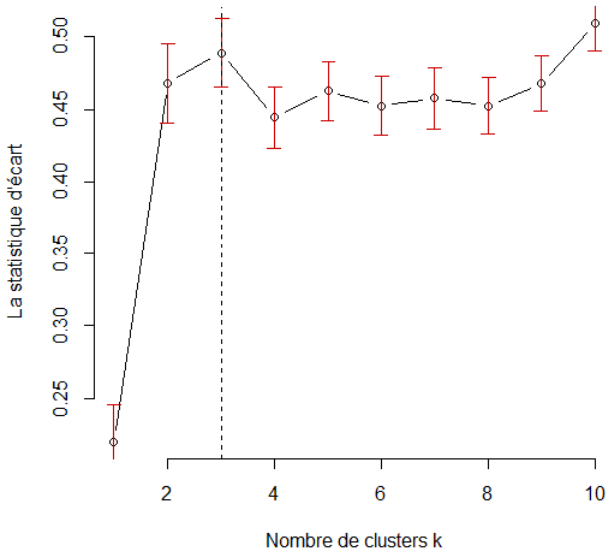
D'autre part, maintenant on va essayer d'utiliser une autre méthode qui est basée sur la théorie statistique, cette méthode est nommée la statistique d'écart qui est décrite dans le chapitre précèdent. En combinent cette méthode avec : k-means, PAM et la classification hiérarchique. Afin de chercher le nombre optimal de clusters convenable.

a) K-means :

En commençant avec k-means :

```
##############\\\ 3- La méthode de l'écart statistique ////##############
# les memes données que la méthode d'ELBOW
#######%%%% 3-1 La méthode de l'écart statistique en utilisant k-means %%%
# Calculer la statistique d'écart
# Nous utilisons B=50
gap_stat <- clusGap(data, FUN = kmeans, nstart = 25,K.max = 10, B = 50)
# afficher les résultats
print(gap_stat, method = "firstmax")
# Tracer de la statistique d'écart
plot(gap_stat, frame = FALSE, xlab = "Nombre de clusters k",
                  ylab= "La statistique d'écart")
abline(v = 3, lty = 2)
```

On aura le résultat suivant :



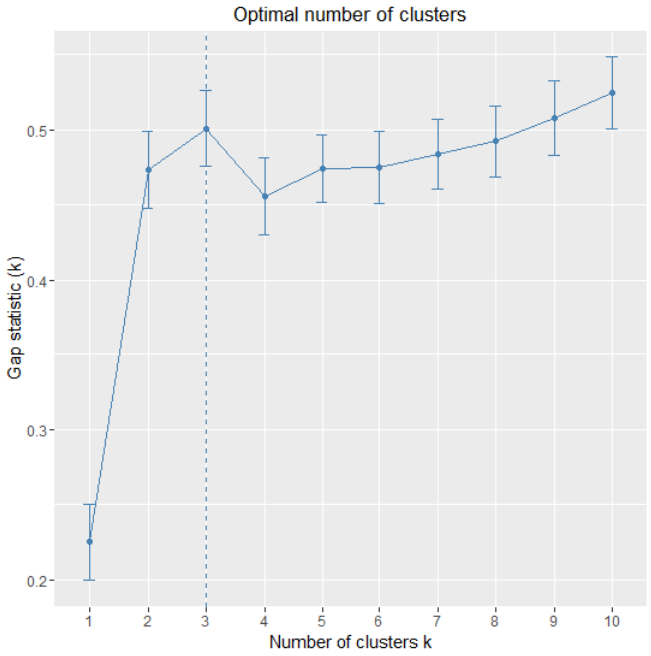
D'après le graphe au-dessus le nombre optimal de clusters est $k = 3$.

b) PAM :

En utilisant la méthode de classification PAM :

```
########## 3-2 La méthode de l'écart statistique en utilisant PAM %%%###
# Calculer de la statistique d'écart
gap_stat <- clusGap(data, FUN = pam, K.max = 10, B = 50)
# Tracer de la statistique d'écart
fviz_gap_stat(gap_stat)
```

On trouvera :



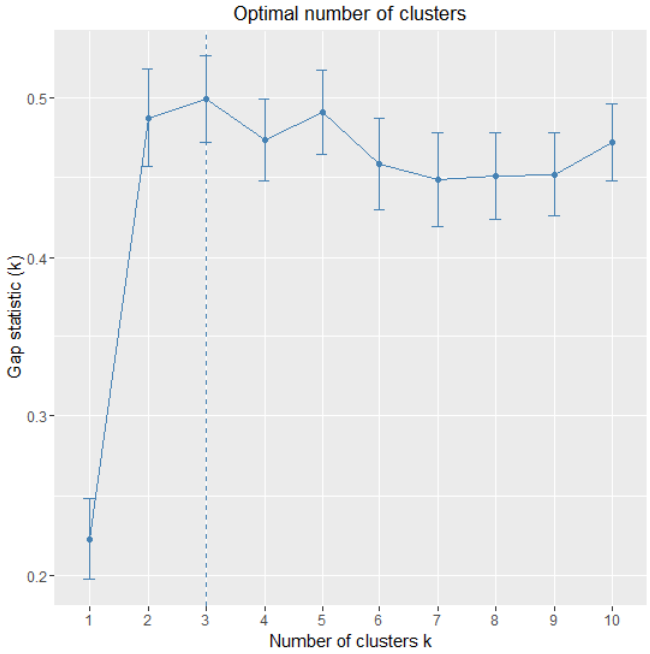
Même résultats, le nombre optimal de clusters est toujours : $k = 3$.

c) La classification hiérarchique :

Finalement, on va faire appel à la l'approche de la classification hiérarchique :

```
##% 3-3 la largeur moyenne de la silhouette en utilisant hiérarchique ###
# Calculer de la statistique d'écart
gap_stat <- clusGap(data, FUN = hcut, K.max = 10, B = 50)
# Tracer la statistique d'écart
fviz_gap_stat(gap_stat)
```

Le résultat est donné par la figure suivante :



Toujours le nombre optimal de clusters est $k = 3$.

Discussion :

A travers les résultats obtenus sur cette méthode on constate que :

- **Cette méthode arrive à estimer le nombre exact de clusters, en combinaison avec les trois algorithmes de clustering, ses résultats sont nettement meilleurs que ceux obtenus par la largeur de silhouette.**
- **Mais le comportement de cette méthode est très particulier si on travail avec des jeu données de grand taille, cette méthode prend beaucoup de temps de calculs, cela nous conduisons à ne pas la choisir par la suite.**

5.1.4 Conclusions entre les méthodes :

Pour faire une comparaison entre les méthodes : ELBOW, la largeur moyenne de silhouette et la statistique d'écart on a les combinés avec trois algorithmes les plus utilisés dans la classification non supervisée : k-means, PAM et classification hiérarchique, on a bien trouvé que :

- Trois solutions de clusters sont proposées en utilisant ces trois algorithmes en combinaison avec la méthode du ELBOW, tous ces trois solutions donnent le même

nombre de clusters $k = 3$ qui sera le nombre optimal de clusters.

- La méthode de la largeur moyenne de silhouette donne deux solutions de cluster $k = 2$ en utilisant des algorithmes k-means et PAM. Mais la combinaison de la classification hiérarchique et la méthode de la largeur moyenne de silhouette retourne au neveau 3 clusters $k = 3$. Par conséquence cette méthode ne donne pas le nombre éxacte de clusters quand cherche.
- Même solutions ont trouvé avec la méthode de ELBOW, ils ont aussi trouvé par la méthode de la statistique d'écart.

Remarque :

A travers les résultats obtenus sur ces trois méthodes nous constatons que les deux méthodes : ELBOW et la statistique d'écart estiment le vrai nombre optimal de clusters par contre à la méthode de silhouette. Mais à la pratique de ces deux méthodes spécifiquement quand on travail avec des données de grande taille qui demandent beaucoup de temps de calcul, on trouvera que la seule méthode qui marche bien et donne les meilleurs résultats est la méthode de ELBOW.

Par suite de cette étude, on s'intéresse seulement à cette méthode ELBOW que sera appliquer à un exemple d'application de la classification non supervisée, dans la deuxième partie de ce chapitre. Mais d'abord, on va juger notre choix de cette méthode par un test de 30 indices de validité existent dans la littérature.

5.1.5 Déterminer le meilleur nombre de clusters :

Comme mentionné dans l'introduction de chapitre précédent, de nombreux indices ont été proposés dans la littérature pour déterminer le nombre optimal de groupes dans un partitionnement d'un ensemble de données au cours du processus de regroupement. Un article publié par Charrad et al. [18], fournit 30 indices pour déterminer le nombre approprié de grappes et propose aux utilisateurs le meilleur schéma de regroupement des différents résultats obtenus en faisant varier les combinaisons de nombre de grappes, des mesures de distance et les méthodes de clustering. Un avantage important d'utiliser le logiciel R, est que l'utilisateur peut calcule simultanément plusieurs indices et déterminer le nombre de grappes dans un appel à des fonctions prés défini.

Calculer de ces 30 indices au jeu de données d'iris :

Nous fournirons des codes R pour calculer tous ces 30 indices afin de déterminer le meilleur nombre de clusters à l'aide de la "règle de la majorité".

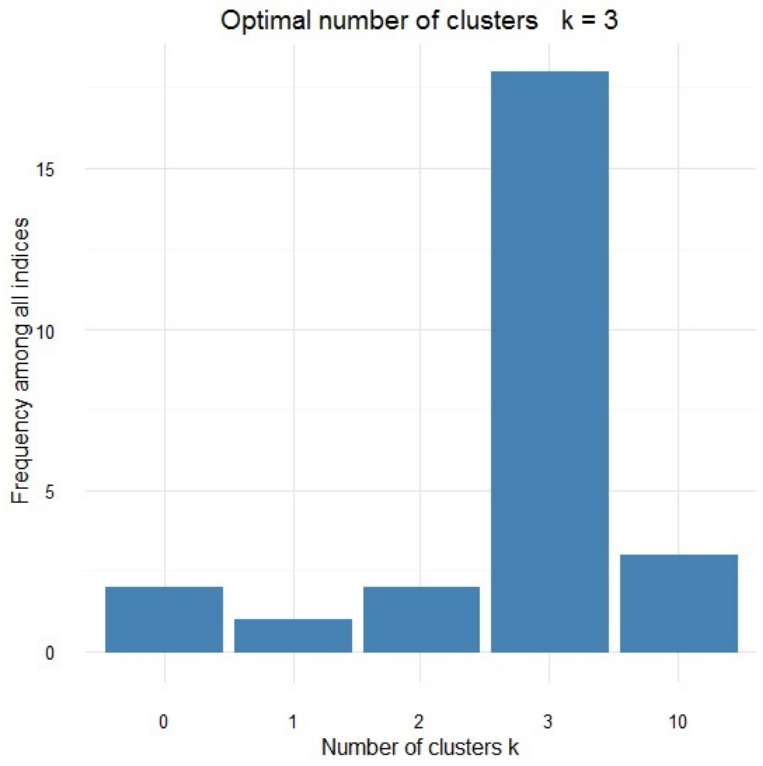
On exécute les instruction suivantes :

```
######%%% détermination de nombre de cluster pour 30 indices de validité
# les memes données que la méthode d'ELBOW en ajoutent la bibiolique
library("NbClust")
nb <- NbClust(data, distance = "euclidean", min.nc = 2,
      max.nc = 10, method = "complete", index ="all")
# afficher les résultats
nb
# Parmi tous les indices:
fviz_nbclust(nb) + theme_minimal()
```

On aura les résultats suivants :

```
Among all indices:
===================
* 2 proposed  0 as the best number of clusters
* 1 proposed  1 as the best number of clusters
* 2 proposed  2 as the best number of clusters
* 18 proposed  3 as the best number of clusters
* 3 proposed  10 as the best number of clusters

Conclusion
==========================
* According to the majority rule, the best number of clusters is  3 .
```



Discussion :

Les résultats présentés dans l'exemple précédent montrent que le choix du nombre optimal de classes varie d'un indice à un autre. En effet, 18 parmi les 30 indices proposent 3 comme le bon nombre de classes, 3 indices proposent 10, 2 indices proposent 2 et un seul indice propose 1 comme le nombre optimal de classes. Par conséquent, l'utilisateur se trouve face au dilemme de choix entre les différentes valeurs proposées par les indices. Pour pallier ce problème, nous proposons d'utiliser la règle du vote majoritaire. Dans l'exemple ci-dessus, 18 indices parmi les 30 indices proposent 3 comme le nombre optimal de classes, ce qui est réellement le bon nombre de classes dans le jeu de données simulées.

Ces résultats montrent bien que notre choix de la méthode de ELBOW est convenable.

5.2 Partie 2 : Application de la méthode de ELBOW à la segmentation d'image

La segmentation est une étape indispensable dans de nombreuses chaînes de traitement de plusieurs domaines fondamentaux de la recherche clinique. La qualité de l'interprétation d'une image dépend fortement de celle de la segmentation qui est une étape de base du traitement d'une image. Parmi les images qu'on a utilisées souvent dans la classification automatique on trouve les images médicales. L'information apportée par l'imagerie médicale est d'un apport considérable en matière de diagnostic.

Dans ce paragraphe nous voulons appliquer les méthodes étudiées dans les chapitres précédents. Nous avons choisi de débuter par l'algorithme k-means car c'est le classificateur non supervisé le plus simple et le plus utilisé. Cependant avant d'appliquer ces méthodes nous présentons une brève description des images médicales utilisées.

5.2.1 L'imagerie médicale :

L'imagerie médicale macroscopique regroupe un ensemble de techniques reposant sur l'utilisation d'un phénomène physique et permettant de visualiser une partie du corps humain ou d'un organe et d'en conserver une image, dans l'objectif de réaliser

un diagnostic, de guider un geste thérapeutique ou de suivre à moyen terme les résultats d'un traitement. On peut classer les modalités d'imagerie médicale par agent physique, par type d'images réalisées, par nuisance ou par leur utilisation médicale. Les agents physiques peuvent être des rayonnements de photons, des champs magnétiques ou des ondes ultrasonores. Les images peuvent être des images de projection planes, des images de coupe appelées images tomographiques, et des séquences temporelles de ces types d'images.

Pour obtenir une bonne classification il faut connaître le nombre exact des classes. Nous avons fait appel à la méthode de ELBOW décrite en avant (on a montré dans la première partie sa performance et sa robuste) afin de déterminer le nombre optimale de classes. Sur certaines images il est facile de trouver ce nombre. Cependant, souvent il est très difficile de sélectionner le nombre exact des classes c'est le cas des images médicales ou les images qui présentent un degré de chevauchement élevé, comme le montre l'exemple suivant :

5.2.2 Exemple de compression d'image :

Considérons les images de cerveau ci-dessous, que pensez-vous combien de couleurs sont là dans ces images ? D'une autre manière, quel est le nombre optimal de clusters qu'on peut tracer de nouveau ces images ?

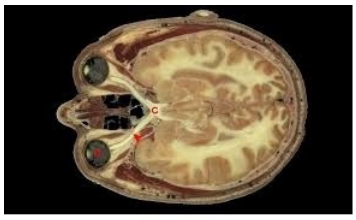


image 1

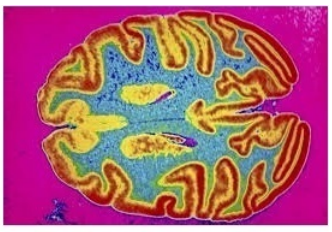


image 2

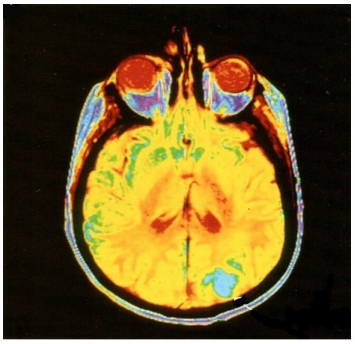


image 3

Ces images contient plus de 124 combinaisons de couleurs uniques pour chaque image, notre objectif est de reproduire l'image similaire en utilisant très peu de couleurs (peut-être 3 ou 4). On va travailler seulement avec l'une de ces images, en choisissent par exemple l'image 3, en considérant cette image comme un ensemble de trois variables (R, G, B) à chaque pixel. Mettons-nous notre image dans un format de trame de données, à savoir que nous allons essayer d'obtenir, la valeur "Red" "Green" "Blue" pour chacun des pixels :

```
library(cluster)
library("jpeg")
library(ggplot2)
# télécharger l'image
cerv_img<-readJPEG("C:/Users/UmArr/Desktop/tpp/application_image/image3.jpg")
# Obtenire les dimensions i.e (Pixels, Valeurs des couleurs)
cerv_Dm <- dim(cerv_img)
# Mettons-nous notre image dans un format de trame de données
cerv_RGB <- data.frame(
x_axis = rep(1:cerv_Dm[2], each = cerv_Dm[1]),
y_axis = rep(cerv_Dm[1]:1, cerv_Dm[2]),
Red = as.vector(cerv_img[,,1]),
Green = as.vector(cerv_img[,,2]),
Blue = as.vector(cerv_img[,,3])
)
```

Maintenant, l'image 3 n'est rien qu'une collection de trois variables à chaque pixel.

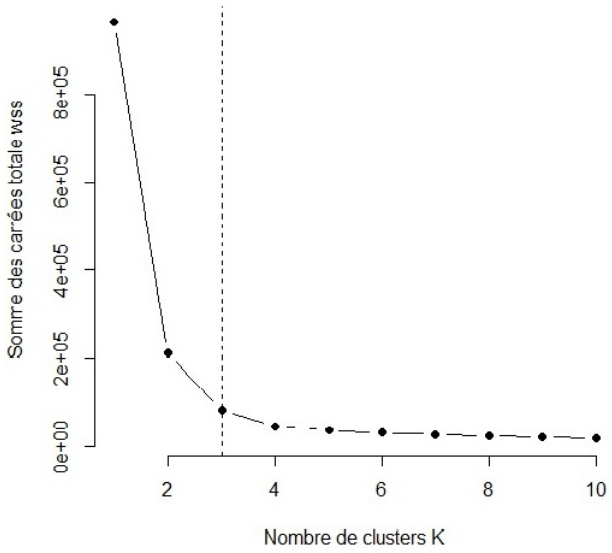
```
> head(cerv_RGB,10)
   x_axis y_axis        Red      Green       Blue
1       1    556 0.03137255 0.03529412 0.01568627
2       1    555 0.03137255 0.03529412 0.01568627
3       1    554 0.03137255 0.03137255 0.02352941
4       1    553 0.03137255 0.03137255 0.03921569
5       1    552 0.03137255 0.03137255 0.03921569
6       1    551 0.03137255 0.03137255 0.03921569
7       1    550 0.02745098 0.03529412 0.03137255
8       1    549 0.01960784 0.03921569 0.01568627
9       1    548 0.01960784 0.03921569 0.01568627
10      1    547 0.01960784 0.03921569 0.01568627
>
```

Appliquons K-Means pour tracer à nouveau cette image :

Nous allons commencer l'algorithme K-Means en utilisant la méthode de ELBOW afin de déterminer le nombre optimale de classes dans cette image. A partir de la trace de cette méthode ci-dessous.

```
# comencer le clustering
############ La méthode de ELBOW ###################
data <- scale(cerv_RGB[c(3,4,5)])
  set.seed(123)
## Calculer et tracer wss pour k = 2 à k = 10
k.max <- 10 # Nombre Maximal des clusters
  wss <- sapply(1:k.max,
  function(k){kmeans(data, k, nstart=10 )$tot.withinss})
plot(1:k.max, wss,type="b", pch = 19, frame = FALSE,
       xlab="Nombre de clusters K",
     ylab="Somme des carrées totale wss")
abline(v = 3, lty =2)
```

Nous pouvons voir que la prise de nombre de classes $k = 3$ devrait être le nombre optimal des classes.



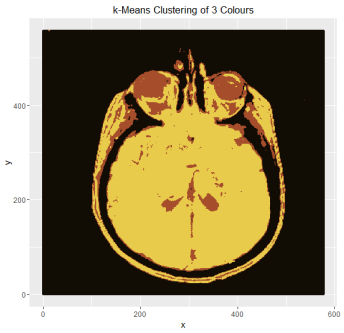
Maintenant, on va essayer de tourner l'algorithme K-Means avec le nombre optimal de grappes déterminer par la méthode de ELBOW $k = 3$:

```
# tourner l'algorithme k-means
k_cluster <- 3
k_cerv_clstr <- kmeans(cerv_RGB[, c("Red", "Green", "Blue")],
                       centers = k_cluster)
k_cerv_colors <- rgb(k_cerv_clstr$centers[k_cerv_clstr$cluster,])
```

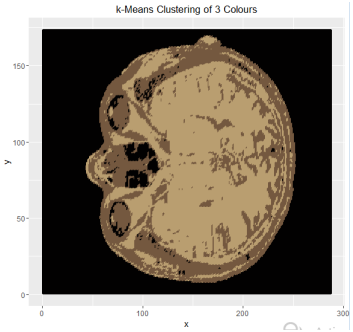
Finalement, une fois que nous obtenons les clusters, l'étape suivante consiste à tracer à nouveau l'image avec seulement trois couleurs, qui ne sont que des centres de clusters.

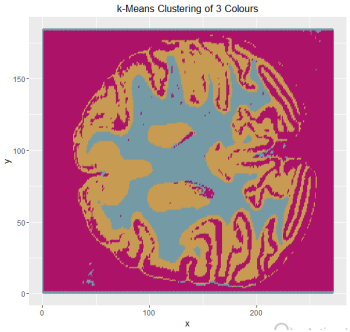
```
# tracer l'image compresser
ggplot(data = cerv_RGB, aes(x = x_axis, y = y_axis)) +
geom_point(colour = k_cerv_colors) +
labs(title = paste("k-Means Clustering of", k_cluster, "Colours")) +
xlab("x") +
ylab("y")
```

Nous pouvons voir que K-Means nous a permis de recréer l'image en utilisant seulement 3 couleurs.



Et pour les images : image 1 et l'image 2 on a utilisé seulement 3 clusters (couleurs), pour retracer les images comme le montre les figures suivantes :





<u>Discussion :</u>

La segmentation d'image constitue une étape essentielle en traitement d'image (en particulier l'image médicale). De nombreuses algorithmes ont été proposées dans la littérature, l'algorithme k-means est l'algorithme de clustering le plus connu et le plus utilisé, du fait de la simplicité de mise en oeuvre. Il partitionne les données d'une image médicale en K clusters. Il est utile de noter que l'algorithme k-means est très performant en termes de temps d'exécution. De fait, il permet de segmenter une image médicale en un nombre K (entier) de classes homogènes, mais il souffre de problème de nombre de clusters K qu'on doit le précisé en avant. Face à ce problème on a fait appeler à la méthode de ELBOW afin de déterminer ce nombre optimal de clusters, Cet indice de validité permet d'identifier le nombre exact de classes qui donne une amélioration de la classification non supervisée. Les résultats présentés dans l'exemple précédent montrent la performance de la méthode de ELBOW.

Conclusion

Dans ce travail, nous nous sommes intéressés essentiellement à la classification floue non supervisée.

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

Dans ce contexte Les travaux de recherche présentés dans ce mémoire concernent les indices de validité et la détermination de nombre optimal de clusters.

En introduction de ce mémoire certaines objectifs ont été présentés. Il convient ici de préciser ceux qui ont été atteints.

Le premier objectif consiste à la mise en place du principe fondamental de clustering ainsi que son processus. de plus, une description générale sur des différents types de clustering qui sont considérés comme une plate forme de notre étude, qui est la détermination du nombre optimal de clustering.

Le deuxième objectif consistait à utiliser quelques techniques de regroupement traditionnelles (clustering hiérarchique, k-means et PAM) dans le contexte de déterminer le nombre optimal de clusters et d'avoir une idée sur la segmentation d'image en imagerie médicale.

Le troisième objectif consistait à décrire les indices de validité qui sont utilisés pour indiquer le meilleur choix possible du nombre de clusters. Un indice de validité est une fonction qui fournit une mesure formelle du résultat d'un algorithme de clustering. Sa valeur est ainsi ces indices de validité permettent d'identifier le nombre exact de classes qui donne une amélioration de la classification non supervisée.

Le quatrième objectif concerne la comparaison entre trois méthodes de clustering (ELBOW, la largeur de silhouette et la statistique d'écart) traitent le problème de la détermination de clusters comme étant notre objectif essentielle de cette étude, afin de sélectionner la meilleure entre eux. A la fin, on a appliqué cette méthode en combinaison avec l'algorithme de la classification K-means, dans un exemple en imagerie médicale comme étant l'un des applications de clustering.

Annexe A : Descriptif de jeu de donnée

Cette annexe présente la principal jeu de donnée sur laquel ont été testés les algorithmes étudiés ELBOW, largeur de silhouette et écart statistique présentés dans le chapitre 4 (partie 1). Cette jeu est choisie parmi les jeux de données disponible dans l'UCI Machine Learning Repository pour démontrer l'adaptabilité des algorithmes. Cette jeu de donnée est présentée par :

Les données d'Iris :

C'est une jeu de données sur la plante Iris dont la source sont les travaux de R.A. Fisher "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936). Ces données sont souvent utilisées en classification. Il y a 3 classes d'Iris à découvrir : Iris Setosa, Iris Versicolor et Iris Virginica. Le jeu de données contient 150 instances réparties à égalité dans chaque classe (50 par classe). Il y a quatre attributs numériques :

1. sepal length (longueur du sépale) en cm,
2. sepal width (largeur du sépale) en cm,
3. petal length (longueur du pétale) en cm
4. petal width (largeur du pétale) en cm.

Le tableau A.1 donne des indications sur les données. La figure A.1 montre la distribution du jeu de données par l'intermédiaire des projections des points de données selon tous les 16 paires de dimensions.

	Min	Max	Mean	Class correlation
sepal length	4.3	7.9	5.84	0.7826
sepal width	2.0	4.4	3.05	-0.4194
petal length	1.0	6.9	3.76	0.9490
petal width	0.1	2.5	1.20	0.9565

Tableau A.1 – Statistiques descriptives du jeu de données Iris

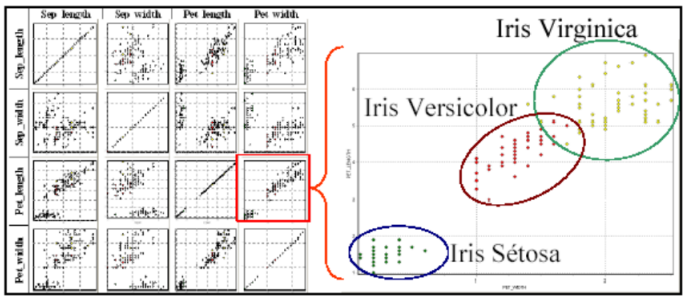


Figure A.1. Distribution du jeu de données iris

Annexe B : Présentation du logiciel R

Cette annexe présente le logiciel R sur lequel on a été tester les algorithmes ELBOW, largeur de silhouette et écart statistique présentés dans le chapitre 4. Il est choisi pour démontrer l'adaptabilité des algorithmes.

B.1 Origines :

Le logiciel R est un logiciel de statistique créé par Ross Ihaka et Robert Gentleman. Il est à la fois un langage informatique et un environnement de travail : les commandes sont exécutées grâce à des instructions codées dans un langage relativement simple, les résultats sont affichés sous forme de texte et les graphiques sont visualisés directement dans une fenêtre qui leur est propre.

C'est un clone du logiciel S-plus qui est fondé sur le langage de programmation orienté objet S. Ce logiciel sert à manipuler des données, à tracer des graphiques et à faire des analyses statistiques sur ces données.

B.2 Pourquoi utiliser R ?

Tout d'abord R est un logiciel gratuit et à code source ouvert. Il fonctionne sous Linux et Windows. Il est développé dans la mouvance des logiciels libres par une communauté sans cesse plus vaste de bénévoles motivés.

Tout le monde peut d'ailleurs contribuer à son amélioration en y intégrant de nouvelles fonctionnalités ou méthodes d'analyse non encore implémentées. Cela en fait donc un logiciel en rapide et constante évolution.

C'est aussi un outil très puissant et très complet, particulièrement bien adapté pour la mise en oeuvre informatique de méthodes statistiques. Il est plus difficile d'accès que certains autres logiciels du marché (comme Minitab par exemple).

L'avantage en est toutefois double :

- L'approche est pédagogique puisqu'il faut maitriser les méthodes statistiques pour parvenir à les mettre en oeuvre ;
- L'outil est très efficace lorsque l'on domine le langage R puisque l'on devient alors capable de créer ses propres outils, ce qui permet ainsi d'opérer des analyses très sophistiquées sur les données.

Bibliographie

[1] 1 B. Stein, S.Meyer zu, E.WiBbrock "On Cluster Validity and the Information Need of Users" ACTA Press, pp 216-221, 2003.

[2] 2 Bernard Desgraupes, "Clustering Indices", University Paris Ouest Lab Modal'X April 2013.

[3] 3 Ch. Yu Yen, K. J. Cios "Image recognition system based on novel measures of image similarity and cluster validity" Neurocomputing, 2008.

[4] 4 Christine Decaestecker ULB, et Marco Reus UCL, "Classification non-supervisée (automatique) Méthodes de regroupement (Clustering)", LINF2275

[5] 5 Christophe Genolini, "Partitionnement (clusterization)", 10 novembre 2010.

[6] 6 Clément Bernard-Pierre Caraveski, "Classification de données", Master Ingénierie Mathématiques, Université Claude Bernard Lyon 1, 2013-2014.

[7] 7 D. Chessel, J. Thioulouse et A.B. Dufour, "Introduction à la classification hiérarchique", Fiche de Biostatistique Stage 7. Biostatistique 2004.

[8] 8 K. K Swami and R.C Jain, "PAMC : Partitioning Around Medoids for Classification", Smart Ashok Technological Institute, Vidisha (MP)-464001, India, 2006

[9] 9 D. L. Davies and D. W. Bouldin, "A cluster separation measure" IEEE Trans. Patt. Anal Machine Intell, vol. PAMI-1, pp. 224-227, 1979.

[10] 10 Damien Nouvel, "Théorie de l'information et mesures d'entropie", Institut national des langues et civilisations orientales.

[11] 11 Faicel CHAMROUKHI, "Projet 3 : Algorithme des centres mobiles (K -means) pour la classification automatique", Licence 2 Sciences Pour l'Ingénieur, Université de Toulon, Année 2013-2014.

[12] 12 F. Kovács, C. Legány et A. Babos, "Cluster Validity Measurement Techniques", Department of Automation and Applied Informatics Budapest University of Technology and Economics Goldmann György tér 3, H-1111 Budapest, Hungary.

[13] 13 G. W. Milligan and M. C. Cooper. An examination of procedures for determiningthe number of clusters in a data set. Psychometrica, 1985.

[14] 14 J. C. Bezdek (1981) : "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York. 1981.

[15] 15 J.C. Dunn " Well separated clusters and optimal fuzzy partitions". J. Cybern. Vol 4, pp $95 - 104$, 1974.

[16] 16 J. B. AUBIN, Y. KADDOUCH et J. BENHAMMOU, "Classification de courbes dans le domaine de l'hydrologie urbaine", Master 1 - INGÉNIERIE MATHÉMATIQUES 2013 - 2014.

[17] 17 Lior Rokach et Oded Maimon, "Chapter 15 : CLUSTERING METHODS", Department of Industrial Engineering Tel-Aviv University.

[18] 18 M. Charad "NBClust : An R package for determining therelevent number of clusters in data set". Journal of statistical software. Volume 61, 2014.

[19] 19 Makhtar MBAO,Mémoire de stage de Master : "Distance Sémantique et Carte Conceptuelle", UNIVERSITÉ MONTPELLIER 2, Aout 2007.

[20] 20 M. BOUGUESSA, "Une approche objective pour déterminer le nombre de clusters dans le cadre de la classification floue non supervisée", FACULTÉ DES SCIENCES UNIVERSITÉ DES SHERBROOKE, Sherbrooke, Québec, Canada, mars 2005.

[21] 21 O. Ammor , N. Rais et K.Slaoui. "Détermination du nombre optimal de classes présentant un fort degré de chevauchement" REVUE MODULAD N : 37, pp 32-45, 2007.

[22] 22 P.J. Rousseeuw "Silhouettes : a graphical aid to the interpretation and validation of cluster analysis" Journal of Computational and Applied Mathematics. Vol 20. pp 53-65.

[23] 23 Periklis Andritsos, "Data Clustering Techniques Qualifying Oral Examination Paper", Department of Computer Science, March 11, 2002.

[24] 24 R. CHAFIKA et Dr.S. MESHOUL, "Mémoire Master : Le clustering des données : une nouvelle approche évolutionnaire quantique", Université Mentouri de Constantine, Algérie.

[25] 25 Ruggero G. Pensa, "Projet K-Means", 16 novembre 2006.

[26] 26 S. Guérif " Réduction de dimension en Apprentissage Numérique Non Supervisé " Thèse, université Paris 13, 2006.

[27] 27 S. Wu and T.W.S Chow, " clustering of self-organising map using a clustering validity index on inter-cluster and intra-cluster density " pattern recognition, vol37, pp 175-188 2004.

[28] 28 Soufiane Khedairia, THESE : "Contribution à la classification non supervisée : application aux données environnementales", BADJI MOKHTAR UNIVERSITY-ANNABA-UNIVERSITE BADJI MOKHTAR-ANNABA-Faculté des Sciences de l'Ingénieur Département d'Informatique, Année : 2013-2014.

[29] 29 Tibshirani R, Walther G, Hastie T Estimating the number of clusters in a data set via the Gap statistics, Journal of the Royal Statistical Society B, 63 :411-423. 2001

[30] 30 T. M. Kodinariya, Dr. P. R. Makwana, "Review on determining number of Cluster in K-Means Clustering", Volume 1, Issue 6, November 2013.

[31] 31 Yanchang Zhao, "R and Data Mining : Examples and Case Studies", April 26, 2013.

[32] 32 Zakwan KREIT, Thèse pour le Doctorat "Contribution à l'étude des méthodes quantitatives d'aide à la décision appliquées aux indices du marché d'actions". UNIVERSITÉ MONTESQUIEU BORDEAUX IV 2007.