L'algorithme "BUILD" { choisir les k objets représentatifs } (5)

- Choisir le 1er objet représentatif comme celui minimisant la somme des distances aux objets restants (l'objet le plus central)
- choisir le second objet représentatif selon les étapes suivantes :

1. Considérer un objet i qui n'a pas été sélectionné
2. Considérer un objet j " " " " "
   - Calculer la distance entre j et l'objet représentatif le plus proche, soit D_j cette distance.
   - Calculer la distance entre j et i. Soit $d(j,i)$ cette distance.
   - Calculer la différence $D_j - d(j,i)$

   { si cette différence est positive, alors j va contribuer à la sélection de i }
3. Calculer la contribution de j à la sélection de i, notée C_{ji} :

$$C_{ji} = \max(D_j - d(j,i), 0)$$

4. Calculer le gain total obtenu pour la sélection de i : $\sum_j C_{ji}$
5. Choisir l'objet i comme le prochain objet représentatif tel que il vérifie

$$\text{maximise}_i \left(\sum_j C_{ji}\right)$$

des étapes 1, 2, 3, 4, 5 ... du k ième objet représentatif.

L'algorithme "SWAP". { améliorer le choix des k medoids sélectionnés par BUILD }

idée principale : consiste à regarder toutes les paires d'objets (i, h) où i est un medoid et h non, et évaluer le gain au niveau de la partition si l'on substitue i par h, cad si i est non medoid et h l'est.

L'évaluation de l'effet de la substitution de i par h sur la qualité de la partition est effectuée en 2 étapes :

1. Considérer un objet non-sélectionné j et calculer sa contribution C_{jih} à la substitution de i par h :

1.1 Si $D_j < d(j,i)$ et $D_j < d(j,h)$
alors $C_{jih} = 0$

1.2 Si $D_j = d(j,i)$ { j est plus proche de i que de tout autre medoid }
alors
  Si $d(j,h) < E_j$ { E_j : distance entre j et le second Medoid le + proche }
  alors $C_{jih} = d(j,h) - d(j,i)$
  Sinon. $C_{jih} = E_j - D_j$
finsi

1.3 Si $d(j,i) > D_j$ et $D_j > d(j,h)$
alors $C_{jih} = d(j,h) - D_j$

2. Calculer le total des contributions pour la substitution de i par h :

$$T_{ih} = \sum_j C_{jih}.$$

3. Sélectionner la paire (i,h) à substituer tel que :

$$\underset{(i,h)}{\text{minimise}}\ (T_{ih})$$

4. Si La valeur Min de T_{ih} est négative
   alors procéder au "swap" de i par h.
   Sinon Arrêt de l'algorithme.

Exercice PAM :

- Utiliser l'algorithme PAM pour extraire une Bi-partition à partir de la matrice de dissimilarités suivante

	A	B	C	D	E	F
A	0					
B	5	0				
C	11	10	0			
D	11	6	6	0		
E	14	13	17	13	0	
F	14	15	21	15	6	0

BUILD

D 1er Medoid.

i	$\sum_j d_{ij}$	A : $d_{\cdot A}$	A : $d(\cdot,D)$	A : $C_{\cdot A}$	B : $d_{\cdot B}$	B : $d(\cdot,D)$	B : $C_{\cdot B}$
A	55				$d_{AB}=5$	$d_{AD}=11$	6
B	49	$d_{BA}=5$	$D_B = d_{BD}=6$	$C_{BA}=1$			
C	65	$d_{CA}=11$	$D_C=6$	$C_{CA}=0$	$d_{CB}=10$	$d_{CD}=6$	0
D	(51)						
E	63	$d_{EA}=14$	$D_E = 13 = d_{ED}$	$C_{EA}=0$	$d_{EB}=13$	$d_{ED}=13$	0
F	71	$d_{FA}=14$	$d_{FD}=15$	$C_{FA}=1$	$d_{FB}=14$	$d_{FD}=15$	1
			Total contrib	$C_A=+2$			$C_B=7$

	C : $d_{\cdot C}$	C : $d(\cdot,D)$	C : $C_{\cdot C}$	E : $d_{\cdot E}$	E : $d(\cdot,D)$	E : $C_{\cdot E}$	F : $d_{\cdot F}$	F : $d(\cdot,D)$	F : $C_{\cdot F}$
A	11	11	0	14	11	0	14	11	0
B	10	6	0	13	6	0	15	6	0
C				17	6	0	21	6	0
D									
E	17	13	0				6	15	9
F	21	15	0	6	15	9			
			0			9			9

2ème Medoid. c'est E ou F.

La partition correspondante (initiale) associé à D, E

$$P_1 = \{D, A, B, C\}$$
$$C_2 = \{E, F\}$$

SWAP

Les caractéristiques d'une classe:

- si c'est un singleton.
- L'isolation: Le degré d'isolation d'une classe s'évalue par deux critère L-cluster et L^*-cluster (introduit par Gordon 1981)

C est dit L-cluster si $\forall$ l'objet $i \in C$ on a:

$$\max_{j \in C} (d(i,j)) < \min_{h \notin C} (d(i,h))$$

C est dit L^*-cluster ssi:

$$\max_{i,j \in C} d(i,j) < \min_{l \in C,\ h \notin C} d(l,h)$$

si C est L^*-cluster $\Longrightarrow$ C est L-cluster

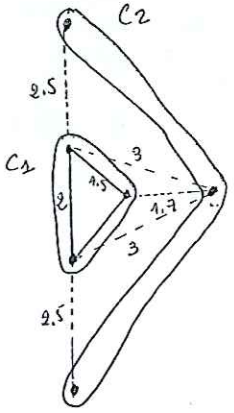
- diamètre de la classe:

$$\text{diamètre}(c) = \max_{i,j \in C} (d(i,j))$$

- séparation de C

$$\text{Sép}(c) = \min_{l \in C \text{ et } h \notin C} d(l,h)$$

exp:



C_1 est un L-cluster mais pas un L^*-cluster car $2 \not< 1.7$

L^* cluster : condition plus stricte.

- Enfin pour chaque cluster on peut mesurer la moyenne des objets au medoid et la distance Max entre objet et Medoid. (sorte de variance / ecart à la Moyenne).

Visualisation de la partition obtenue. (Silhouette graph.)

- chaque classe de la partition est représentée par une silhouette indiquant pour chaque objet de la classe s'il est bien situé dans la classe (central) ou se situe en bordure de classe.

La construction du graph est fondée sur le calcul de la valeur dite "silhouette" pour chaque objet i notée s(i).

Calcul de la valeur silhouette s(i):

- On note A la classe d'appartenance de l'objet i,
- a(i): dissimilarité moyenne entre i et les autres objets de A. (A est non singleton)
- d(i, C) = dissimilarité moyenne entre i et tous les objets d'une classe $C \neq A$.
- $b(i) = \min_{C \neq A} (d(i,C))$

{ b(i): dissimilarité Moyen entre i et les objet de la classe la plus proche }.

On note la classe la plus proche B.

- On note B comme classe voisine de i
- On définit s(i):

$$s(i) = 1 - \frac{a(i)}{b(i)} \quad \text{si } a(i) < b(i)$$

$$= 0 \quad \text{si } a(i) = b(i)$$

$$= \frac{b(i)}{a(i)} - 1 \quad \text{si } a(i) > b(i)$$

Autre formule de s(i) plus compact:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \qquad -1 \leq s(i) \leq 1$$

Si $s(i) \simeq 1 \Rightarrow$ l'objet i est bien classé

Si $s(i) \simeq -1 \Rightarrow$ l'objet i est mal classé
il est plus utile de l'affecté à B plutôt qu'à A. en modifiant son $s(i)$ par $-s(i)$.

- pour chaque classe on peut calculer sa valeur Silhouette moyenne

$$s(A) = \frac{1}{|A|} \sum_{i} s(i)$$

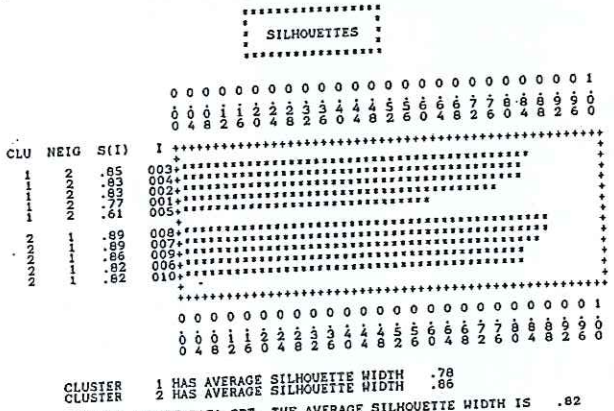
- pour toute la partition on évalue sa Silhouette Moyenne

$$s(P) = \sum_{C} \frac{|c|}{|P|} s(C)$$

$|c|$: cardinal de C
$|P|$: Taille de l'ensemble de données

La silhouette associée à la partition des données (Table 1)

SILHOUETTES

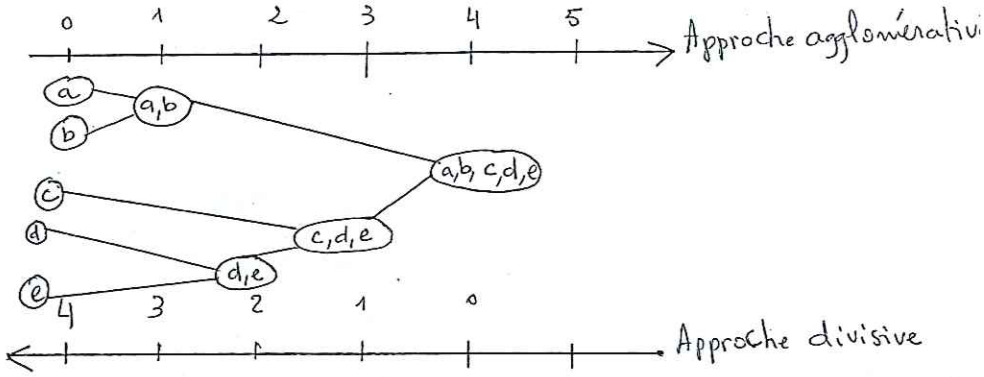


CLUSTER 1 HAS AVERAGE SILHOUETTE WIDTH .78
CLUSTER 2 HAS AVERAGE SILHOUETTE WIDTH .86

FOR THE ENTIRE DATA SET, THE AVERAGE SILHOUETTE WIDTH IS .82

SC	Proposed Interpretation
0.71–1.00	A strong structure has been found
0.51–0.70	A reasonable structure has been found
0.26–0.50	The structure is weak and could be artificial; please try additional methods on this data set
≤ 0.25	No substantial structure has been found

L'Analyse Divisive



Rappel : L'approche agglomérative considère initialement l'ens des individus ; puis aggrège progressivement les individus jusqu'à l'obtention d'1 seule classe.

L'approche divisive considère initialement 1 classe regroupant l'ensemble des n individus ; puis procède à la partition du groupe en 2 sous-groupes ; et ceci jusqu'à l'obtention de n classes singletons.

Inconvenient principal de l'approche agglomérative c'est de commencer par les descriptions "detail" ; une fois un groupe constitué ; il est impossible de le scinder dans les étapes suivantes.
A l'inverse ; l'analyse divisive considère initialement l'information dans sa globalité.

1ère approche divisive

- à partir de l'ensemble des n individus, étudier toutes les bi-partitions possibles.

$\Rightarrow$ problème calculatoire $2^{n-1} - 1$ possibilité

TROP COUTEUX !!

2ème approche divisive (Macnaughton-Smith et al. (1964))

L'idée de base :

- L'individu le plus éloigné du groupe le quitte pour en constituer un autre;
- à l'étape suivant, les individus du groupe initial vont soit rester ou suivre l'individu sortant.
- Les deux étapes se répètent jusqu'à l'atteinte d'un point d'équilibre.

Exemple illustratif :

Considérons la matrice de dissimilarités entre les individus $\{a, b, c, d, e\}$:

1ère étape :

	a	b	c	d	e
a	0.0	2.0	6.0	10.0	9.0
b	2.0	0.0	5.0	9.0	8.0
c	6.0	5.0	0.0	4.0	5.0
d	10.0	9.0	4.0	0.0	3.0
e	9.0	8.0	5.0	3.0	0.0

Object	Average Dissimilarity to the Other Objects
a	(2.0 + 6.0 + 10.0 + 9.0)/4 = 6.75
b	(2.0 + 5.0 + 9.0 + 8.0)/4 = 6.00
c	(6.0 + 5.0 + 4.0 + 5.0)/4 = 5.00
d	(10.0 + 9.0 + 4.0 + 3.0)/4 = 6.50
e	(9.0 + 8.0 + 5.0 + 3.0)/4 = 6.25

Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
b	(5.0 + 9.0 + 8.0)/3 ≈ 7.33	2.00	5.33
c	(5.0 + 4.0 + 5.0)/3 ≈ 4.67	6.00	−1.33
d	(9.0 + 4.0 + 3.0)/3 ≈ 5.33	10.00	−4.67
e	(8.0 + 5.0 + 3.0)/3 ≈ 5.33	9.00	−3.67

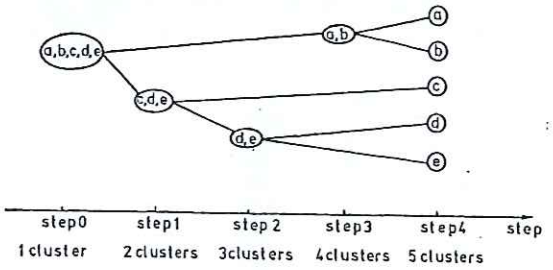
Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
c	(4.0 + 5.0)/2 = 4.50	(6.0 + 5.0)/2 = 5.50	−1.00
d	(4.0 + 3.0)/2 = 3.50	(10.0 + 9.0)/2 = 9.50	−6.00
e	(5.0 + 3.0)/2 = 4.00	(9.0 + 8.0)/2 = 8.50	−4.50

	c	d	e
c	0.0	4.0	5.0
d	4.0	0.0	3.0
e	5.0	3.0	0.0

Object	Average Dissimilarity to the Other Objects
c	(4.0 + 5.0)/2 = 4.50
d	(4.0 + 3.0)/2 = 3.50
e	(5.0 + 3.0)/2 = 4.00

	d	e
d	0.0	3.0
e	3.0	0.0

Object	Average Dissimilarity to the Other Objects
d	3.00
e	3.00



Algorithme de l'analyse divisive

Notations : R : l'ensemble à partitionner

A , B : les deux groupes issus de la partition de l

Divisive (R)

Initialisation : A = R et B = ∅

Si $|A| \geq 1$ alors

1. calculer pour chaque individu i de A

$$d(i, A\setminus\{i\}) = \frac{1}{|A|-1} \sum_{\substack{j\in A \\ j\neq i}} d(i,j)$$

soit $i' = \max_i \left(d(i, A\setminus\{i\}) \right)$

2. Déplacer i' de A vers B.

$A_{New} = A_{old} \setminus \{i'\}$

$B_{New} = B_{old} \cup \{i'\}$

3. Répéter le déplacement d'individus de A vers B tant que il y a des individus plus proche de B que de A :

Calculer pour chaque i dans A

$$(*)\ d(i, A\setminus\{i\}) - d(i,B) = \frac{1}{|A|-1} \sum_{\substack{j\in A \\ j\neq i}} d(i,j) - \frac{1}{|B|} \sum_{h\in B} d(i,h)$$

Soit $i'' = \max_i \left(d(i, A\setminus\{i\}) - d(i,B) \right)$

4. Si $d(i'', A\setminus\{i''\}) - d(i'',B) > 0$ alors

- déplacer i'' de A vers B.
- répéter 2. et 3.

Sinon aller à l'étape 5.

5. Réitérer l'algorithme pour pour le groupe au diamètre le plus large : diam (A) = max (d(j,h))

Classification Monothétique :

	Variables					
Objects	1	2	3	4	5	6
AAA	1	1	0	1	1	0
BBB	1	1	0	0	0	1
CCC	1	1	1	1	1	0
DDD	1	1	1	1	1	0
EEE	0	0	0	1	0	1
FFF	0	0	0	0	0	0
GGG	0	0	1	1	0	1
HHH	0	0	1	1	1	0

L'idée de base : - il s'agit de diviser l'ensemble des individus en choisissant 1 variable binaire.

- L'étape suivante consiste à choisir 1 variable binaire parmi les variables restantes et de même partitionner les 2 nouveaux sous-ensembles.
- cette procédure continue tant qu'il existe un sous-ensemble "divisible" ou non-singleton.

⚠ La principale question : c'est comment choisir la variable binaire de coupure ?

La variable choisie est celle maximisant la "similarité" avec les variables restantes (i.e. la variable la + central

Critère d'association entre deux variables binaires.

- soit x_f, x_g deux variables binaires.

$x_g \backslash x_f$	1	0
1	a_{fg}	b_{fg}
0	c_{fg}	d_{fg}

$Asso(x_f, x_g) = | a_{fg} \cdot d_{fg} - b_{fg} \cdot c_{fg}$

Considerons les deux exemples suivants:

Le tableau de contingence associé aux variable 1 et 3

3 \ 1	1	0	
1	2	2	
0	2	2	

$Ass(1,3) = |4-4| = 0$

Le tableau de contingence associé à deux variables totalement opposées:

y \ x	1	0	
1	0	4	
0	4	0	

$Ass(x,y) = |0-16| = 16.$

$Ass(x,y) > Ass(1,3).$

La mesure d'association considéré ne mesure pas une similarité "classique" entre les deux variable mais une proximité entre les deux partitions issues de ces deux variables.

Remarques:

- Cette mesure d'association suppose que les variables binaires sont symétriques. (0 et 1 jouent le même rôle)
- Sensible aux variables binaires non-equilibrées (beaucoup de 0 - peu de 1 ou inversement) à savoir donne des valeurs d'association faibles.

Algorithme de la classification Monothetique.

1. Pour chaque variable f calculer son association avec les autres variables:

$$A_f = \sum_{g \neq f} A_{fg}$$

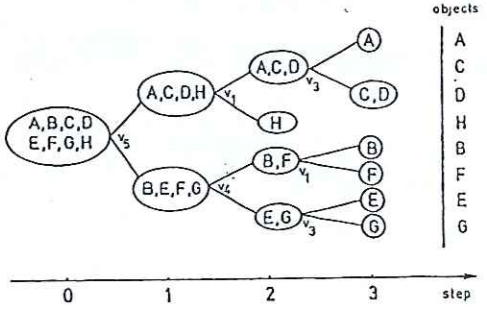
avec $A_{fg} = a_{fg} \cdot d_{fg} - b_{fg} \cdot c_{fg}$

soit t la variable : $A_t = \max_f (A_f)$

2. Partitionner l'ensemble des n individus en 2 classes selon la variable A_t (tous les individus du 1er groupe ont $A_t = 0$; tous ceux du deuxième groupe ont $A_t = 1$)

3. Réitère la séparation des groupes descendants en utilisant les variables restantes.

4. Arrêt si il n'y a plus de groupes à séparer (tous les singleton) ou qu'il n'existe plus de variables séparant les groupes.



Classification pour grands corpus

(Clustering Large Applications CLARA)

- L'algorithme PAM est non applicable sur des grands volume de données,
- Clara est proposée par (Kaufman et Rousseeuw 1986) pour partitionner des grandes masses de données,

CLARA consiste en 2 principales étapes:

1 - Un échantillon est extrait à partir de l'ensemble de données ; puis partitionné par un algorithme PAM en k classes.

2 - Chaque objet du jeu de données restant est affecté à la classe du medoid le plus proche.

3 - on évalue la qualité de la partition obtenue par la moyenne des dissimilarité entre les objets et leur medoid.

4 - On réitère les étapes 1, 2, 3 n fois, puis on retient la partition minimisant la dissimilarité moyenne (le critère de qualité le meilleur)

Algorithme CLARA

1. Sélectionner un échantillon de taille n à partir de l'ensemble total de taille N.

   avec $n = 40 + 2k$.

   k étant le nombre de classe à extraire.

   Pour chaque objet sélectionné aléatoirement vérifier qu'il n'appartient pas déjà à l'ensemble.

2. Si la taille de l'échantillon est trop petite $n < \frac{40+2k}{2}$ (dû aux objets doublons éliminés)

   alors on extrait de nouveaux objets et ceci tant que $n < \frac{40+2k}{2}$.

3. Si c'est le 1er échantillon à analyser.

   alors a - appliquer l'étape "BUILD" sur l'échantillon
   b - " " "SWAP" " "
   pour l'amélioration du choix des objets représentatifs.
   c - affecter les objets à l'échantillon aux Medoids le plus proche.
   e - évaluer la qualité de la partition.

   sinon:
   - Rajouter les médoids de la meilleur partition comme objets de l'échantillon ; puis
   - Compléter la sélection aléatoire d'autres objets
   - appliquer les étapes a, b, c, d, e.

4. Répéter les étapes 1, 2 et 3. P fois. Retenir la partition minimisant la moyenne des dissimilarités aux medoi