

## I. Introduction:

Le datamining est une technologie qui regroupe plusieurs méthodes et techniques d'exploration de données. Ces méthodes sont classées selon deux modèles, le modèle prédictif dont les méthodes analysent des performances antérieures afin d'évaluer des comportements spécifiques, comme par exemple : l'apprentissage des règles d'associations [50][51][52] et l'apprentissage supervisé, dans lequel la classification par les arbres de décision et la régression linéaire appartiennent [49] ; et le modèle descriptif qui quantifie les relations entre les données afin de les classer ou les affecter à des groupes. Contrairement au modèle prédictif, le modèle descriptif est un processus plus exploratoire du datamining et tend à être itératif.

Le clustering ou l'apprentissage non supervisé [2][3][4], faisant partie du modèle descriptif du datamining a pour but de grouper ou de classer des objets selon certaines similitudes dans un contexte applicatif donné. Le processus consiste à former des groupes nommés clusters de telle sorte que les objets dans le même cluster sont similaires entre eux et dissimilaires aux objets des autres clusters. Le processus ne classe pas un objet parmi un ensemble de classes prédéfinies, car aucune connaissance n'est fournie sur les futures catégories qui peuvent être formées, ceci entre dans le cadre de l'apprentissage supervisé.

Nous sommes intéressés dans ce chapitre par l'apprentissage non supervisé (le clustering).

## II. Définition formelle du clustering:

Le problème de clustering peut être vu de la manière suivante:

Soit  $X = \{x_1, \dots, x_n\}$  un ensemble de  $n$  objets  $x_i$ , aussi appelés (items, exemples, prototypes, échantillons, points, entités, ...). Nous gardons dans tout ce qui suit l'appellation « items »

L'item  $x_i$  est décrit par une suite d'attributs:  $(x_{i1}, \dots, x_{ip})$  où  $p$  est considéré comme la dimension de l'espace des objets.

La tâche de clustering permet de générer un ensemble de  $t$  clusters  $C = \{C_1, \dots, C_t\}$ , tel que chaque cluster  $C_a$  est un sous ensemble de  $X$  ( $C_a \subset X$ ) et l'union des clusters couvre l'ensemble des objets de départ ( $\bigcup_{a=1}^t C_a = X$ ). Ce résultat est appelé schémas du clustering.

## III. Les étapes de clustering:

Le processus de clustering se divise en trois étapes majeures:

- Préparation des données.
- L'algorithme de clustering
- Exploitation du résultat de l'algorithme.

### III.1 La préparation des données:

Les attributs qui décrivent les items à grouper sont de différents types:

- **Symbolique (catégoriel):** ensemble limitativement énumérable de valeurs Symboliques avec éventuellement une notion d'ordre.
- **Numérique:** de type entier ou réel.

L'étape de préparation des données consiste à sélectionner et/ou pondérer des variables, voir à créer de nouvelles variables afin de mieux discriminer entre eux les objets à traiter. En effet les variables ne sont pas nécessairement toutes pertinentes: certaines peuvent être redondantes et d'autres non pertinentes pour la tâche ciblée. Dans cette étape, la mesure de similarité ou de dissimilarité entre les paires d'objets est aussi déterminée, le choix de cette mesure est déterminant pour la suite du processus. Chaque domaine d'application possède ces propres mesures, selon la nature des attributs décrivant les objets.

### **III.2 Le choix de l'algorithme de clustering:**

Le choix de l'algorithme de clustering doit donner lieu à une analyse globale du problème: quelle est la nature (qualitative ou quantitative) des données? Quelle est la nature des clusters attendus (nombre, forme, densité, ..., etc.) ? Et quelle est la quantité des données à traiter ?

### **III.3 L'exploitation des clusters:**

A la fin de l'algorithme de clustering, deux situations sont possibles, soit la tâche de clustering s'inscrit dans un traitement global d'apprentissage soit les clusters générés par clustering constituent un résultat final.

Dans le premier cas, l'analyse des clusters obtenus peut aider à orienter le traitement suivant. Une description des clusters n'est pas nécessaire dans cette situation. En revanche, dans le cas où le clustering constitue seul un processus global de découverte de classes, l'exploitation des clusters pour une application donnée passe par une description de ces dernières.

## **IV. Méthodes de clustering:**

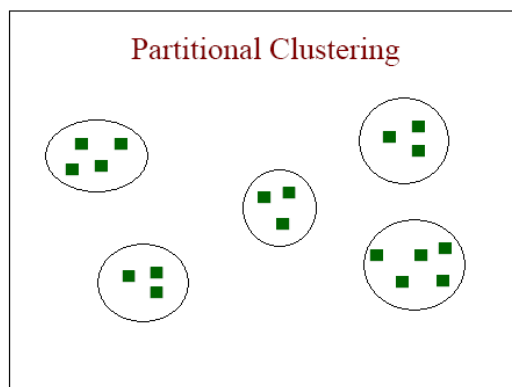
L'utilisation à facettes multiples du clustering a imposé plusieurs exigences sur les méthodes de clustering. Ces méthodes doivent être capables de traiter les différents types d'attributs et découvrir des clusters de formes différentes. A ne pas confondre algorithme de clustering et méthode de clustering, un algorithme est un moyen particulier pour implémenter une méthode. Plusieurs classifications des méthodes de clustering ont été proposées dans la littérature [4][53][54], les différentes études de synthèse proposent des organisations de ces méthodes selon plusieurs critères:

### **IV.1. Les structures produites (partitions vs hiérarchies):**

Il existe deux types de structures de clustering: hiérarchies et partitions:

#### **IV.1.1 Clustering par partitionnement:**

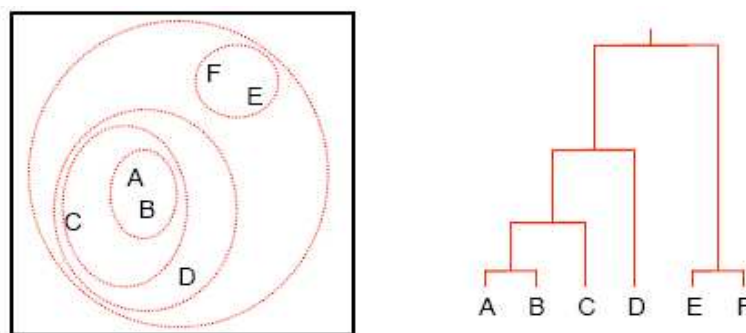
Le clustering par partitionnement regroupe les objets en des sous ensembles sans chevauchement de telle façon que chaque objet appartient exactement à un cluster (**Fig.3**). Le principe est de construire une seule partition de l'ensemble de départ. Un critère de regroupement est adopté comme par exemple: minimiser la fonction d'erreur quadratique. L'un des principaux algorithmes implémentant le clustering par partitionnement est l'algorithme des k-moyennes (k-means) [55].



**Fig.3 Clustering par partitionnement**

#### IV.1.2 Clustering hiérarchique:

Les approches de clustering hiérarchiques [3] produisent des séries imbriquées de partitions. Le principe est de construire un arbre de clusters (ou dendrogramme) (**Fig.4**). À partir de ce dendrogramme, il est possible d'obtenir une partition de  $X$  en coupant l'arbre à un niveau  $l$  donné. Une importante caractéristique des méthodes de clustering hiérarchique est l'impact visuel du dendrogramme, qui permet à l'analyste de données de voir quels sont les objets qui rejoignent les clusters ou les quittent dans les niveaux successifs de proximité. L'analyste de données peut alors décider si le dendrogramme entier décrit les données où il peut sélectionner un clustering dans un niveau fixe de proximité, qui donne un sens pour l'application dans la main.



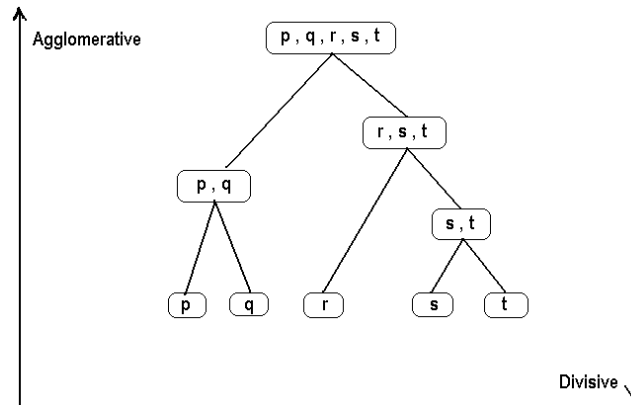
**Fig.4 Clustering hiérarchique.**

#### IV.2 Méthodes agglomératives (ascendantes) vs divisives (descendantes):

Le clustering hiérarchique peut être agglomératif ou divisif (**Fig.5**), le clustering agglomératif place chaque objet dans son propre cluster et élargit ce cluster atomique graduellement jusqu'à ce que tous les objets aient un seul cluster. Le clustering hiérarchique divisif renverse le processus en commençant avec un cluster qui regroupe tous les objets et le subdivise en de petits groupes. Pour chacune de ces deux méthodes, l'arbre hiérarchique n'est pas nécessairement construit totalement. Le

processus peut être stoppé lorsque le nombre de clusters désiré est atteint ou lorsqu'un seuil de qualité est dépassé.

Il existe peu d'algorithmes en clustering divisif, notamment à cause de la difficulté à définir un critère de séparation d'un cluster, l'algorithme le plus connu parmi d'autres est l'algorithme DIANA (DIvisive ANALysis), et en clustering agglomératif, l'algorithme AGNES [3].



**Fig.5 Clustering hiérarchique divisif et agglomératif**

#### IV.3 Clustering dur, clustering flou, clustering avec recouvrement:

Dans le clustering dur ou (hard clustering), un objet appartient exactement à un et un seul cluster. Le résultat prend alors la forme de partitions ou de hiérarchies strictes. Le clustering flou ou (fuzzy clustering) correspond à une représentation plus souple de l'organisation des données. Dans ce formalisme, chaque objet  $x_i$  est affecté de façon fractionnaire à un cluster. Un objet appartient à un ou plusieurs groupes suivant un degré d'appartenance. Ce type de formalisme permet de tenir en compte des incertitudes et ambiguïtés pouvant intervenir dans l'appartenance d'un objet à une classe. L'utilisation d'algorithmes de clustering flou est très fréquente dans les domaines d'application tels que le regroupement de données textuelles ou la segmentation d'image. L'algorithme le plus utilisé en clustering flou est l'algorithme FCM (*fuzzy-c-means*) [5][56] qui est une extension de l'algorithme k-means.

Pour combiner les avantages des deux types de clustering et en éviter les limitations, le clustering avec recouvrement, ou soft clustering propose une affectation dure de chaque objet à une ou plusieurs classes. Par exemple, grouper des individus par âge ou par sexe est exclusif, tandis que grouper par catégories de maladies est non exclusif car une personne peut avoir plusieurs maladies simultanément. L'un des algorithmes appliquant cette méthode est l'algorithme PoBOC (*Pôle-Based Overlapping Clustering*) [57].

Nous nous intéressons dans ce qui suit par le clustering par partitionnement en particulier l'algorithme des k-moyens (k-means), pour cela, nous présentons quelques notions de base sur lesquelles le clustering par partitionnement est fondé.

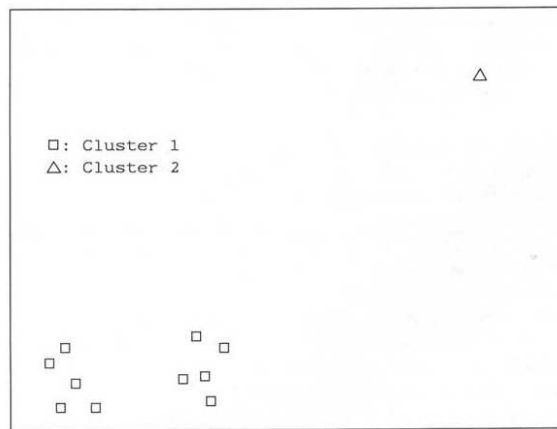
#### V. Notions de base:

##### V.1. Centroïdes:

Le centroïde est défini comme le centre de gravité d'un cluster, C'est un objet qui ne fait généralement pas partie des objets constituant le cluster. Chaque valeur d'un attribut du centroïde représente la moyenne des valeurs de l'attribut correspondant pour tous les objets du cluster.

### V.2. Outliers:

Un outlier est un objet suffisamment différent du reste de l'ensemble des objets à un point que l'on soupçonne qu'il est inclus par erreur, comme par exemple une erreur dans le processus de mesure ou d'encodage de données. Les outliers peuvent fournir des informations utiles sur le processus de génération de données mais aussi ils peuvent déformer le schéma de clustering, comme illustré dans la figure 6.



**Fig.6 L'effet des outliers sur le résultat du clustering**

### V.3. Mesure de similarité:

Les méthodes de clustering ont besoin d'un indice de proximité qui doit être établis entre chaque pair d'objets. Une matrice de proximité ou de dissimilarité  $[d(i,j)]$  accumule par pair, les indices de proximité dans une matrice dans laquelle les lignes et les colonnes représentent les objets.

L'indice de proximité peut être une similarité ou une dissimilarité. Les  $i$ -ème et  $j$ -ème objets qui se ressemblent le plus ont le plus grand indice de similarité et le plus petit indice de dissimilarité. Par exemple, la distance euclidienne entre deux objets est un indice de dissimilarité, tandis que le coefficient de corrélation est un indice de similarité.

L'indice de proximité le plus utilisé est la mesure de distance ou de dissimilarité. Une mesure de dissimilarité est une fonction  $d$  qui associe pour chaque pair d'objets de l'ensemble  $X$  une valeur positive qui représente la distance entre les deux objets. Cette fonction vérifie les propriétés suivantes:

#### Positivité:

$$\forall x, y \in X, d(x,y) \geq 0$$

#### Symétrie:

$$\forall x, y \in X, d(x,y) = d(y,x)$$

**Identité:**

$\forall x, y \in X, d(x, y) = 0$  seulement si  $x=y$

**Inégalité triangulaire:**

$\forall x, y, z \in X, d(x, z) \leq d(x, y) + d(y, z)$

**a) Similarité des objets à attributs numériques:**

La distance la plus connue et la plus utilisée est la distance de Minkowski définie par:

$$d(x, y) = \left[ \sum_{j=1}^p |x_j - y_j|^r \right]^{1/r}$$

Pour  $r=2$  : on parle de distance euclidienne

$$d(x, y) = \sqrt{\sum_{j=1}^p (x_j - y_j)^2}$$

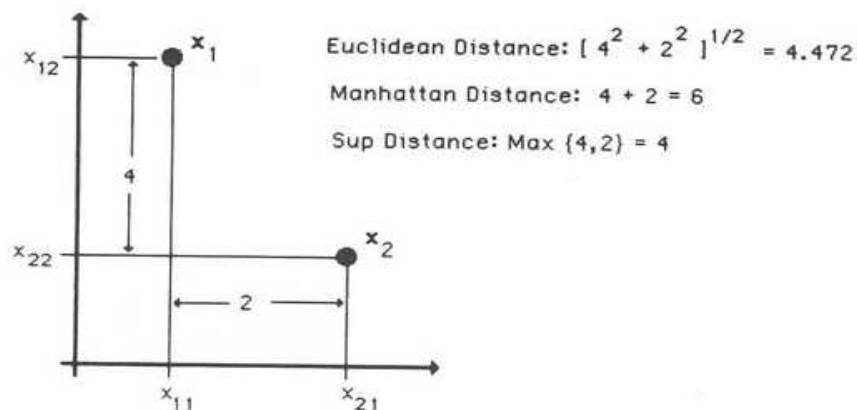
Pour  $r=1$ : distance de Manhattan

$$d(x, y) = \sum_{j=1}^p |x_j - y_j|$$

Pour  $r= \infty$ : distance de Chebechev

$$d(x, y) = \max_{1 \leq j \leq p} |x_j - y_j|$$

La figure 7 illustre les différentes distances:



**Fig. 7 Les distances de Minkowski**

La distance euclidienne est la plus répandue parmi les distances de Minkowski, elle mesure la distance directe entre deux points dans l'espace  $\mathbf{R}^p$

**b) Similarité des objets à attributs catégoriels:**

Lorsque les attributs ne sont pas numériques, les mesures de dissimilarités décrites précédemment ne peuvent pas être calculés. Dans ce cas, une redéfinition de l'espace de description des objets à l'aide d'attributs binaires est nécessaire. Pour convenance, toutes les valeurs d'attributs deviennent 1 ou 0. Si par exemple "1" désigne "large" pour le premier attribut et "0" pour "petit", alors "1" doit dénoter "large" pour tous les autres attributs mesurant la taille. L'indice de proximité entre les  $i$ -ème et  $k$ -ème objets est calculé par le comptage des propriétés partagées ou non pour les deux objets. Deux indices sont proposés, l'indice de Rand [58] noté R et l'indice de Jaccard [59] noté J.

Le coefficient de Rand:  $d(x,y) = (a_{00} + a_{11})/(a_{00}+a_{11}+a_{10}) = (a_{00} + a_{11})/p$

Le coefficient de Jaccard:  $d(x,y) = a_{11}/(a_{11}+a_{01}+a_{10}) = a_{11}/p-a_{00}$

où  $a_{11}$  est le nombre d'attributs qui ont pour valeurs "1" pour les deux objets, et  $a_{10}$  est le nombre d'attributs qui ont pour valeur "1" pour l'objet x, et "0" pour l'objet y.  $p$  étant le nombre des attributs.

Cette solution exige une nouvelle représentation des valeurs d'attributs en binaire ce qui ne convient pas pour toutes les applications, d'autres mesures de dissimilarité sont proposées dans la littérature, le champ d'application de l'algorithme de clustering est le meilleur guide pour choisir l'indice de proximité. Autres métriques sont aussi utilisées

**V.4 Les distances: intra – cluster, inter – cluster et la variance d'un cluster:****a. La distance intra-cluster:**

La distance intra-cluster correspond à la somme des carrés des distances au centroïde du cluster:

$$I_{\text{intra}}(C_a) = \sum_{x_i \in C_a} d(x_i, x_a^*)^2$$

**b. La distance inter - cluster:**

Etant donné un schéma de clustering, la distance inter-clusters correspond à la somme des carrés des distances entre les centroïdes des clusters:

$$I_{\text{inter}}(C) = \sum_{i=1}^t \sum_{j < i} d(x_i^*, x_j^*)^2$$

**c. La variance d'un cluster:**

La variance d'un cluster est égale à la moyenne des carrés des distances au centroïde:

$$V(C_a) = \frac{1}{|C_a|} \sum_{x_i \in C_a} d(x_i, x_a^*)^2 \quad \text{où } |C_a| \text{ est le cardinal du cluster } C_a$$

**VI. L'algorithme de clustering k-means:****VI.1 Présentation générale de l'algorithme k-means:**

Parmi les formulations de clustering par partitionnement, basés sur la minimisation d'une fonction objective, l'algorithme des k-moyennes (k-means), est le plus largement utilisé et étudié, proposé pour la première fois en 1967 par MacQueen [55].

Soit un ensemble de  $n$  items dans un espace réel de p-dimension  $\mathbf{R}^p$  et un entier  $k$ , le problème est de déterminer un ensemble de  $k$  objets dans  $\mathbf{R}^p$  appelés des centroïdes de telle manière de minimiser la moyenne des distances carrées de chaque objet à son plus proche centroïde. L'une des plus populaires heuristiques pour résoudre le problème k-moyennes est basée sur un simple schéma itératif pour trouver une solution minimale locale. Cet algorithme est souvent appelé "l'algorithme k-moyennes".

L'idée principale est de définir  $k$  centroïdes, le meilleur choix est de placer chacun plus loin de l'autre le maximum possible. La prochaine étape consiste à prendre chaque objet et l'associer au centroïde le plus proche. A ce moment de nouveaux  $k$  centroïdes vont être calculés. Et une autre allocation d'objets est faite par rapport aux nouveaux centroïdes. De ce fait, une boucle est générée. Comme résultat de cette boucle, les centroïdes changent leurs locations étape par étape, jusqu'à ce qu'aucun changement n'est donné.

## VI.2 L'algorithme des k-moyennes (k-means):

Entrée:  $k$  le nombre de clusters tel que  $0 < k < n$

Sortie: Une partition  $\mathcal{C} = \{C_1, \dots, C_k\}$

1<sup>ère</sup> étape: Sélection d'une partition initiale:

1.1- choisir aléatoirement dans  $X$ ,  $k$  objets centres :  $x_{1,0}^*, \dots, x_{k,0}^*$

1.2- Constitution d'une première partition initiale  $\mathcal{C}_1 = \{C_{1,1}, \dots, C_{k,1}\}$  en affectant chaque objet  $x_i \in X$  à son centre le plus proche:

$$C_l = \left\{ x_i \in X / d(x_i, x_{l,1}^*) = \min_{h=1, \dots, k} d(x_i, x_{h,1}^*) \right\}$$

2<sup>ème</sup> étape: Mettre à jour la partition:

2.1- Calcul des centroïdes des  $k$  nouvelles partitions (clusters) obtenus:

$$x_{1,t}^*, \dots, x_{k,t}^*$$

2.2- Constitution d'une nouvelle partition  $\mathcal{C}_t = \{C_{1,t}, \dots, C_{k,t}\}$  en affectant chaque objet  $x_i \in X$  au centroïde le plus proche:

$$C_l = \left\{ x_i \in X / d(x_i, x_{l,t}^*) = \min_{h=1, \dots, k} d(x_i, x_{h,t}^*) \right\}$$

3<sup>ème</sup> étape: - Répéter les étapes 2.1 et 2.2 jusqu'à ce qu'aucun changement ne s'opère d'un schéma de clustering  $\mathcal{C}_t$  à un schéma  $\mathcal{C}_{t+1}$

- Retourner le schéma de clustering final  $\mathcal{C}_{finale}$

## VI.3 Détails des étapes de l'algorithme:

### VI.3.1 Sélection d'une partition initiale:

La partition initiale est formée en choisissant  $k$  objets (centroïdes) aléatoirement. Pour avoir des résultats satisfaisants, il est préférable de les prendre



bien séparés, chacun des autres. Ensuite les  $(k - n)$  objets sont affectés, chacun à son plus proche centroïde. Des partitions initiales différentes mènent à des schémas de clustering différents car l'algorithme vise à minimiser la variance intra-clusters. Cependant, il est tout à fait possible qu'une autre configuration initiale des partitions puisse mener à un schéma de clustering rendant encore plus faible la variance intra-clusters. Ce processus peut converger donc à un minimum local. Une manière de pallier cet inconvénient est d'exécuter l'algorithme plusieurs fois avec des partitions initiales différentes. Et prendre le meilleur schéma de clustering trouvé.

### VI.3.2 Mettre à jour la partition:

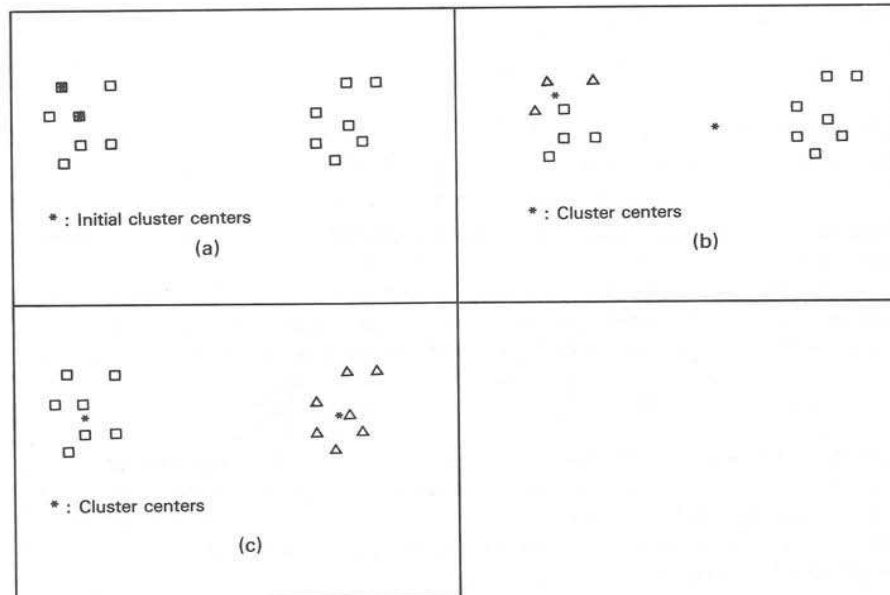
Après que la première partition est construite, les nouveaux centroïdes sont calculés. A chaque itération, les clusters sont mis à jour en leur réaffectant les objets selon leurs positions par rapport aux nouveau centroïdes afin de réduire la variance intra-clusters. La distance euclidienne est la mesure de dissimilarité la plus utilisée pour calculer la distance entre les objets et les centroïdes.

### VI.3.3 Convergence:

L'algorithme k-means s'arrête lorsqu'il ne peut plus baisser la valeur de la variance intra-clusters ou aussi appelé critère des moindres carrés:

$$V = \sum_{h=1}^k \sum_{i=1}^n d(x_i, x_h^*)^2$$

En pratique, l'algorithme k-means converge rapidement. Dans [60], les auteurs ont prouvé rigoureusement la convergence de k-means. Cependant, Il n'existe pas une garantie que l'algorithme atteint un minimum global. La figure 8 montre les itérations de l'algorithme k-means.



**Fig. 8 Convergence de k-means: (a) partition initiale; (b) les clusters candidats après la première itération; (c) les clusters candidats après la seconde itération.**

### VI.3.4 Complexité:

La complexité de l'algorithme des *k*-moyennes est de l'ordre  $O(npkt)$ , où  $n$  est le nombre d'objets,  $p$  le nombre d'attributs,  $k$  est le nombre de clusters désiré et  $t$  le nombre d'itérations. La valeur de  $t$  dépend de l'initialisation des centres de clusters, la distribution des objets et la taille du problème de clustering.

### VI.4 Discussion sur l'algorithme *k*-means:

K-means est un algorithme de clustering par partitionnement très efficace, et parmi les algorithmes les plus utilisés, ceci est dû à sa simplicité conceptuelle, sa rapidité et ses faibles exigences en ressources mémoire. Cependant l'algorithme souffre de certains défauts:

- La valeur de  $k$ : le nombre de clusters, doit être choisi à priori. Ce choix peut être fait par un simple examen visuel dans le cas de données bidimensionnelles, mais il n'en est pas le cas pour les données de dimension supérieure. Une manière pour résoudre ce problème est d'éclater ou fusionner des clusters dans le schéma final du clustering, si le nombre de clusters désiré n'est pas approprié. La méthode ISODATA (Iterative Self-Organizing Data Analysis Technique) [61] est parfois utilisée pour affiner les clusters obtenus par un algorithme de partitionnement. Les clusters sont fusionnés ou éclatés selon des paramètres définis par l'utilisateur.

- Pour une valeur donnée de  $k$ , le schéma de cluster final dépend de la configuration initiale des centres de clusters sélectionnés (partition initiale). Une solution consiste à comparer les schémas de clustering finaux résultant de l'exécution multiple de l'algorithme pour des configurations initiales différentes et choisir le meilleur selon le critère de la minimisation de la variance intra-clusters.

- L'algorithme *k*-means se restreint à traiter des objets à attributs numériques permettant ainsi le calcul des distances et les représentants des clusters (i.e centroïdes ou centres de clusters), ceci est facilement accomplis dans l'espace euclidien. Ainsi dans le cas des données catégoriels, il est parfois nécessaire de les convertir en des vecteurs à valeurs numériques ce qui défausse la structure de l'information, par exemple lorsqu'on utilise le modèle vecteur pour représenter un document. Cependant, plusieurs variantes de l'algorithme pour ce type de données ont été proposées, le principe est de choisir un représentant du cluster parmi ses éléments appelé médoïde au lieu de calculer le centroïde (le centre du cluster), la théorie des graphes est souvent utilisée dans ces méthodes pour le calcul de dissimilarité entre les objets. L'algorithme des *k*-médoïdes est la version *k*-means utilisant ce principe et ses variantes: l'algorithme PAM (Partitioning Around Medoids) [62] et la méthode des nuées dynamiques [63]. Il est noté que la recherche des médoïdes étant plus coûteuse que le simple calcul des centroïdes.

- L'algorithme *k*-means est très sensible à la présence des outliers.

- L'algorithme *k*-means construit des clusters convexes en affectant les objets à un centroïde, les clusters sont totalement disjoints.

**VII. Conclusion:**

Dans ce chapitre nous avons présenté l'algorithme de clustering k-means qui fait partie de la méthode de clustering par partitionnement. L'algorithme k-means produit une partition de l'ensemble de données par étapes successives de réallocations simples des entités aux centroïde qui sont les représentants des clusters. Il s'agit le plus souvent d'une entité centrale pour chaque cluster. Les entités sont groupées alors en minimisant ainsi la somme des carrés des distances entre les données et les centroïdes des clusters correspondants. Comme mesure de distance, k-means est souvent implémenté avec la formule euclidienne.

L'algorithme k-means a prouvé son efficacité lorsqu'il est exécuté sur un ensemble de données dans un même site. Le besoin de préservation de privacy dans l'algorithme k-means intervient lorsqu'il est exécuté sur un ensemble de données réparti sur plusieurs sites, qu'on nomme « parties », et que l'on souhaite faire du clustering sur l'union de leurs ensembles de données. L'objectif est d'empêcher une partie de voir ou d'en déduire les données d'une autre partie durant l'exécution de l'algorithme sur l'union de leurs ensembles de données. Ceci est réalisé en se basant sur le calcul multi partie sécurisé qui fournit une méthode formelle pour préserver la privacy des données. Ce qui sera l'objet du chapitre suivant.