Université Mohamed Boudiaf – M'sila
Département Informatique
Master Intelligence Artificielle

2020-2021

Chapitre 4
Clustering

Dr. Mehenni Tahar

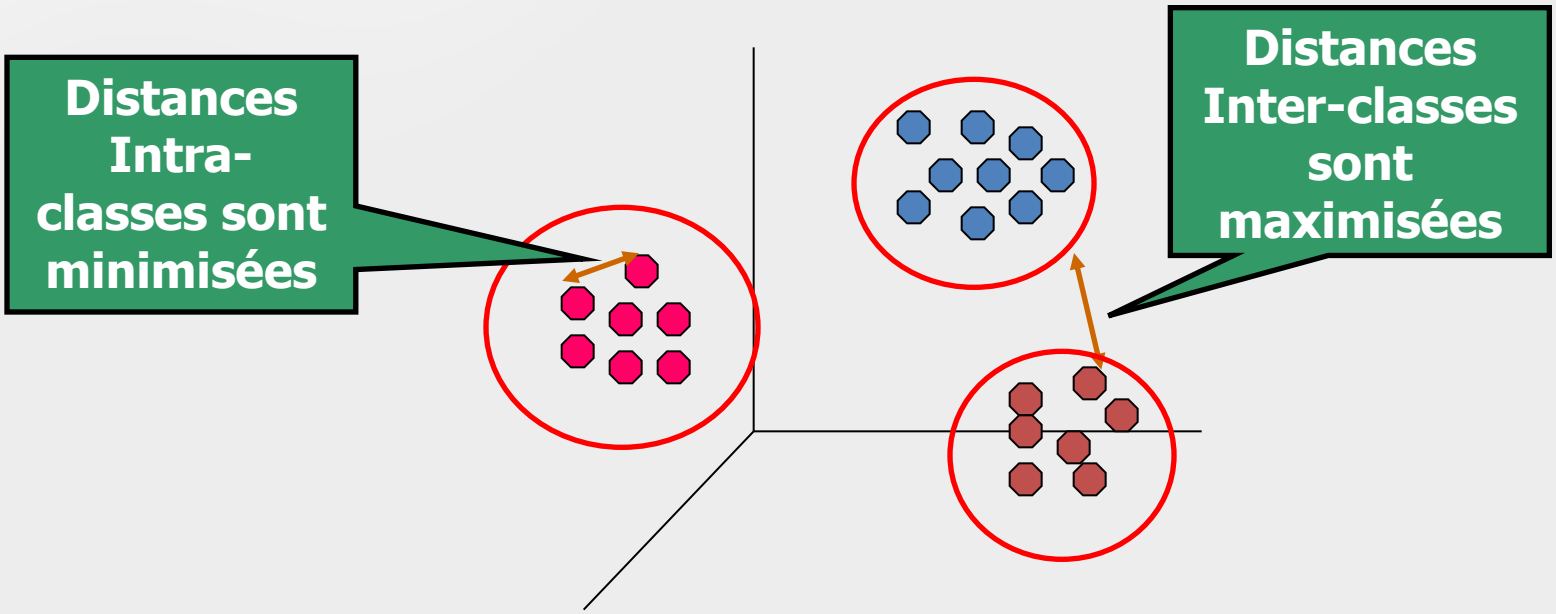
Clustering

- La Classification est un apprentissage supervisé. La supervision est faite en nommant les classes des instances d'apprentissage.
- Le Clustering est un apprentissage non supervisé. Il n'y a pas une connaissance apriori des classes, ni un ensemble d'apprentissage.
- L'algorithme de clustering nécessite une affectation de chaque instance à un groupe ou classe (cluster) de telle façon que tous les objets d'un même groupe sont plus semblables que les autres.

Clustering

- **Trouver des groupes (classes) d'objets tels que chaque objet d'un groupe est similaire qu'un autre objet du même groupe et différent des autres objets des autres groupes**
- L'objectif est de trouver un groupement le plus naturel possible des instances.

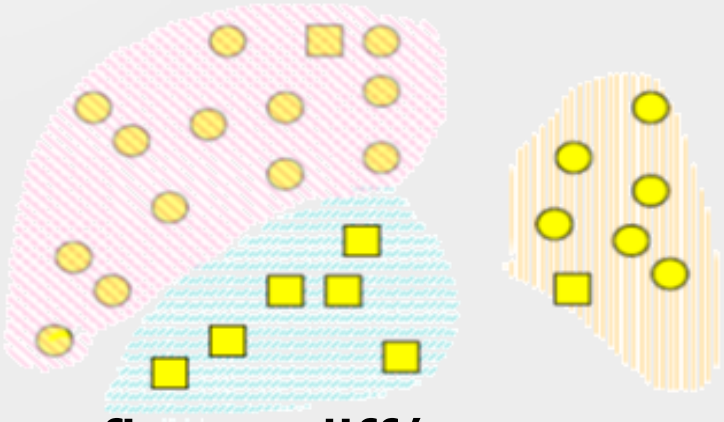
- A l'intérieur d'un groupe: Maximiser la similarité entre instances.
- Entre les groupes: Minimiser la similarité entre les instances.



Clustering

- **Par exemple, soit l'ensemble de figures suivant:**

- **Un algorithme de clustering peut trouver les clusters suivants:**



- **Bien que certaines figures différentes coexistent dans un cluster.**

Le problème du Clustering

- Etant donnée une base de données $D=\{t_1,t_2,\ldots,t_n\}$ de tuples et une valeur entière k, le *Clustering est de définir une application* $f:D\rightarrow\{1,..,k\}$ où chaque t_i est affecté à un seul cluster (groupe ou classe) K_j, $1<=j<=k$.

- Un *Cluster*, K_j, contient exactement les tuples qui lui sont affectés.

- Contrairement au problème de classification, les clusters ne sont pas connus apriori.

Qu'est ce qu'un bon regroupement?

- Une bonne méthode de regroupement permet de garantir
  - Une grande similarité intra-groupe
  - Une faible similarité inter-groupe
- La qualité d'un regroupement dépend donc de la mesure de similarité utilisée par la méthode et de son implémentation

Structures de données

- Matrice de données

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Matrice de similarité

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Types des variables

Attribute Type	Description	Examples	Operations
Nominal	The values of a nominal attribute are just different names, i.e., nominal attributes provide only enough information to distinguish one object from another. (=, ≠)	zip codes, employee ID numbers, eye color, sex: {*male*, *female*}	mode, entropy, contingency correlation, χ^2 test
Ordinal	The values of an ordinal attribute provide enough information to order objects. (<, >)	hardness of minerals, {*good*, *better*, *best*}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. (+, -)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, *t* and *F* tests
Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Mesurer la qualité d'un clustering

- Métrique pour la similarité: La similarité est exprimée par le biais d'une **mesure de distance**
- Une autre fonction est utilisée pour la mesure de la qualité
- Les définitions de distance sont très différentes que les variables soient des intervalles **(continues), catégories, booléennes ou ordinales**
- En pratique, on utilise souvent une pondération des variables

Similarité entre objets

- Les distances expriment une similarité
- Ex: la *distance de Minkowski* :

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)}$$

où $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ et $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ sont deux objets p-dimensionnels et q un entier positif

- Si $q = 1$, d est la distance de Manhattan

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

Similarité entre objets(I)

- *Si q = 2, d est la distance Euclidienne* :

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

  - Propriétés
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$

Variables binaires

- Une table de contingence pour données binaires

Objet *i* \ Objet *j*	1	0	*sum*
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

a= nombre de positions où i a 1 et j a 1

- Exemple o_i=(1,1,0,1, 0) et

o_j=(1,0,0,0,1)

a=1, b=2, c=1, d=1

Mesures de distances

- Coefficient d'appariement (matching) simple (invariant pour variables symétriques):

$$d(i, j) = \frac{b+c}{a+b+c+d}$$

Exemple o_i=(1,1,0,1,0) et o_j=(1,0,0,0,1)

$d(o_i, o_j)$=3/5

- Coefficient de Jaccard

$$d(i, j) = \frac{b+c}{a+b+c}$$

$d(o_i, o_j)$=3/4

Variables binaires (I)

- Variable symétrique: Ex. le sexe d'une personne, i.e coder masculin par 1 et féminin par 0 c'est pareil que le codage inverse

- Variable asymétrique: Ex. Test HIV. Le test peut être positif ou négatif (0 ou 1) mais il y a une valeur qui sera plus présente que l'autre. Généralement, on code par 1 la modalité la moins fréquente
  - 2 personnes ayant la valeur 1 pour le test sont *plus similaires* que 2 personnes ayant 0 pour le test

Variables binaires(II)

- Exemple

Nom	Sexe	Fièvre	Toux	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Sexe est un attribut symétrique
- Les autres attributs sont asymétriques
- Y et P ≡ 1, N ≡ 0, la distance n'est mesurée que sur les asymétriques

$$d(jack, mary) = \frac{0 + 1}{2 + 0 + 1} = 0.33$$

$$d(jack, jim) = \frac{1 + 1}{1 + 1 + 1} = 0.67$$

$$d(jim, mary) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

Les plus similaires sont Jack et Mary⇒atteints du même mal

Approches de Clustering

- Algorithmes de Partitionnement: Construire plusieurs partitions puis les évaluer selon certains critères
- Algorithmes hiérarchiques: Créer une décomposition hiérarchique des objets selon certains critères
- Algorithmes basés sur la densité: basés sur des notions de connectivité et de densité
- Algorithmes de grille: basés sur un structure à multi-niveaux de granularité
- Algorithmes à modèles: Un modèle est supposé pour chaque cluster ensuite vérifier chaque modèle sur chaque groupe pour choisir le meilleur

Algorithmes à partitionnement

- Construire une partition à ***k*** clusters d'une base ***D*** de ***n*** objets
- Les *k clusters doivent* optimiser le critère choisi
  - Global optimal: Considérer toutes les k-partitions
  - Heuristic methods: Algorithmes *k-means* et *k-medoids*
    - *k-means* (MacQueen'67): Chaque cluster est représenté par son centre
    - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Chaque cluster est représenté par un de ses objets

La méthode des k-moyennes (*K-Means)*

- L'algorithme *k-means* est en 4 étapes :

1. Choisir k objets formant ainsi k clusters
2. (Ré)affecter chaque objet O au cluster C_i de centre M_i tel que dist(O,M_i) est minimal
3. Recalculer M_i de chaque cluster (le barycentre)
4. Aller à l'étape 2 si on vient de faire une affectation

K-Means :Exemple

- A={1,2,3,6,7,8,13,15,17}. Créer 3 clusters à partir de A
- On prend 3 objets au hasard. Supposons que c'est 1, 2 et 3. Ca donne C_1={1}, M_1=1, C_2={2}, M_2=2, C_3={3} et M_3=3
- Chaque objet O est affecté au cluster au milieu duquel, O est le plus proche. 6 est affecté à C_3 car dist(M_3,6)<dist(M_2,6) et dist(M_3,6)<dist(M_1,6)

  On a C_1={1}, M_1=1,

  C_2={2}, M_2=2

  C_3={3, 6,7,8,13,15,17}, M3=69/7=9.86

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11
- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12
- dist(2,M1)<dist(2,M2)➔2 passe en C1. dist(7,M2)<dist(7,M3)➔ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25
- dist(3,M1)<dist(3,M2)➔3 passe en 1. dist(8,M2)<dist(8,M3)➔8 passe en 2 C1={1,2,3}, M1=2, C2={6,7,8}, M2=7, C3={13,15,17}, M3=15

Plus rien ne bouge

Commentaires sur la méthode des *K-Means*

- Force
  - *Relativement efficace*: *O*(*tkn*), où *n* est # objets, *k* est # clusters, et *t est* # itérations. Normalement, *k*, *t* << *n*.
  - Tend à réduire $$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - m_i \right|^2$$
- Faiblesses
  - N'est pas applicable en présence d'attributs qui ne sont pas du type intervalle (moyenne=?)
  - On doit spécifier k (nombre de clusters)
  - Les clusters sont construits par rapports à des objets inexistants (les milieux)
  - Ne peut pas découvrir les groupes *non-convexes*

La méthode des *K-Medoids (PAM)*

- Trouver des objets représentatifs (medoïdes) dans les clusters (au lieu de la moyenne)
- Principe
  - Commencer avec un ensemble de medoïdes puis itérativement remplacer un par un autre si ça permet de réduire la distance globale
  - Efficace pour des données de petite taille

Algorithme des k-Medoides

Choisir arbitrairement k medoides
Répéter
    affecter chaque objet restant au medoide le plus proche
    Choisir aléatoirement un non-medoide O_r
    Pour chaque medoide O_j
        Calculer le coût TC du remplacement de O_j par O_r
        Si TC < o alors
            Remplacer O_j par O_r
            Calculer les nouveaux clusters
        Finsi
    FinPour
Jusqu'à ce ce qu'il n'y ait plus de changement

PAM (Partitioning Around Medoids) (1987)

Choisir arbitrairement ***k*** objets représentatifs

- Pour toute paire (h,j) d'objets t.q h est choisi et j non, calculer le coût $\boldsymbol{TC_{jh}}$ du remplacement de j par h
  - Si $TC_{ih} < 0$, ***j*** est remplacé par ***h***
  - Puis affecter chaque objet non sélectionné au medoïde qui lui est le plus similaire
- Répéter jusqu'à ne plus avoir de changements

La méthode des *K-Medoids*

- TC_{jh} représente le gain en distance globale que l'on va avoir en remplaçant h par j
- Si TC_{jh} est négatif alors on va perdre en distance. Ca veut dire que les clusters seront plus compacts.
- $TC_{jh}=\Sigma_i$ dist(j,h)-dist(j,i)= $\Sigma_i C_{ijh}$

La méthode des *K-Medoids:* Exemple

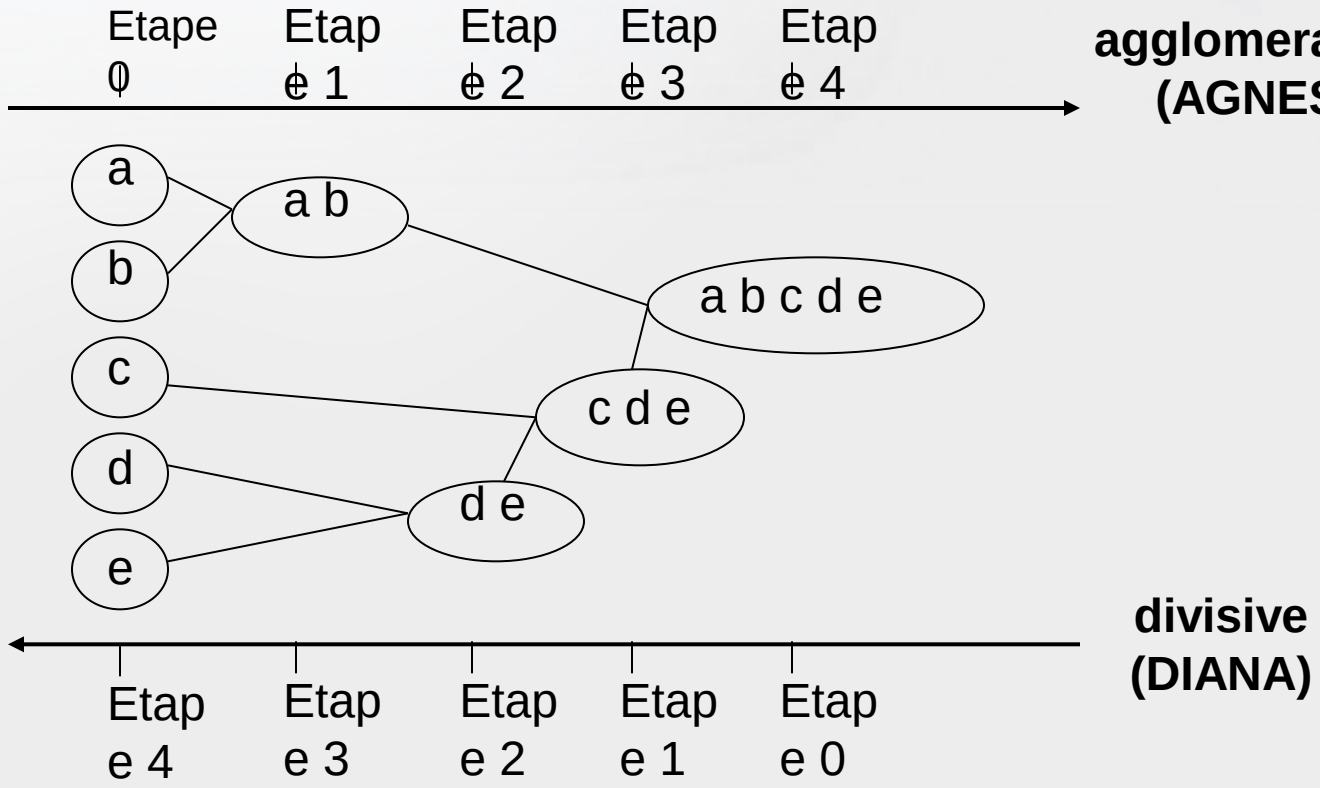
- Soit A={1,3,4,5,8,9}, k=2 et M={1,8} ensemble des medoides
  ➔C1={1,3,4} et C2={5,8,9}
  $E_{\{1,8\}}=dist(3,1)^2+dist(4,1)^2+dist(5,8)^2+dist(9,8)^2=23$

- Comparons 1 et 3➔M={3,8}➔C1={1,3,4,5} et C2={8,9}
  $E_{\{3,8\}}=dist(1,3)^2+dist(4,3)^2+dist(5,3)^2+dist(9,8)^2=10$
  $E_{\{3,8\}} - E_{\{1,8\}}= -13 <0$ donc le remplacement est fait.

- Comparons 3 et 4➔ M={4,8}➔ C1 et C2 inchangés et
  $E_{\{4,8\}}=dist(1,4)^2+dist(3,4)^2+dist(5,4)^2+dist(8,9)^2= 12$➔ 3 n'est pas remplacé par 4
- Comparons 3 et 5➔M={5,8}➔ C1 et C2 inchangés et E{5,8}>E{3,8}

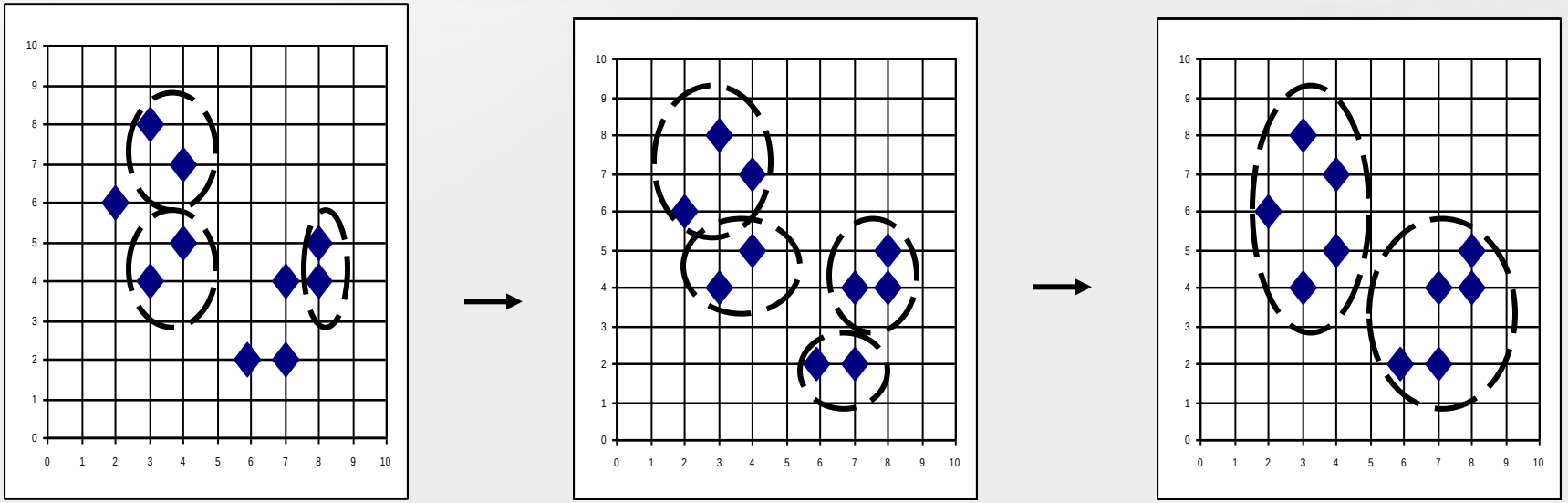
Clustering Hiérarchique

- Utiliser la matrice de distances comme critère de regroupement. ***k*** n'a pas à être précisé, mais a besoin d'une condition d'arrêt



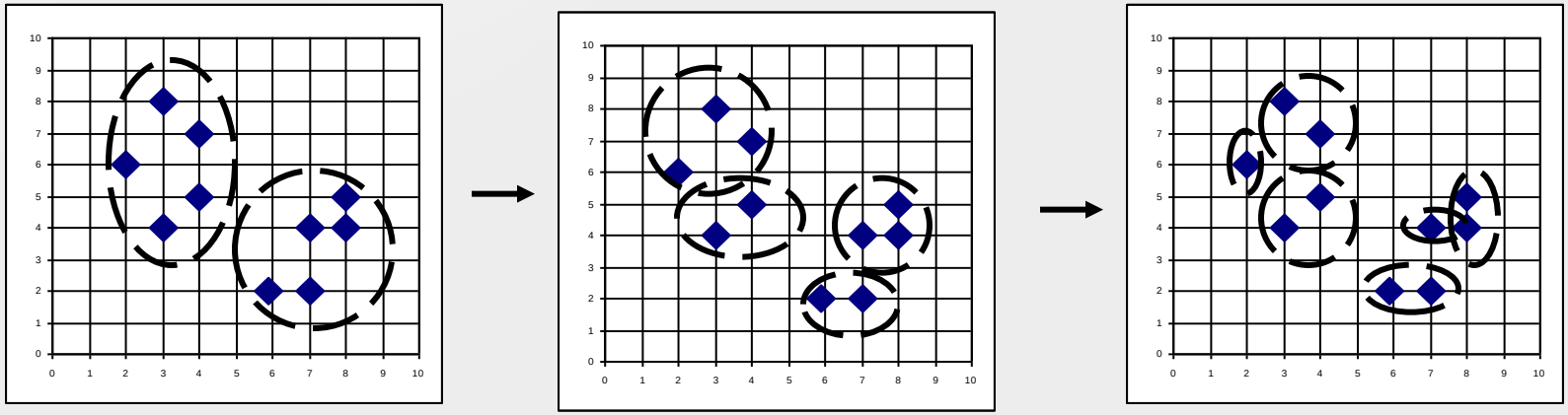
AGNES (Agglomerative Nesting)

- Utilise la matrice de dissimilarité.
- Fusionne les nœuds qui ont la plus faible dissimilarité
- On peut se retrouver dans la situation où tous les nœuds sont dans le même groupe



DIANA (Divisive Analysis)

- L'ordre inverse de celui d'AGNES
- Il se peut que chaque objet forme à lui seul un groupe



Critères de fusion-éclatement

- Exemple: pour les méthodes agglomératives, C1 et C2 sont fusionnés si
  - il existe o1 ∈ C1 et o2∈ C2 tels que dist(o1,o2) ≤ seuil, ou
  - il n'existe pas o1 ∈ C1 et o2∈ C2 tels que dist(o1,o2) ≥ seuil, ou
  - distance entre C1 et C2 ≤ seuil avec

$$dist\ (C_1, C_2) = \frac{1}{n1 * n2} \sum_{o1 \in C1, o2 \in C2} dist(o1, o2)$$

  et n1=|C1|.
- Ces techniques peuvent être adaptées pour les méthodes divisives

Méthodes d'agrégation

- **Lien minimum**
- $\delta(A, B) = \min\{d(a, b), a \in A, b \in B\}$

- **Lien maximum**
- $\delta(A, B) = \max\{d(a, b), a \in A, b \in B\}$

- **Distance des centres de gravité**
- $\delta(A, B) = d(ga, gb)$

Exemple

Agrégation selon le lien minimum

	a	*b*	*c*	*d*	*e*
a	0	23	35	43	50
b	23	0	21	32	45
c	35	21	0	11	25
d	43	32	11	0	17
e	50	45	25	17	0

Tableau des dissimilarités

$G_1 = \{c, d\} \Rightarrow$

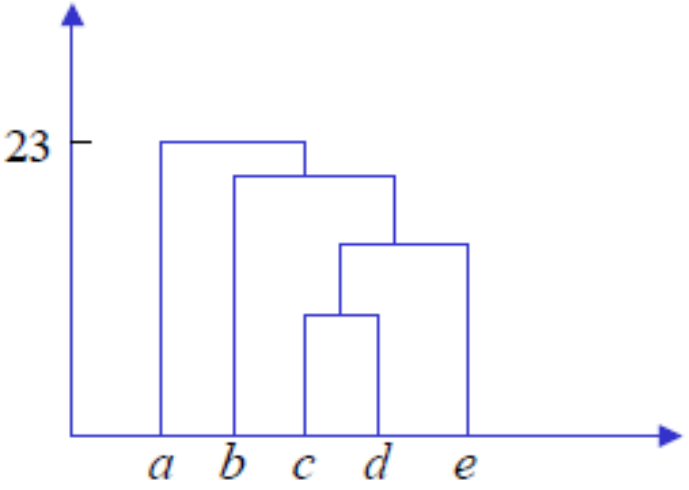
	a	*b*	*e*	G_1
a	0	23	50	35
b	23	0	45	21
e	50	45	0	17
G_1	35	21	17	0

$G_2 = \{e, G_1\} \Rightarrow$

	a	*b*	G_2
a	0	23	35
b	23	0	21
G_2	35	21	0

$G_3 = \{b, G_2\} \Rightarrow$

	a	G_3
a	0	23
G_3	23	0



Exemple (suite)

Agrégation selon le lien maximum

	a	*b*	*c*	*d*	*e*
a	0	23	35	43	50
b	23	0	21	32	45
c	35	21	0	11	25
d	43	32	11	0	17
e	50	45	25	17	0

Tableau des dissimilarités

$G_1 = \{c, d\} \Rightarrow$

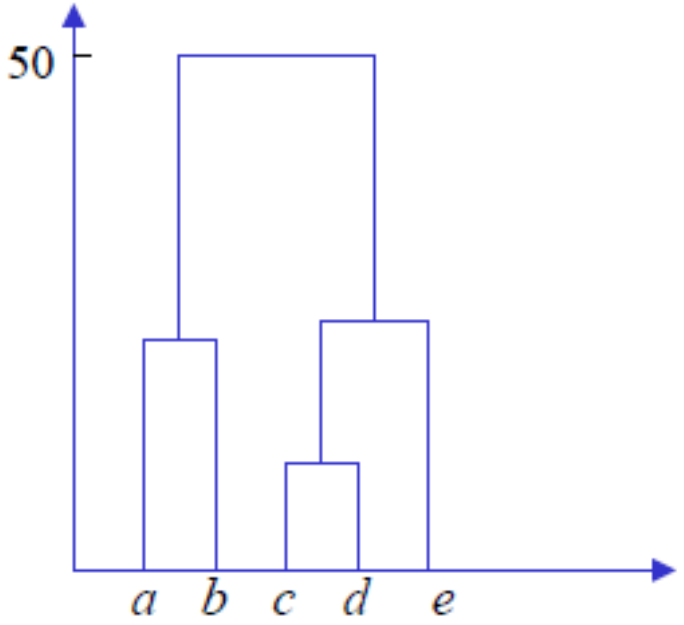
	a	*b*	*e*	G_1
a	0	23	50	43
b	23	0	45	32
e	50	45	0	25
G_1	43	32	25	0

$G_2 = \{a, b\} \Rightarrow$

	e	G_1	G_2
e	0	25	50
G_1	25	0	43
G_2	50	43	0

$G_3 = \{e, G_1\} \Rightarrow$

	G_2	G_3
G_2	0	50
G_3	50	0



Inerties interclasse et intraclasse

Soit une classification en k groupes d'effectifs $n_1, \ldots, n_k$, les individus étant des points d'un espace euclidien. Notons les groupes $G_1, \ldots, G_k$, et $g_1, \ldots, g_k$ leurs centres de gravité (g est le centre de gravité du nuage).

Inertie totale :
$$I_{tot} = \frac{1}{n} \sum_{i=1}^{n} d^2(e_i, g)$$

Inertie interclasse :
$$I_{inter} = \frac{1}{n} \sum_{i=1}^{k} n_i \cdot d^2(g_i, g)$$

Inertie intraclasse :
$$I_{intra} = \frac{1}{n} \sum_{i=1}^{k} \sum_{e \in G_i} d^2(e, g_i)$$

Critère d'agrégation selon l'inertie

Théorème de Huygens :

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

- Au fur et àmesure que les regroupements sont effectués, l'inertie intra-classe augmente et l'inertie interclasse diminue, car leur somme est une constante liée aux données analysées.