

Cours de Machine Learning

Classification non supervisée

M2 Informatique pour la Science des Données –2020-2021
Université Paris Saclay, D. Jeannel
ML.M2.ISD.Orsay@gmail.com

Classification non supervisée:
algorithmes k-means, PAM et
CLARA

Sommaire

- Objectif et Notions de base
 - Distance
 - Dissimilarité
- Algorithmes
 - Méthode classique : K-means
 - Méthodes robustes : PAM, CLARA
- Méthodes de validation de classification
- Applications numériques

Objectif et notions de base

- Objectif : déterminer une partition dans un jeu de données i.e. identifier des groupes de données dont les caractéristiques sont similaires
 - Exemple : en marketing, comportement d'achats de clients
- Conditions d'utilisation
 - Variables quantitatives
- Algorithmes de classification
 - Méthodes des k-means
 - Objectif : déterminer k groups distincts parmi un jeu de données
 - Fonctionnement :
 - Etape 0 : fixer le nombre de classes a priori K
 - Etape 1 : sélectionner aléatoirement K points appelés centres de classes G_k
 - Etape 2 : jusqu'à convergence répéter les phases suivantes :
 - Allocation : affecter chaque point x_i au centre de classes G_k le plus proche
 - Représentation : réactualiser les centres des classes G_k à partir des données attachées
 - Répéter jusqu'à ce que les centres de classes ne changent plus

Classification non supervisée : notions de base

- Objectif : déterminer une partition dans un jeu de données i.e. identifier des groupes de données dont les caractéristiques sont similaires

- Exemple : en marketing, comportement d'achats de clients

- Conditions d'utilisation

- Variables quantitatives

- Notions de distance

- Distance Euclidienne

$$d(x_i, y_i) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Distance de Manhattan

$$d(x_i, y_i) = \sum_{i=1}^n |x_i - y_i|$$

NB : Distance euclidienne plus sensible aux valeurs aberrantes que distance de Manhattan

Classification non supervisée : notions de base

▪ Notions de dissimilarités

- Dissimilarité de Pearson

$$d_{corr}(x_i, y_i) = 1 - corr(x_i, y_i)$$

Si 2 observations sont assez semblables (très différentes), dissimilarité tend vers zéro (1 respectivement)

- Dissimilarité de Spearman

$$d_{rank}(x_i, y_i) = 1 - corr(rank(x_i), rank(y_i))$$

▪ En pratique

- Transformation des données brutes en données centrées réduites

$$x_i \rightarrow \frac{x_i - \bar{x}}{\sigma_x}$$

Moyenne

Écart-type

- Distance Euclidienne très utilisée mais sensibilité forte aux valeurs aberrantes
- Dissimilarité de Spearman privilégiée

➔ Choix dépend des données et des problématiques :

- Génétique : distance de Manhattan
- Marketing : recherche de profils dissimilarité de Spearman

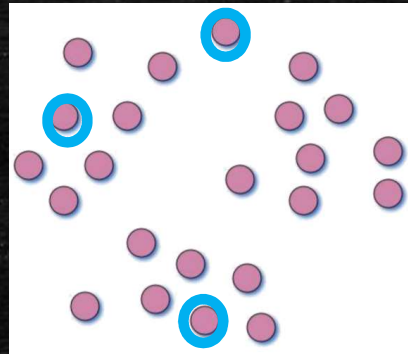
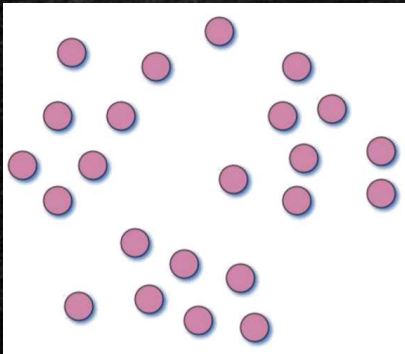
Classification non supervisée : algorithme des k-means

- Algorithmes de classification
 - Méthodes des k-means
 - Objectif : déterminer k groups distincts parmi un jeu de données
 - Condition : variables quantitatives
 - Fonctionnement :
 - Etape 0 : fixer le nombre de classes a priori K
 - Etape 1 : sélectionner aléatoirement K points appelés centres de classes G_k
 - Etape 2 : jusqu'à convergence répéter les phases suivantes :
 - Allocation : affecter chaque point x_i au centre de classes G_k le plus proche
 - Représentation : réactualiser les centres des classes G_k à partir des données attachées
 - Répéter jusqu'à ce que les centres de classes ne changent plus

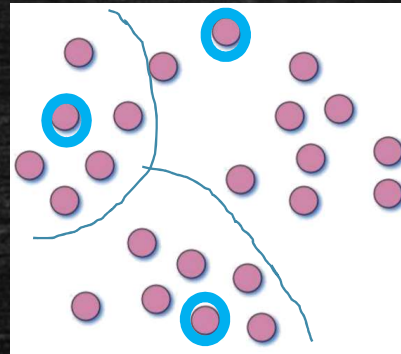
Classification non supervisée : algorithme des k-means

- Fonctionnement de l'algorithme
 - On fixe $K = 3$ le nombre de classes

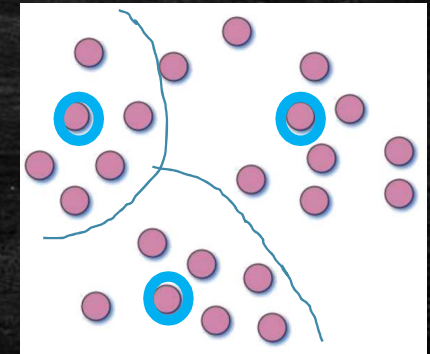
Etape 0 : sélection aléatoire des centres de classes



Etape 1 : allocation des points aux centres de classes

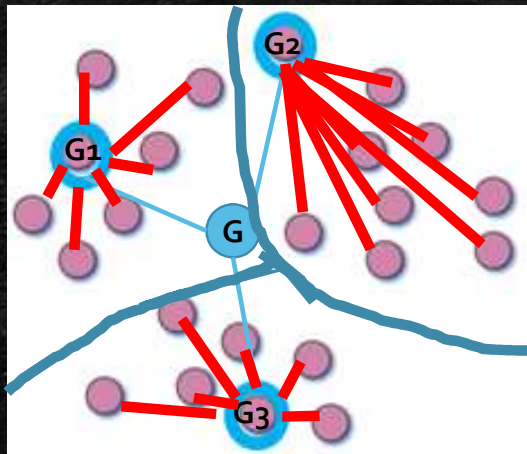


Etape 1 : actualisation des centres de classes



Classification non supervisée : algorithme des k-means

- Critère de sélection du nombre de classes
 - Définition inertie :
 - distance au carré d'un point par rapport à un autre (définition statistique)



- Théorème de Huygens
 - Inertie totale d'un nuage (T) =
inertie des centres de classes par rapport au centre du nuage (B = inertie inter classe)
+ somme des inerties des points d'une classe par rapport à leur centre de classe Gk (W = inertie intra classe)

Inertie totale = Inertie inter - classes + Inertie intra - classe

$$T = B + W$$

$$\sum_{i=1}^n d^2(i, G) = \underbrace{\sum_{k=1}^K n_k d^2(G_k, G)}_{\text{Dispersion des barycentres conditionnels autour du barycentre global. Indicateur de séparabilité des classes.}} + \underbrace{\sum_{k=1}^K \sum_{i=1}^{n_k} d^2(i, G_k)}_{\text{Dispersion à l'intérieur de chaque groupe. Indicateur de compacité des classes.}}$$

Dispersion des barycentres conditionnels autour du barycentre global. Indicateur de séparabilité des classes.

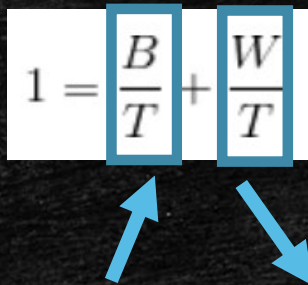
Dispersion à l'intérieur de chaque groupe. Indicateur de compacité des classes.

Classification non supervisée : algorithme des k-means

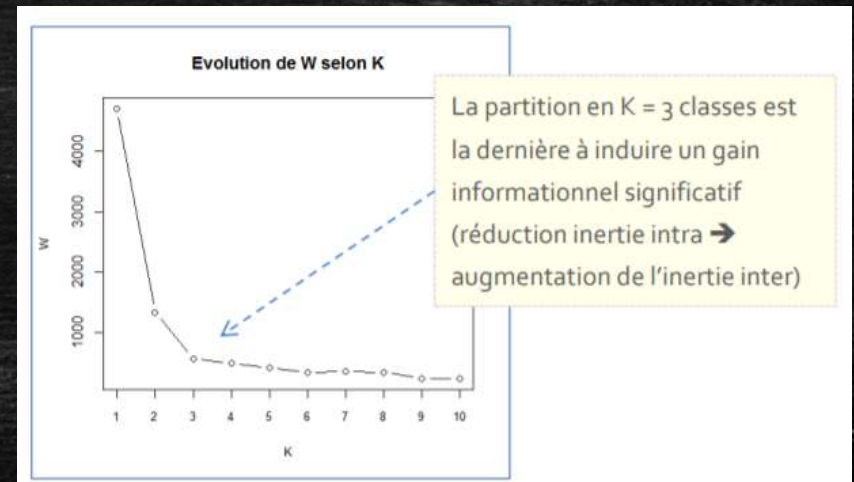
- Critère de sélection du nombre de classes
 - En pratique sélection du nombre de classes de telle sorte que

Ratio B/T à maximiser

⇔ Minimiser ratio W/T étape 0 : sélection aléatoire des centres de classes

$$1 = \frac{B}{T} + \frac{W}{T}$$


- Sélection du nombre de classes par analyse graphique en représentant le ratio W/T en fonction du nombre de classes (ou le ratio B/W)



- Convergence rapide de l'algorithme mais sensible aux valeurs aberrantes (défaut)

Classification non supervisée : algorithme PAM

- Idée de base :
 - réduire l'impact de la sensibilité des valeurs aberrantes de l'algorithme k-means
 - Introduction d'une fonction objective dans la détermination des classes
 - ➔ PAM (Partitioning Around Medoids)
- Vocabulaire :
 - Médoïde : individu représentatif d'une classe (différent du barycentre d'une classe)
 - Silhouette : indice de qualité de la partition obtenue (inertie intra-classe pour l'algorithme k-means)

Classification non supervisée : algorithme PAM

- Déroulement en 3 phases

1. Phase INIT

- Choix aléatoire des k médoïdes (k fixé)

2. Phase BUILD

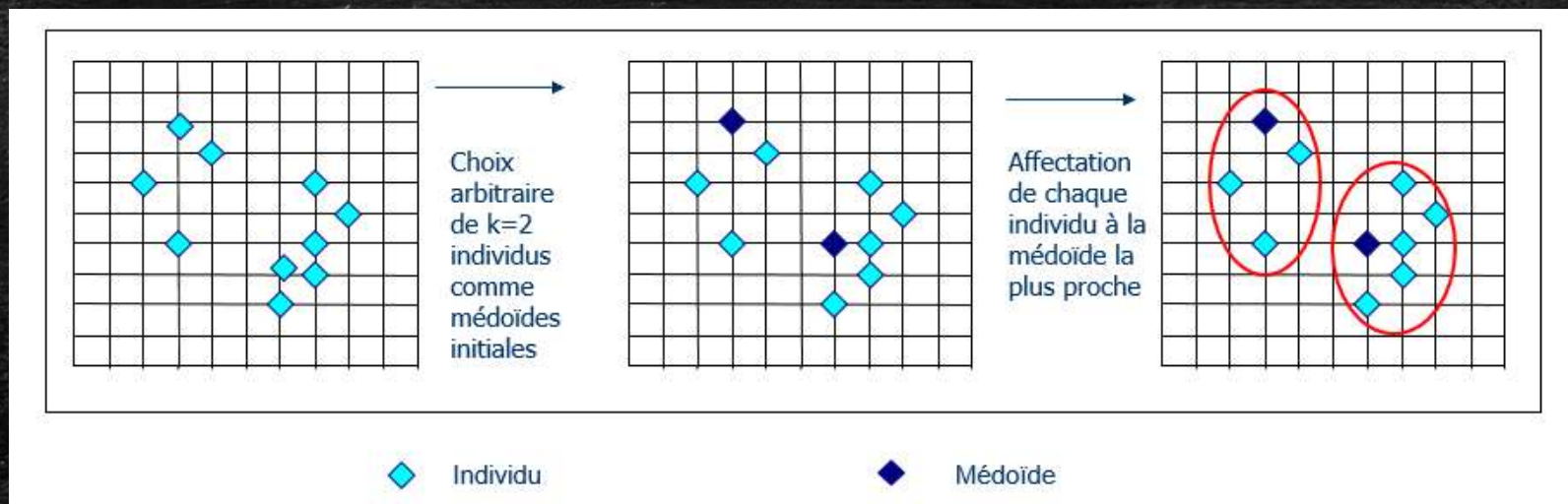
- Identification des nœuds ou éléments représentatifs (médoïdes)
- Construction de la partition

3. Phase SWAP

- Amélioration du choix des médoïdes

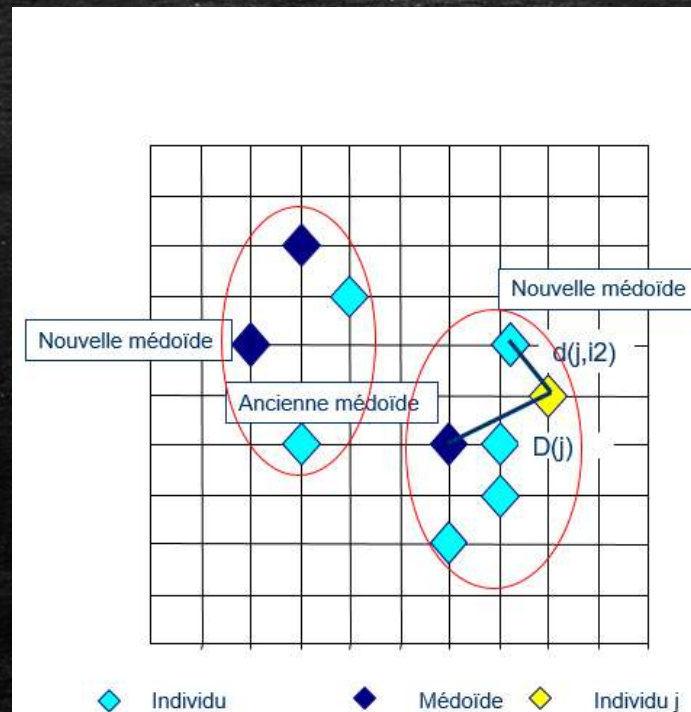
Classification non supervisée : algorithme PAM

- Phase INIT
 - Choix aléatoire des médoïdes



Classification non supervisée : algorithme PAM

- Phase BUILD
 - Identification des médoïdes et des groupes



IDENTIFICATION DES MEDOÏDES

Gain

$$G = \sum_j C_{j,i} = \sum_j (D(j) - d(j, i))$$

$D(j)$ distance de l'individu j à son ancienne médoïde
 $d(j, i)$ distance de l'individu j à la médoïde i

Individu j affecté à la médoïde i si $D(j) > d(j, i)$ (gain positif)

Choix des médoïdes

→ maximisation du gain

$$\text{Max}_i \sum_j C_{j,i}$$

Exemples

- Position par rapport à la médoïde $i1$:**
 $D(j) < d(j, i1) \rightarrow$ gain négatif.
Individu j affecté à son ancienne médoïde $i1$
- Position par rapport à la médoïde $i2$:**
 $D(j) > d(j, i2) \rightarrow$ gain positif.
Individu j affecté à la nouvelle médoïde $i2$.

Classification non supervisée : algorithme PAM

- Phase SWAP
 - Amélioration des médoïdes

AMELIORATION CHOIX DES MEDOÏDES

Coût

$$C = \sum_j C_{j,i,h} = \sum_j (d(j,h) - d(j,i))$$

$$\begin{cases} d(j,h) & \text{distance de l'individu } j \text{ à la médoïde } h \\ d(j,i) & \text{distance de l'individu } j \text{ à la médoïde } i \end{cases}$$

Considération de couples de médoïdes (i,h) avec

- i médoïde i sélectionné ;
- et h nouvelle médoïde candidate

→ minimisation du coût

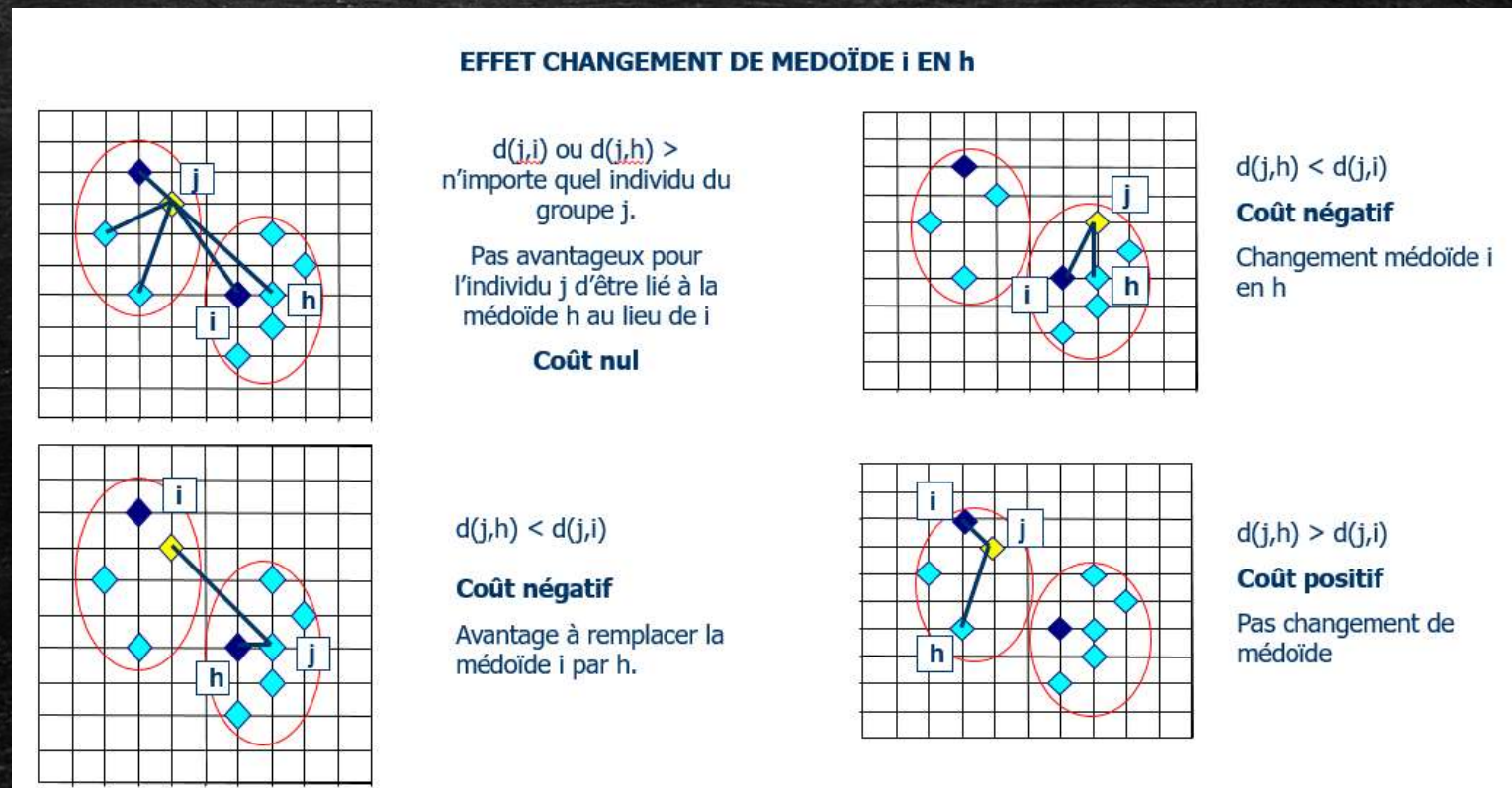
$$\text{Min}_{i,h} \sum_j C_{j,i,h}$$

Changement de médoïde si coût négatif

Plusieurs cas de figures

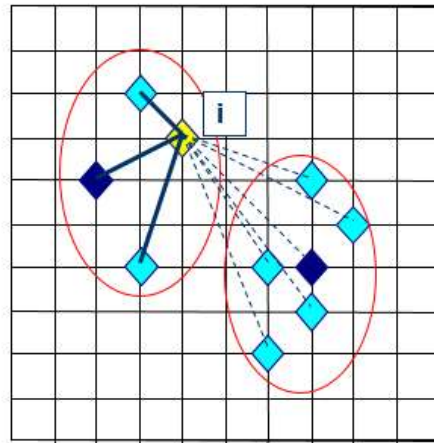
Classification non supervisée : algorithme PAM

- Phase SWAP
 - Amélioration des médoïdes



Classification non supervisée : algorithme PAM

- Critère de qualité de la partition obtenue



Calcul de paramètre s_k

- pour tout individu : calcul de s_i

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

——— distance moyenne de l'individu i à tous les individus de la classe dans laquelle i appartient (a_i)

- - - - distance moyenne de l'individu à tous les individus de la classe la plus proche (b_i)

- par classe : calcul de s_k

→ s_k : moyenne des $s_i \in$ à la classe k

- pour l'ensemble des classes : calcul de SC

→ SC : moyenne des s_k

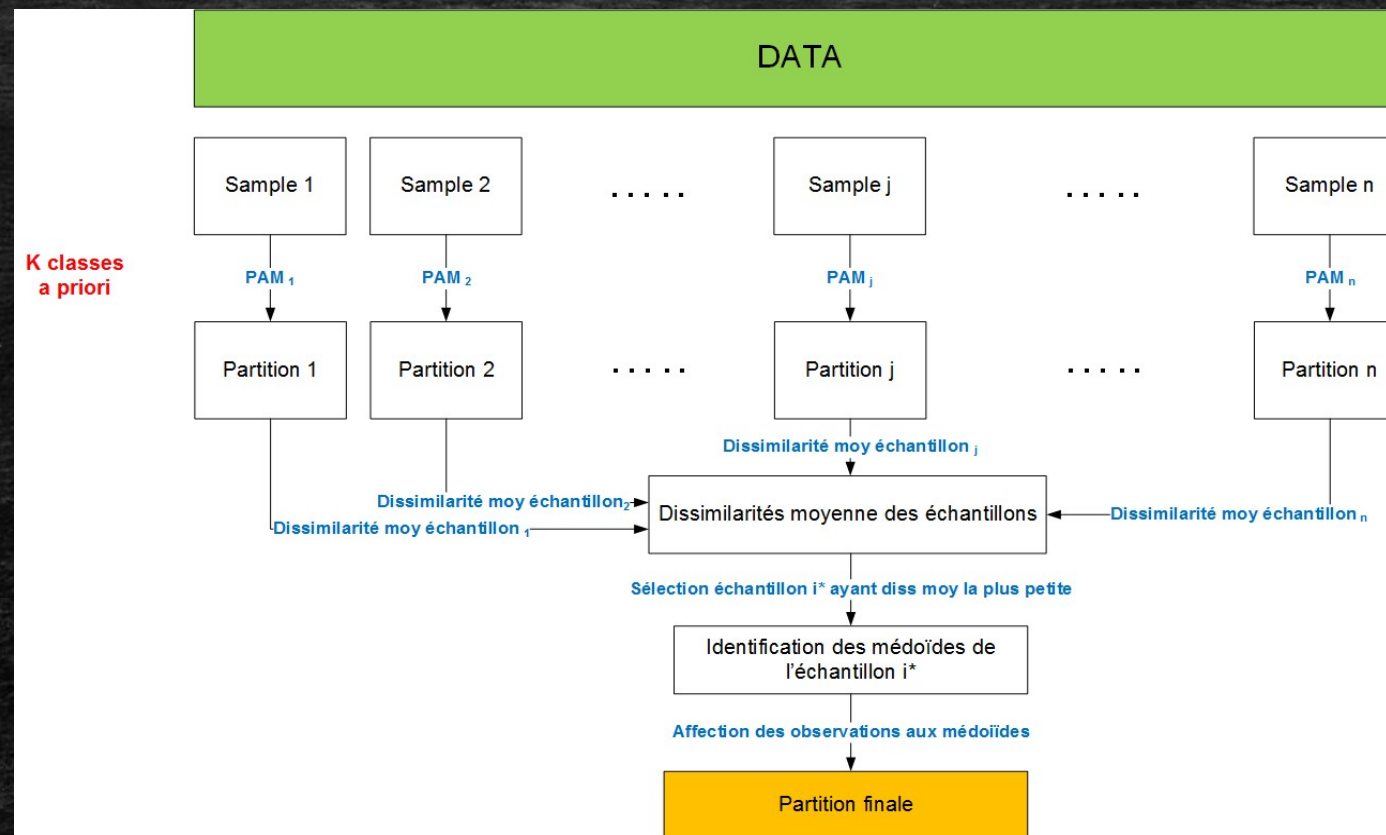
SC	Interprétation
0,71-1	Une structure forte a été trouvée.
0,51-0,70	Une structure raisonnable a été trouvée
0,26-0,50	La structure de partition est faible.
$\leq 0,25$	Pas de structure substantielle n'a été détectée

Classification non supervisée : algorithme CLARA

- CLARA : Clustering LARge Applications
 - Algorithme adapté pour les grands jeux de données
- Basé sur l'algorithme PAM

Classification non supervisée : algorithme CLARA

- Description de l'algorithme



Classification non supervisée : validation

■ Méthodes de validation :

- Test statistique de Hopkins

- Construction :

- Jeu de données D

- Tirage d'observations de D (p_1, p_2, \dots, p_n)

- Pour chaque p_i , calcul de la distance la plus proche / aux autres points p_j

$$x_i = \text{dist}(p_i, p_j)$$

- Calcul de la somme des distances mini des observations (p_1, \dots, p_n) :

- Tirage d'observations uniformément selon la même variation des données D (q_1, q_2, \dots, q_n)

- Pour chaque q_i , calcul de la distance la plus proche / aux autres points q_j

- Calcul de la somme des distances mini des observations (q_1, \dots, q_n) :

$$y_i = \text{dist}(q_i, q_j)$$

- Calcul statistique de test H :

- H proche de 0 \rightarrow il existe une partition dans D

$$H = \frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n x_i + \sum_{i=1}^n y_i}$$

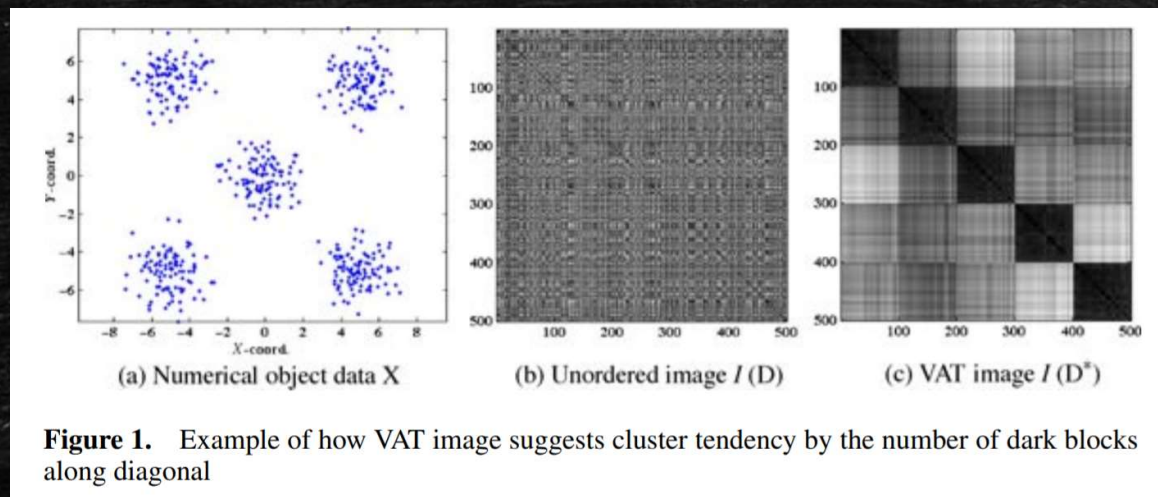
H_0 : D uniformément distribué \Leftrightarrow pas de partition distincte

H_1 : D non uniformément distribué, présence de segmentation

Classification non supervisée : validation

- Méthodes de validation :

- Représentation graphique (algorithme VAT : Visual Assessment of cluster Tendency)
- Etapes de calcul :
 - Calcul des dissimilarités des observations entre elles → matrice DM
 - Ordonnancement des observations les plus proches par dissimilarités → matrice ODM
 - Représentation graphique de la matrice ODM



Nombre de blocks
le long de la
diagonale de la
matrice ODM

Applications numériques

▪ Algorithme k-means

```
library(cluster)
# kmeans
set.seed(123)
km.res <- kmeans(df, 4, nstart = 25)
print(km.res)
```

Nombre de classes a posteriori

Nombre de répétition de l'algorithme avec génération de points initiaux (sélection de la meilleure partition en fonction du critère de minimisation de la variance intra-classe)

▪ Algorithmes PAM et CLARA

```
library(cluster)
# PAM
pam.res <- pam(df, 2)
pam.res$medoids

# CLARA
clara.res <- clara(df, 2, samples = 50, pamLike = TRUE)
clara.res$medoids
```

Nombre de classes

▪ Méthodes de validation

```
# Test de Hopkins
library(clustertend)
hopkins(df, n = nrow(df)-1)

# Matrice VAT
fviz_dist(dist(df), show_labels = FALSE)
```

Test de Hopkins

Matrice VAT