Classification non supervisée

Points importants à considerer lors de la conception d'un classi

1 - Conception en 1 étape / en Hiérarchie

- Conception en Hiérarchie (Arbre binaire) : les éléments les plus distincts sont discriminés dans les noeuds supérieurs. (Mui & Fu, 1980)

2 - Méthode paramétrique / non-paramétrique

- Méthode paramétrique suppose connu la Forme de la fdp (en général Gaussienne) + Forme Unimodal
- Méthode non-paramétrique : Remise en cause de la loi de fdp et de l'aspect Unimodal ⇒ Recherche à estimer la fdp.

3 - Dimensionalité / Taille échantillon d'apprentissage

- il est conseillé statistiquement de disposer d'1 échantillon d'apprentissage par classe d'une Taille d'au moins 5 à 10 fois le nombre de descripteurs

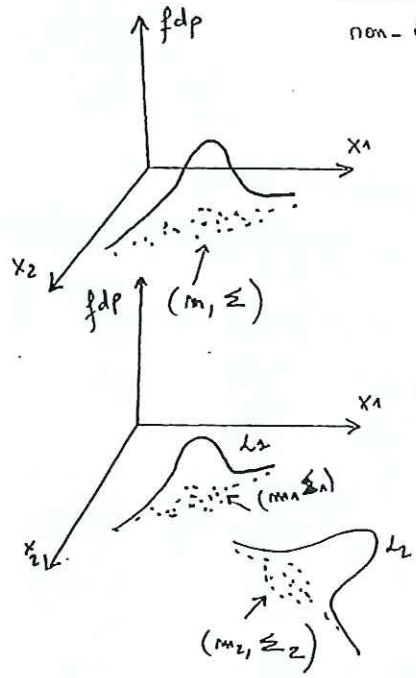
4 - Selection de Variable (descripteurs)

- Méthode standard Non-Onéreuse "Branch & Bound" (Narendra & FUKUNAGA (1977))

5 - Problème du partitionnement de l'échantillon global en un échantillon d'apprentissage et de Tests assurant la conception d'un Classifier performant. (Devijver & Kittler (1982), Toussaint (1974))

La classification.

1) Problématique 1 : Connaître par apprentissage la structure multidimensionnelle d'1 ensemble d'objets (non-étiquetés)

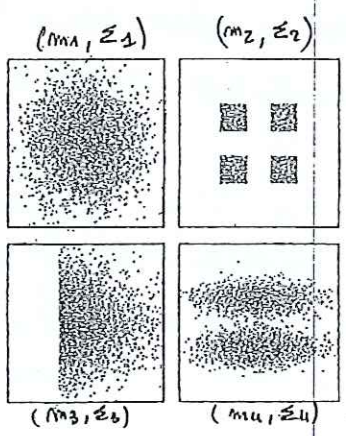


L'échantillon se distribue selon une seule loi Normale de paramètres (m, Σ) à estimer.

Méthodes paramétriques : estimer (m(m1, ..., mp), Σ(Σ1, ...

Méthodes non-paramétriques : estimer la f.dp mixte

⚠ Les classes correspondent aux différents modes de la fdp.



avec : $m_1 = m_2 = m_3 = m_4$

$\Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4$

⚠ Rappel : m indicateur des zones de densité forte

Σ indicateur de la dispersion du nuage selon différentes dire

2) Problématique 2 : Extraire les groupements naturels d'objets, en indiquant pour chaque objets sa classe d'appartenance

2.1 Mesures de similarité :

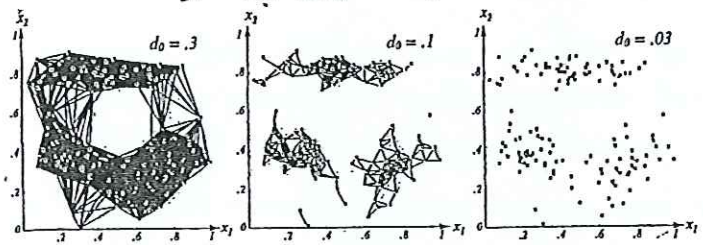
- Choix d'1 mesure de similarité adéquate (rapprochant les objets appartenant à une même classe et éloignant les objets de classes différentes)

Classification selon le principe de distance Minimale

$O_i, O_j \in$ à une classe C ssi $d(O_i, O_j) \leq d_0$

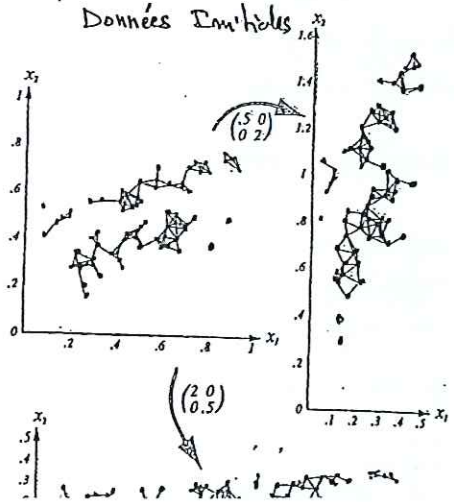
$d_0 >>>$ $\Rightarrow$ Nombre de classe extraite $\rightsquigarrow 1$

si $d_0 <<<$ $\Rightarrow$ Nombre de classe extraite $\rightsquigarrow N$ (Taille Echantillon)



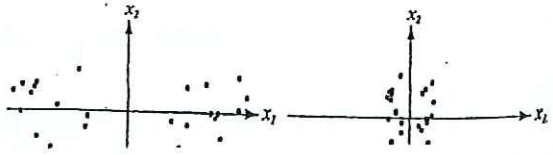
Choix stratégique de d_0 : distance Intra classe $\leq d_0 \leq$ distance Inter classes

Données Entrées



Effets du changement d'échelle et de la normalisation sur la classification.

Normalisation à $(m=0, \sigma=1)$



3) Les fonctions critères en classification.

- Soit $C = \{x_1, \ldots, x_N\}$
- objectif : partitionner l'ensemble d'objets en q classes disjointes $C_1, \ldots C_q$
- à chaque partition $C_1, \ldots, C_q$ on associe une mesure de qualité définie par la fonction critère.
- La problématique de partitionnement revient à chercher la partition $C_1^i, \ldots C_q^i$ qui optimise la fonction critère.

3.1 Critère de la somme des carrés des erreurs (SCE)

On note n_i : nb d'objets de la classe C_i
m_i : la moyenne des objets (centre de gravité, meilleur représentant ...)

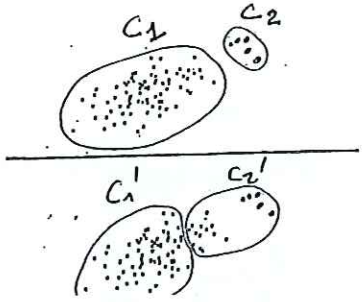
$$m_i = \frac{1}{n_i} \sum_{x \in C_i} x$$; On note J_e : le critère SCE à Minimiser

$$J_e = \sum_{i=1}^{q} \sum_{x \in C_i} \| x_i - m_i \|^2$$

$\Rightarrow$ c'est une classification à variance (Intra) Minimale.

Conditions d'applicabilité du critère J_e :

- des classes compactes et bien séparées



Partitions : $P_1 = \{C_1, C_2\}$; $P_2 = \{C_1', C_2'\}$

avec $J_e(P_1) > J_e(P_2)$

J_e pénalise les groupements naturels de grande taille ou de grande variabilité

Rq: Généralisation de Jc à une mesure de dissimilarité d.

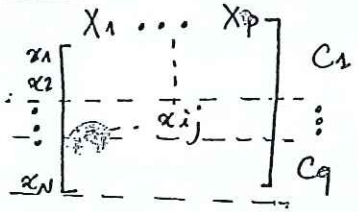
$$J_c = \sum_{i=1}^{q} \sum_{x \in C_i} \|x_i - m_i\|^2 = \frac{1}{2} \sum_{i=1}^{q} n_i s_i$$

où $$S_i = \frac{1}{n_i^2} \sum_{x \in C_i} \sum_{x' \in C_i} \|x - x'\|^2$$ (cas particulier de distance euclidienne)

On peut généraliser S_i à une mesure de dissimilarité d :

$$S_i = \frac{1}{n_i^2} \sum_{x \in C_i} \sum_{x' \in C_i} d(x, x')$$

1.2 Critère fondé sur la matrice des Variance-Covariance



Définitions

Vecteur Moyen de la classe C_i	$m_i = \frac{1}{n_i} \sum_{x \in C_i} x$
Vecteur Moyen Global	$m = \frac{1}{N} \sum_{C} x = \frac{1}{N} \sum_{i=1}^{q} n_i m_i$
Matrice de Var-Cov de la classe C_i $(p \times p)$	$V_i = \sum_{x \in C_i} (x - m_i)(x - m_i)^T$
Matrice de Var-Cov Intra Classes $(p \times p)$	$V_{Intra} = \sum_{i=1}^{q} V_i$
Matrice Var-Cov Inter classes $(p \times p)$	$V_{Inter} = \sum_{i=1}^{q} n_i (m_i - m)(m_i - m)^T$
Matrice de Var-Cov Globale $(p \times p)$	$V_{Tot} = \sum_{x \in C} (x - m)(x - m)^T$ $= V_{Intra} + V_{Inter}$

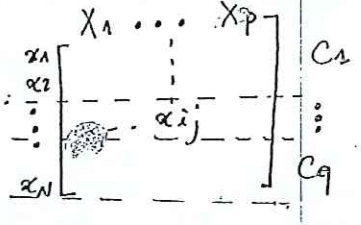
Jc = $\sum_{i=1}^{q} \sum_{x \in C_i} \|x_i - m_i\|^2 = \frac{1}{2} \sum_{i=1}^{q} n_i s_i$

où $S_i = \frac{1}{n_i^2} \sum_{x \in C_i} \sum_{x' \in C_i} \|x - x'\|^2$ (cas particulier de distance euclidienne)

On peut généraliser S_i à une mesure de dissimilarité d :

$S_i = \frac{1}{n_i^2} \sum_{x \in C_i} \sum_{x' \in C_i} d(x, x')$

1.2 Critère fondé sur la matrice des Variance-Covariance



Définitions

Vecteur Moyen de la classe C_i	$m_i = \frac{1}{n_i} \sum_{x \in C_i} x$
Vecteur Moyen Global	$m = \frac{1}{N} \sum_{C} x = \frac{1}{N} \sum_{i=1}^{q} n_i m_i$
Matrice de Var-Cov de la classe C_i $(p \times p)$	$V_i = \sum_{x \in C_i} (x - m_i)(x - m_i)^T$
Matrice de Var-Cov Intra Classes $(p \times p)$	$V_{Intra} = \sum_{i=1}^{q} V_i$
Matrice Var-Cov Inter classes $(p \times p)$	$V_{Inter} = \sum_{i=1}^{q} n_i (m_i - m)(m_i - m)^T$
Matrice de Var-Cov Globale $(p \times p)$	$V_{Tot} = \sum_{x \in C} (x - m)(x - m)^T$ $= V_{Intra} + V_{Inter}$

La meilleur partition $C_1, \dots C_q$ est celle qui

- Minimise la variance Intra V_{Intra}.

Comme V_{TOT} est indépendant de la partition :

Minimiser V_{Intra} $\Rightarrow$ Maximiser V_{Inter}.

Les deux mesures scalaires utilisés :

Trace , Determinant

Trace

Minimiser V_{Intra} $\Rightarrow$ $Min\left(Tr\left(V_{Intra}\right)\right)$

Rq: $Tr(V_{Intra}) = J_e$

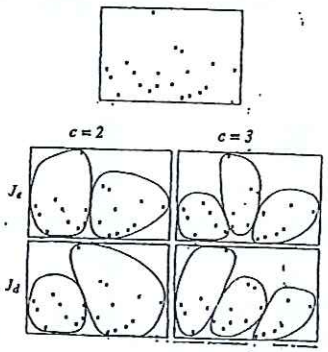
Déterminant

Minimiser V_{Intra} $\Rightarrow$ $\boxed{Min\left(|V_{Intra}|\right) = J_d}$

Résumé :

J_e favorise la construction de partitions à classes de tailles équivalentes

J_d favorise la construction de partitions à classes de tailles très différentes



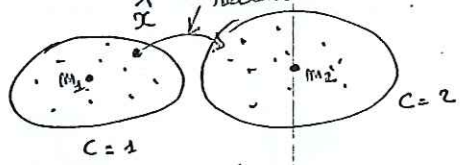
Algorithmes classifications en apprentissage non supervisé

1/ K-Means

1 begin initialisation n, c, $m_1, m_2, \dots, m_c$ { c : nb de cl à extr
n : Taille vol élément
m_i : Centre d la classe
2 Do affecter les n objets aux centres m_i les plus proches
3 recalculer les centres m_i
4 until plus de changement pour m_i
5 Return $m_1, m_2, \dots, m_c$
6 End

2/ Algorithme de classification itérative fondée sur le critère J_e (IC

Rappel : $J_e = \sum_{i=1}^{c} J_i$; $J_i = \sum_{x_i \in D_i} \|x - m_i\|$



$m_1 = \frac{1}{n_1} \sum_{x \in D_1} x$

$J_1 = \sum_{x \in D_1} \|x - m_1\|^2$

$m_2 = \frac{1}{n_2} \sum_{x \in$

$J_2 = \sum_{x \in D_2} \|x$

le reclassement de $\hat{x}$ dans C_2 induit :

$$\boxed{m_2^* = m_2 + \frac{\hat{x} - m_2}{n_2 + 1}}$$

et une augmentation de J_2 :

$$\boxed{J_2^* = J_2 + \frac{n_2}{n_2 + 1} \|\hat{x} - m_2\|^2}$$

De manière similaire, dans C_1 :

$$\boxed{m_1^* = m_1 - \frac{\hat{x} - m_1}{n_1 - 1}}$$

et une diminution de J_1 :

$$\boxed{J_1^* = J_1 - \frac{n_1}{n_1 - 1} \|\hat{x} - m_1\|^2}$$

Ci vers la classe C_j ssi :

$$\frac{n_i}{n_i - 1} \| \vec{x} - m_i \|^2 > \frac{n_j}{n_j + 1} \| \vec{x} - m_j \|^2$$

Algorithme

1 begin initialisation $n, c, m_1, \dots, m_c$
2 do selectionner $\hat{x}$ aleatoirement
3 $i \leftarrow \arg\min_{i'} \| m_{i'} - \hat{x} \|$ { affecter $\hat{x}$ à la classe la plus proche }
4 if $(n_i \neq 1)$ Then
5 $$\rho_j = \begin{cases} \frac{n_j}{n_j+1} \| \hat{x} - m_j \|^2 & (j \neq i) \\ \frac{n_j}{n_j-1} \| \hat{x} - m_i \|^2 & (j = i) \end{cases}$$
6 if $\rho_k \leq \rho_j$ $(\forall j)$ then transfer $\hat{x}$ dans D_i
7 recalculer m_i, m_k, J_e
8 UNTIL J_e demeure constant durant n itérations
9 Return $m_1, \dots, m_c$
10 end

3/ Comparaison K-MEANS et ICMSE

K-MEANS

- Mise à jour des centres après toutes les affectations
- indépendance de l'ordre de Traitement des objets

ICMSE

- Mise à jour des centres après chaque affectation
- les résultats dépendent de l'ordre de traitements des objets
- classement optimal à chaque itération (risque d'être pris ds un minimum local)
- plus approprié pour effectuer une classific on-line (cas où les objets se présentent séquentiellement).

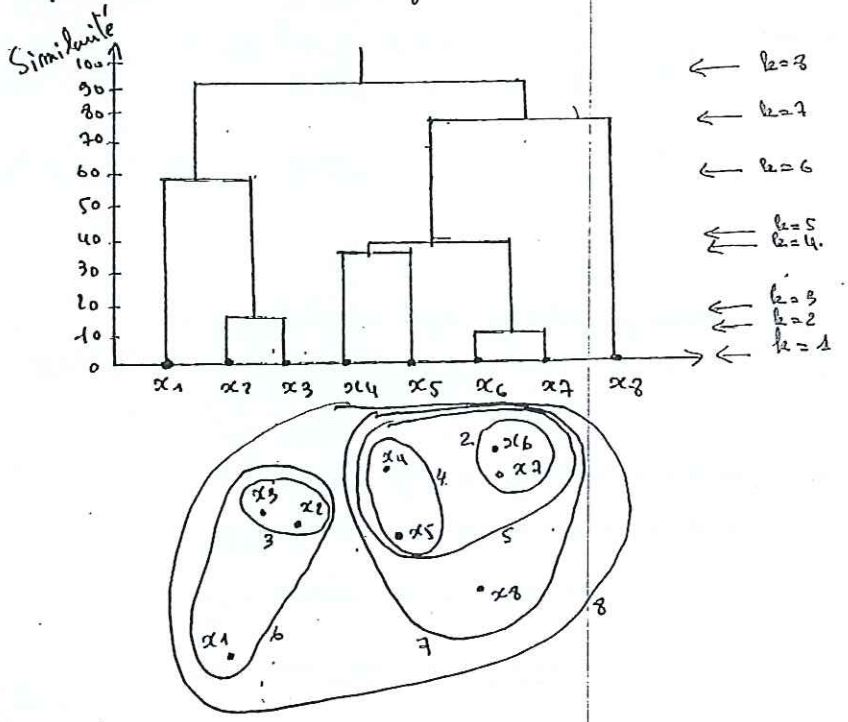
4/ Classification Hiérarchique

- considérant n objets à classer en c classes.

la première étape consiste à partitionner l'ensemble des objets en n classes (singleton). Soit P_1 cette partition.
la deuxième étape est de construire une partition (à partir de P_1) à $n-1$ classes. Soit P_2 la partition obtenue.
⋮
à l'étape k on construit une partition P_k à $n-k+1$

Propriété : soit x_1, x_2 deux objets :
si $x_1, x_2 \in P_i \Rightarrow x_1, x_2 \in P_j \quad \forall j > i$

Représentation d'1 classification Hiérarchique



Plan

- Introduction & Notations
- Principales étapes de construction d'un arbre binaire
- Définition des divisions admissibles
- Définition du critère de sélection de la meilleure division.
- Règle d'affectation
- Estimation du risque d'erreur

1 Début initialisation $C \leftarrow n$, $D_i = \{x_i\}$; $i = 1 \dots n$
2 do $C \leftarrow C - 1$
3 Trouver les deux classes les plus proches, soit D_i et D_j
4 Regrouper D_i et D_j
5 until $C = S$
6 Retourn C classes
7 end

Pour évaluer la proximité entre deux classes D_i, D_j on choisit une des mesures suivantes :

$$d_{min}(D_i, D_j) = \min_{\substack{x \in D_i \\ x' \in D_j}} \|x - x'\|$$

$$d_{max}(D_i, D_j) = \max_{\substack{x \in D_i \\ x' \in D_j}} \|x - x'\|$$

$$d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{x' \in D_j} \|x - x'\|$$

$$d_{mean}(D_i, D_j) = \|m_i - m_j\|$$

XIII. Algorithme de classification Hiérarchique

1 begin initialisation $c, \hat{c} \leftarrow n$, $D_i = \{x_i\}$; $i = 1 \dots n$
2 do $\hat{c} \leftarrow \hat{c} - 1$
3 Trouver les deux classes les plus proches, soit D
4 fusionner D_i et D_j
5 until $c = \hat{c}$
6 return c classes
7 end

pour évaluer la proximité entre deux classes D_i, D_j on choisit une des mesures suivantes :

$$d_{min}(D_i, D_j) = \min_{\substack{x \in D_i \\ x' \in D_j}} \|x - x'\|$$

$$d_{max}(D_i, D_j) = \max_{\substack{x \in D_i \\ x' \in D_j}} \|x - x'\|$$

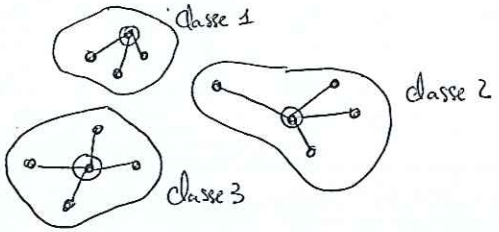
$$d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{x' \in D_j} \|x - x'\|$$

$$d_{mean}(D_i, D_j) = \|m_i - m_j\|$$

Partitioning Around Medoids (P.A.M)

Idée :

- choisir k objets représentatifs,
- affecter les objets restants aux k objets représentatifs



Problème : Comment choisir les bons objets représentatifs

Un bon objet représentatif doit minimiser la moyenne des distances avec les autres objets de la classe.

Number	x Coordinate	y Coordinate
1	1.0	4.0
2	5.0	1.0
3	5.0	2.0
4	5.0	4.0
5	10.0	4.0
6	25.0	4.0
7	25.0	6.0
8	25.0	7.0
9	25.0	8.0
10	29.0	7.0

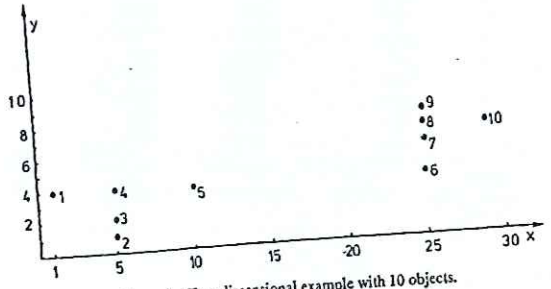


Figure 1 Two-dimensional example with 10 objects.

1er cas : supposons le choix des objets 1 et 5 comme objets représentatifs

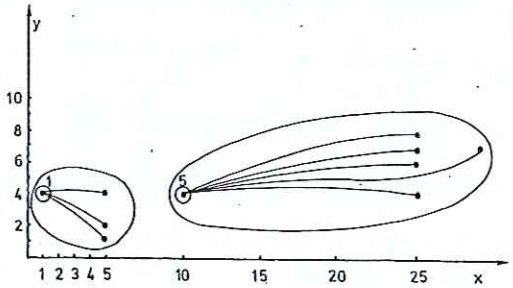
Notation : 1 et 5 dits "Medoids".

- Calculons la dissimilarité entre tous les objets et les medoïc 1 et 5 (Table 2)

Object Number	Dissimilarity from Object 1	Dissimilarity from Object 5	Minimal Dissimilarity	Closest Representative Object
1	0.00	9.00	0.00	1
2	5.00	5.83	5.00	1
3	4.47	5.39	4.47	1
4	4.00	5.00	4.00	1
5	9.00	0.00	0.00	5
6	24.00	15.00	15.00	5
7	24.08	15.13	15.13	5
8	24.19	15.30	15.30	5
9	24.33	15.52	15.52	5
10	28.16	19.24	19.24	5
			Average 9.37	

- La moyenne de la distance des objets aux Medoids est de 9,37 (critère de qualité de la partition retenue)

- La partition obtenue

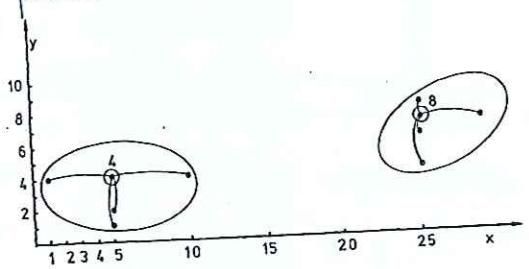


2ème cas : Supposons le choix des medoids 4 et 8

- Calculons la dissimilarité (distance euclidienne) entre tous les objets et les medoids 4, 8 (Table 3)

Object Number	Dissimilarity from Object 4	Dissimilarity from Object 8	Minimal Dissimilarity	Closest Representative Object
1	4.00	24.19	4.00	4
2	3.00	20.88	3.00	4
3	2.00	20.62	2.00	4
4	0.00	20.22	0.00	4
5	5.00	15.30	5.00	4
6	20.00	3.00	3.00	8
7	20.10	1.00	1.00	8
8	20.22	0.00	0.00	8
9	20.40	1.00	1.00	8
10	24.19	4.00	4.00	8
			Average 2.30	

- La qualité de la partition est évaluée à 2.30.
- La partition obtenue



- La partition correspondant aux medoids (4,8) est meilleur que celle associée aux medoids (1,5)

$2.30 < 9.37$

Pourquoi choisir PAM

- plus stable que K-MEANS face aux outliers.
- permet l'extraction des représentants des classes.
  → utile pour la réduction de la dimension des données
  → utile pour généraliser la description d'1 classe (description en intention de la classe)
- elle fournit des statistiques permettant d'apprécier la position des objets dans les classes, le voisinage des classes, la qualité des classes (ie compacité), la séparabilité des classes. Un graphique "silhouette" également produit illustrant la partition obtenue

Description de l'algorithme PAM

L'algorithme PAM consiste en deux phases :

- Une phase dite "BUILD" qui permet le choix de k medoids et l'obtention d'une première partition.
- Une seconde phase dite "SWAP" qui cherche à améliorer le choix fait dans la phase 1 des k-medoids, ainsi améliorer la qualité de la partition.