Classification et groupement de données

Apprentissage non supervisé

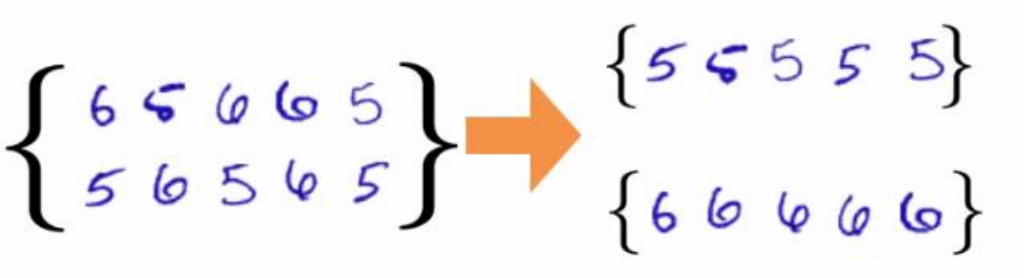
- Contrairement à l'apprentissage supervisé qui fournit les valeurs réelles de sorties pour les données d'entrées, l'apprentissage non supervisé ne fournit pas ces sorties.

- En général, les données d'apprentissage sont données par:

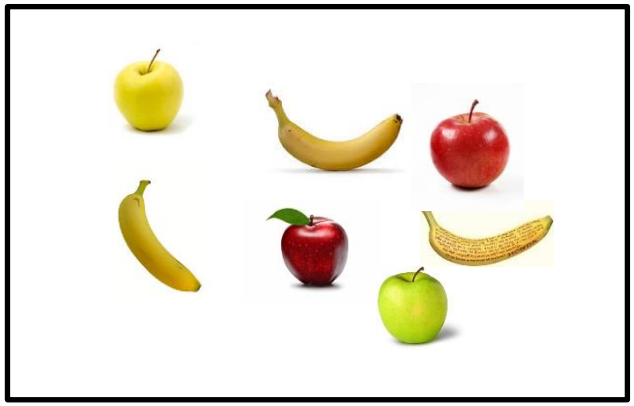
$$\mathcal{D} = \left\{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\right\}$$

- Le but est d'utiliser l'ensemble $\mathcal{D}$ pour construire des groupes homogènes qui constitueront des classes de données:

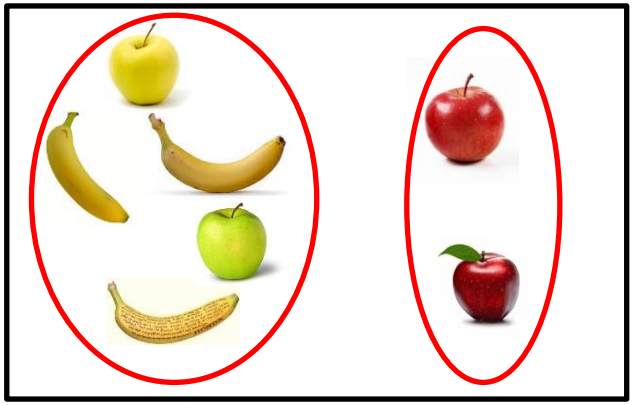
Ex:



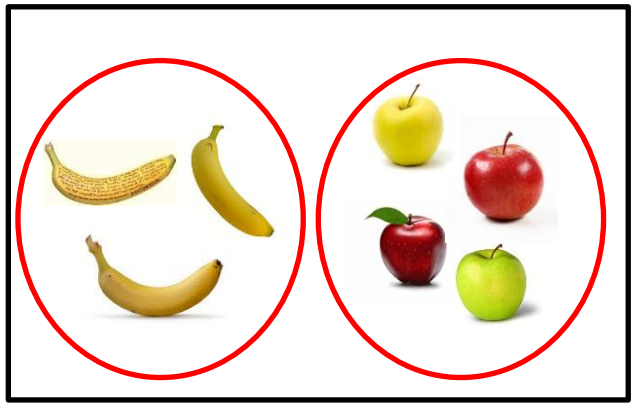
Apprentissage non supervisé



Regroupement par couleur



Regroupement par forme



Groupement de données

- Il y a plusieurs motivation pour le groupement de données.

  Ex: Soit le nombre **‘sept’** qui peut être écrit **‘7’** en version nord-américaine et et **‘7’**, en version européenne.

  ☞ Lorsque nous possédons des documents contenant les deux version du symbole, la classe du chiffre **‘sept’** doit être représentée par deux groupes.

- Un exemple similaire peut être donné dans la **prononciation d’un mot** qui dépendra de l’accent, de l'âge, etc.:

  Ex. ‘oui’, ‘ouai’, ‘ouiii’, etc.

Groupement de données

En général, on peut faire un groupement de données:

- ☞ Pour réduire le nombre de données d'apprentissage (**ex.** pour étiqueter manuellement une grande base de données).
- ☞ Pour le codage et la compression de données.
- ☞ Pour découvrir la nature et la structure des données.
- ☞ Pour la classification de données.
- ☞ Pour grouper des données dont les caractéristiques peuvent changer avec le temps (**ex**. groupes de personnes, etc.) .

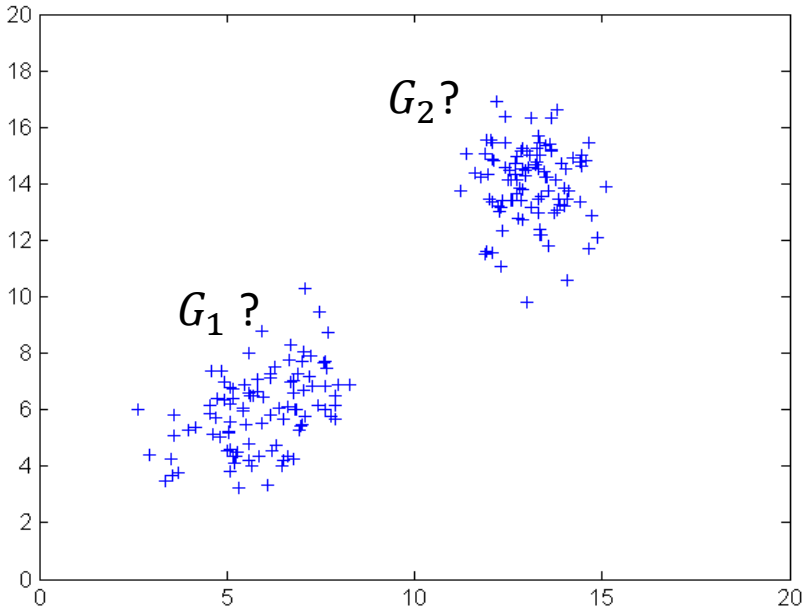
Algorithme des K-moyennes

Principe

Intuitivement:

☞ Un groupe est un ensemble de données dont les éléments sont proches en distance.

☞ Les éléments en dehors du groupe auront une grande distance par rapport aux éléments du groupe.



Algorithme du K-moyennes

- Supposons que nous avons un ensemble de N données $\mathcal{D} = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$. Chaque donnée $x^{(i)}$ possède D dimensions réelles: $({x_1}^{(i)}, {x_2}^{(i)}, \ldots, {x_D}^{(i)})$.

- Supposons que les données dans $\mathcal{D}$ appartiennent à K différents groupes inconnus $\{G_1, \ldots., G_K\}$ (*le nombre de groupe est fournie pour le moment*).

- Soit un ensemble de K vecteurs $\{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \ldots, \boldsymbol{\mu}^{(K)}\}$ inconnus. Chaque vecteur $\boldsymbol{\mu}^{(k)}$ possèdera D dimensions : $({\mu_1}^{(k)}, {\mu_2}^{(k)}, \ldots, {\mu_D}^{(k)})$.

- On associera un vecteur $\boldsymbol{\mu}^{(k)}$ (centre) pour chaque groupe G_k.

Algorithme du K-moyennes

- Pour chaque donnée $x^{(i)}$, on associe une variable indicatrice $r_{ik} \in \{0,1\}$, qui prendra sa valeur comme suit:

$$r_{ik} = \begin{cases} 1 & si \quad x^{(i)} \in G_k \\ 0 & si \quad x^{(i)} \notin G_k \end{cases}$$

- On peut définir une fonction objective (somme des carrés résiduels: *SCR*) à minimiser pour réaliser un groupement optimal:

$$SCR = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \left\| x^{(i)} - \boldsymbol{\mu}^{(k)} \right\|^2$$

- Le but est de trouver les r_{ik} et les $\mu^{(k)}$ pour minimiser SCR?

Algorithme du K-moyennes

- La solution peut être atteinte par une procédure itérative. Chaque itération impliquera deux étapes successives:

  ☞ Optimiser SCR par rapport à r_{ik}, $i \in \{1, \dots, N\}$ $k \in \{1, \dots, K\}$.

  ☞ Optimiser SCR par rapport à $\boldsymbol{\mu}^{(k)}$, $k \in \{1, \dots, K\}$.

- Ces deux étapes sont répétées jusqu'à la convergence.

- Au départ, il faudra initialiser les $\boldsymbol{\mu}^{(k)}$, $k \in \{1, \dots, K\}$ (**ex.** de manière aléatoires).

Algorithme du K-moyennes

Étape 1:

- On peut facilement optimiser SCR par rapport à chaque r_{ik}, en choisissant r_{ik} égal à 1 lorsque $\boldsymbol{\mu}^{(k)}$ est le centre le plus proche de la donnée $x^{(i)}$.

- Autrement dit, on assigne la donnée $x^{(i)}$ au groupe dont le centre est le plus prêt de la donnée:

$$r_{ik} = \begin{cases} 1 & si\ k = \text{argmin}_l \left\| x^{(i)} - \boldsymbol{\mu}^{(l)} \right\| \\ 0 & sinon \end{cases}$$

Algorithme du K-moyennes

Étape 2:

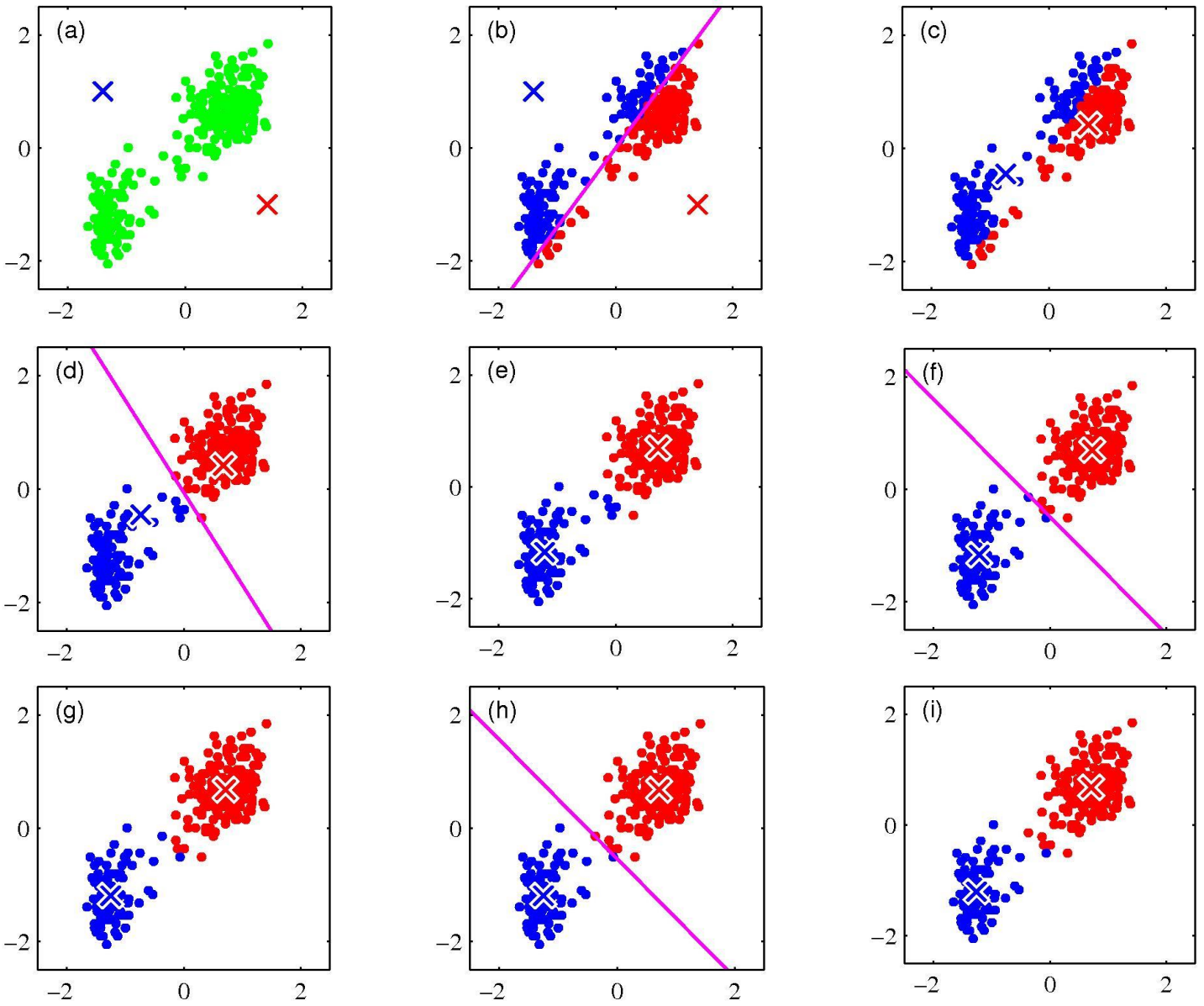
- En gardant tous les r_{ik} fixes, on optimise SCR par rapport à chaque $\boldsymbol{\mu}^{(k)}$ en minimisant SCR par rapport à ce dernier.

- En mettant la dérivée de SCR par rapport à $\boldsymbol{\mu}^{(k)}$ égale à 0, on aura:

$$2\sum_{i=1}^{N} r_{ik}\left(x^{(i)} - \boldsymbol{\mu}^{(k)}\right) = 0 \quad \Rightarrow \quad \boldsymbol{\mu}^{(k)} = \frac{\sum_{i=1}^{N} r_{ik} x^{(i)}}{\sum_{i=1}^{N} r_{ik}}$$

- Noter que $\sum_{i=1}^{N} r_{ik}$ donne le nombre de données assignées au groupe G_k.

Algorithme du K-moyennes (exemple 1)



Algorithme du K-moyennes (exemple 2)

Segmentation d'images:

pixel avec valeurs: R, G, B.

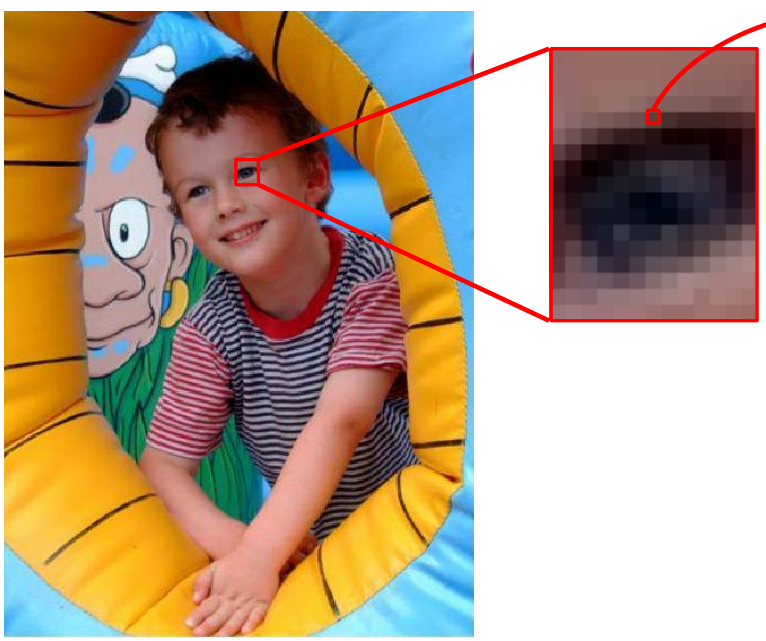
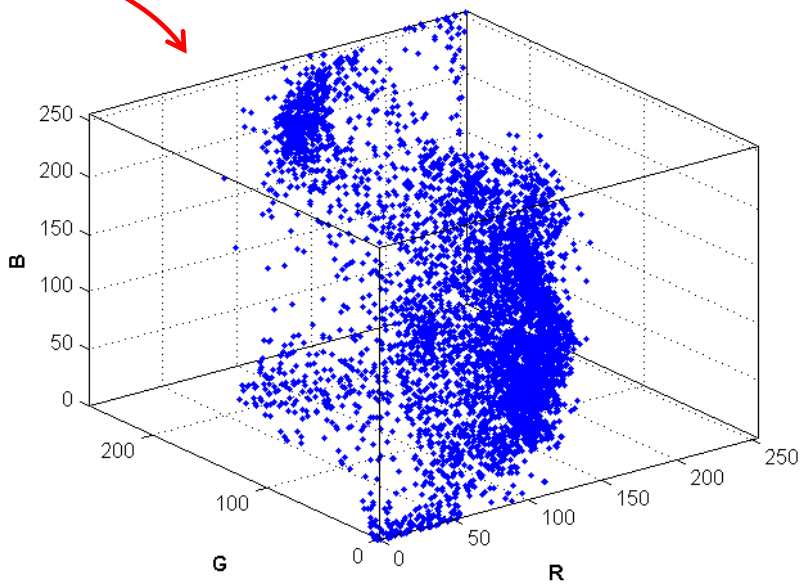


Image couleur

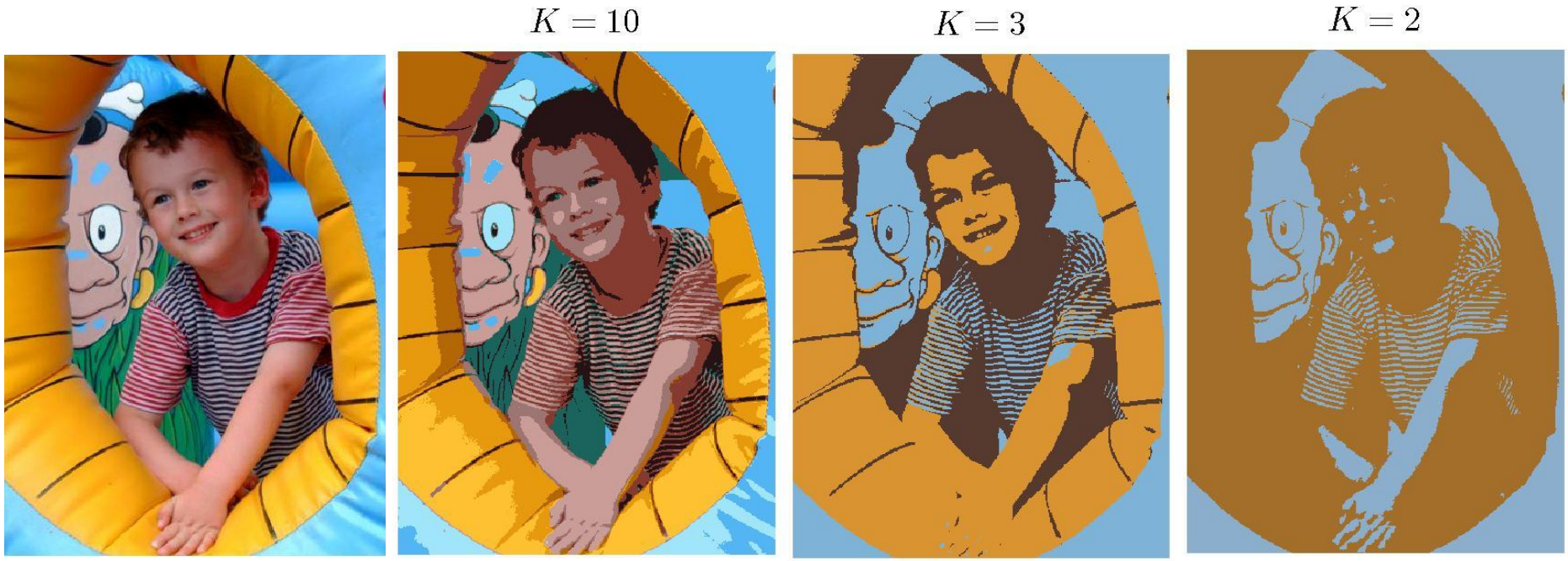


Distribution des couleurs

Algorithme du K-moyennes (exemple 2)

Segmentation d'images:

En remplaçant chaque pixel par la moyenne de son groupe:



Mélange de distributions Gaussiennes (MDG)

Rappel sur la loi Gaussienne

- Soit D variables aléatoires $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_D$ et soit $\mu_1, \mu_2, \ldots, \mu_D$ leurs moyennes, respectivement.

- On définit la matrice de covariance $\boldsymbol{\Sigma}$ de dimension $D \times D$ dont les entrées sont définies par: $\boldsymbol{\Sigma}_{i,j} = cov(\mathrm{x}_i, \mathrm{x}_j)$.

- On définit la loi Gaussienne multivariée par: $\mathrm{x} \rightsquigarrow \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ où $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_D)^T$ et $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_D)^T$:

$$f(\mathrm{x} = u) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(u - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(u - \boldsymbol{\mu})\right)$$

Rappel sur la loi Gaussienne

- Par exemple, pour $D = 2$, on peut définir les distribution Gaussiennes suivantes $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Pour chaque distribution on a $\boldsymbol{\mu} = (0,0)^T$ et les matrices de covariances sont données comme suit:

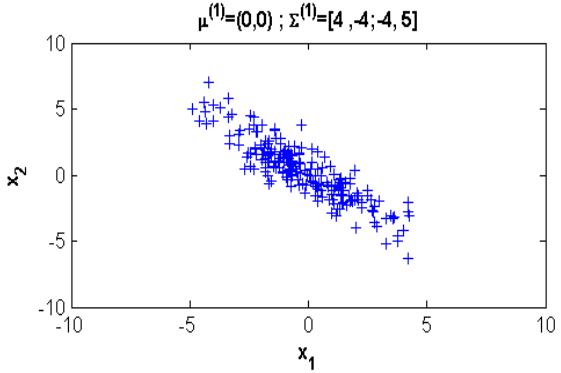
$$\boldsymbol{\Sigma}^{(1)} = \begin{bmatrix} 4 & -4 \\ -4 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(2)} = \begin{bmatrix} 4 & -3 \\ -3 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(3)} = \begin{bmatrix} 4 & -2 \\ -2 & 5 \end{bmatrix}$$

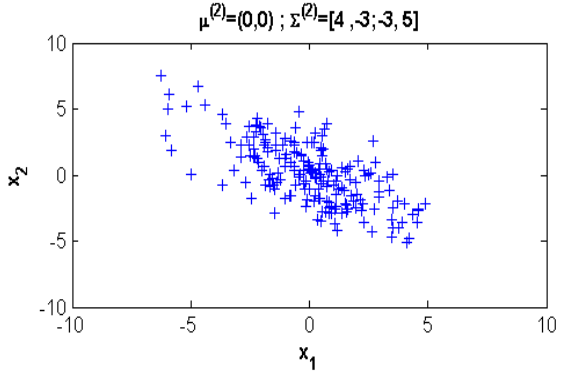
$$\boldsymbol{\Sigma}^{(4)} = \begin{bmatrix} 4 & -1 \\ -1 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(5)} = \begin{bmatrix} 4 & 0 \\ 0 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(6)} = \begin{bmatrix} 4 & 1 \\ 1 & 5 \end{bmatrix}$$

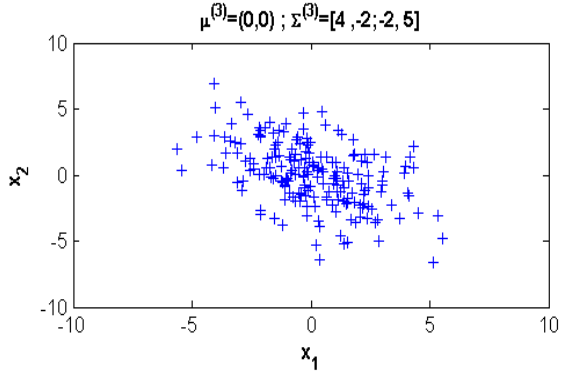
$$\boldsymbol{\Sigma}^{(7)} = \begin{bmatrix} 4 & 2 \\ 2 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(8)} = \begin{bmatrix} 4 & 3 \\ 3 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(9)} = \begin{bmatrix} 4 & 4 \\ 4 & 5 \end{bmatrix}$$

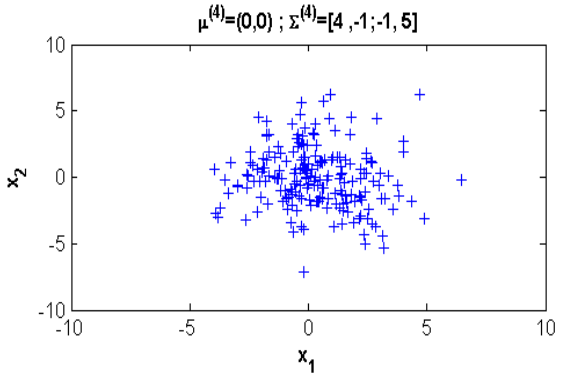
- 200 données ont été générées pour chaque Gaussienne.

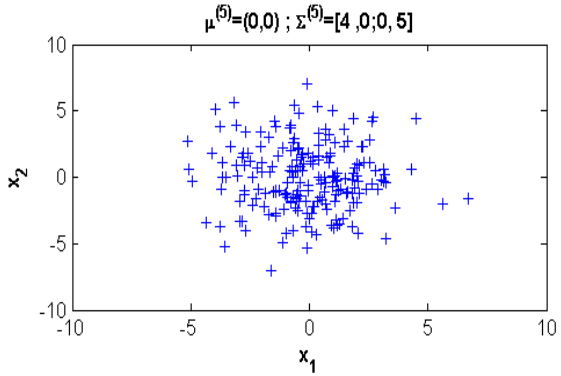
Rappel sur la loi Gaussienne

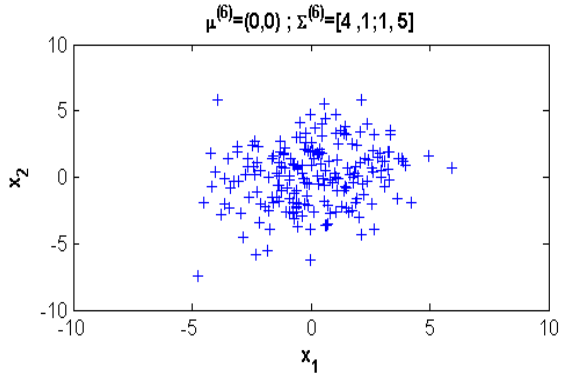


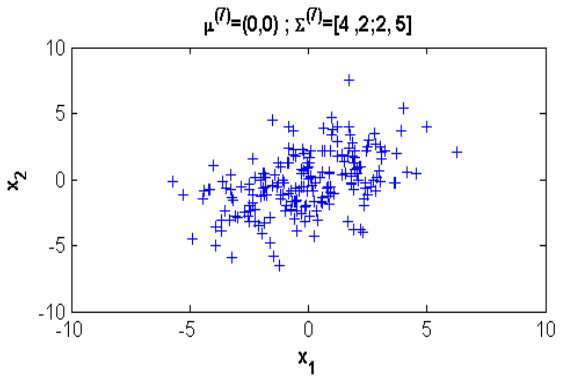


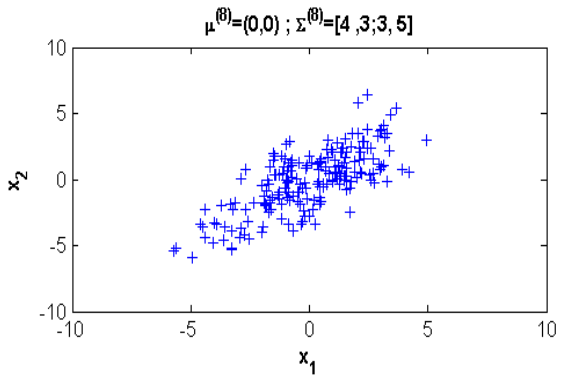


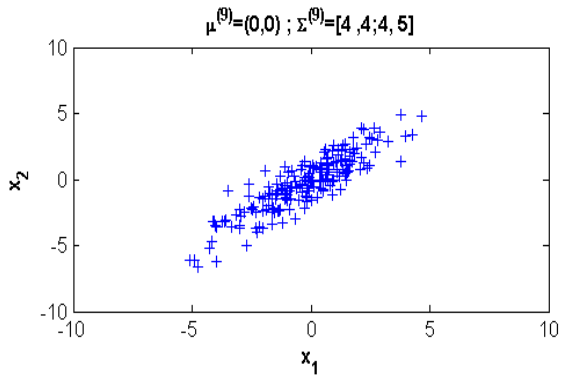












Mélanges de distributions Gaussiennes (MDG)

- Soit $\mathcal{D} = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ un ensemble de N données d'apprentissage.

- On suppose que les données de $\mathcal{D}$ sont constituées de K groupes: $G_1, \ldots, G_K$. Chaque groupe G_k a une distribution Gaussienne de moyenne $\boldsymbol{\mu}^{(k)}$ et covariance $\boldsymbol{\Sigma}^{(k)}$.

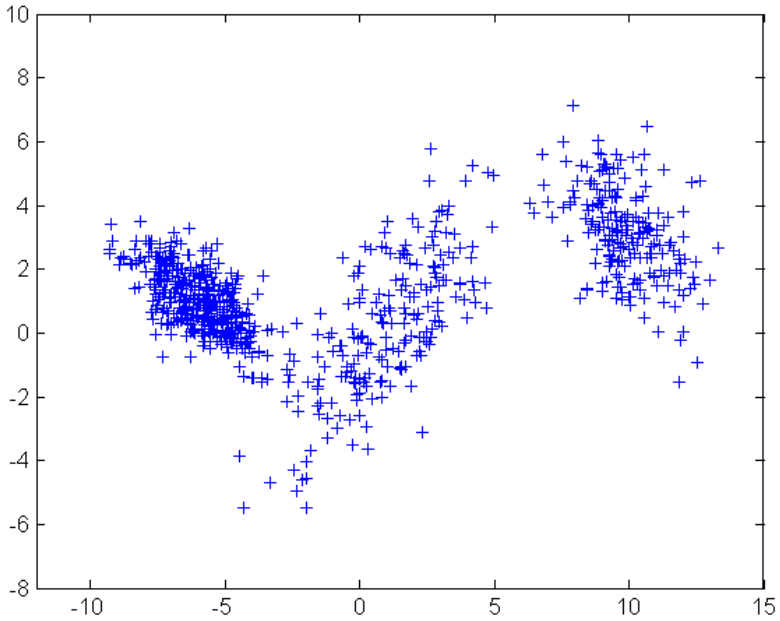
- Soit une donnée $x^{(i)} \in \mathcal{D}$. Par marginalisation, on définit la probabilité $p(x^{(i)})$ de cette donnée comme suit:

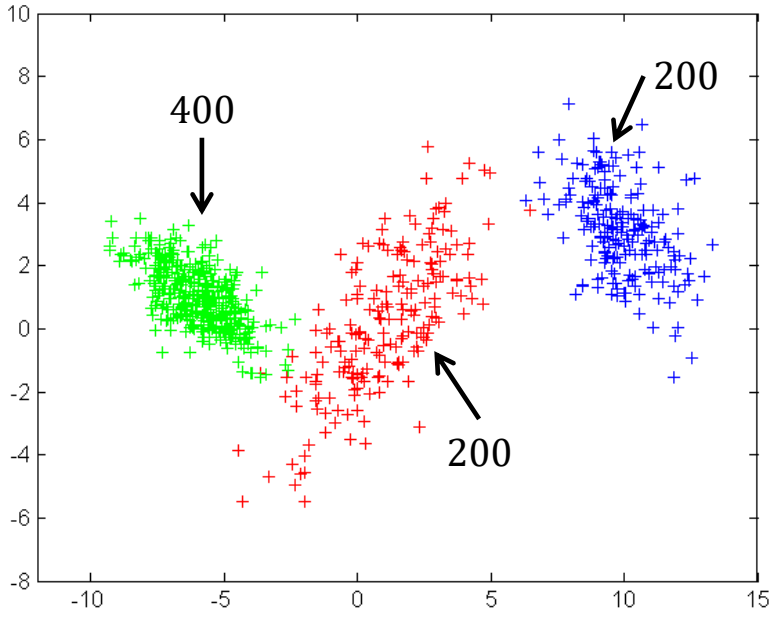
$$p(x^{(i)}) = \sum_{k=1}^{K} p(x^{(i)}|G_k)p(G_k) \qquad \text{où} \qquad \sum_{k=1}^{K} p(G_k) = 1$$

Où $p(x|G_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ et $p(G_k)$ est la probabilité du groupe G_k.

Mélanges de distributions Gaussiennes (MDG)

- Par exemple, soit l'ensemble $\mathcal{D}$ de $N = 800$ données de dimension $D = 2$ qui forme un mélange de $K = 3$ groupes:





$$\boldsymbol{\mu}^{(1)} = \begin{bmatrix} -6 \\ 1 \end{bmatrix} \quad \boldsymbol{\mu}^{(2)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \boldsymbol{\mu}^{(3)} = \begin{bmatrix} 10 \\ 3 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{(1)} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(2)} = \begin{bmatrix} 4 & 3 \\ 3 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(3)} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

$$p(G_1) = 1/2 \quad p(G_2) = 1/4 \quad p(G_3) = 1/4$$

Estimation des paramètres d'un MDG

Cas 1: Les données sont déjà assignées aux groupes.

- Pour K groupes: $G_1, \ldots, G_K$, on doit estimer 3 paramètres pour chaque groupe G_k: $\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}, p(G_k)$.

- On dénote par $\boldsymbol{\varphi}$ l'ensemble de tous les paramètres à estimer:

$$\boldsymbol{\varphi} = \{G_k: \boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}, p(G_k),\ k = 1, \ldots, K\}$$

- Pour chaque donnée $x^{(i)}$, on associe une variable indicatrice:

$$r_{ik} = \begin{cases} 1 & si \quad x^{(i)} \in G_k \\ 0 & si \quad x^{(i)} \notin G_k \end{cases},\ k \in \{1, \ldots, K\}.$$

Estimation des paramètres d'un MDG

Cas 1: Les données sont déjà assignées aux groupes.

- Il est facile de démontrer que:

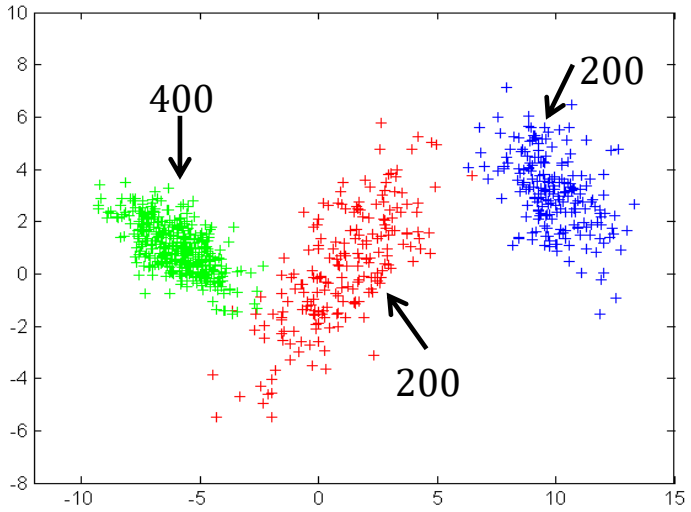
$$p(G_k) = \frac{\sum_{i=1}^{N} r_{ik}}{N}$$

$$\widehat{\boldsymbol{\mu}}^{(k)} = \frac{\sum_{i=1}^{N} r_{ik} x^{(i)}}{\sum_{i=1}^{N} r_{ik}}$$

$$\widehat{\boldsymbol{\Sigma}}^{(k)} = \frac{\sum_{i=1}^{N} r_{ik} \left(x^{(i)} - \mu^{(k)}\right)\left(x^{(i)} - \mu^{(k)}\right)^{T}}{\sum_{i=1}^{N} r_{ik}}$$

- **Ex.** Soit notre ensemble $\mathcal{D}$ de 800 données de dimension 2, qui a 3 groupes, et le groupe est connu pour chaque donnée.

Estimation des paramètres d'un MDG



- L'estimation des paramètres par Matlab avec $\mathcal{D}$ va donner :

$$\widehat{\boldsymbol{\mu}}^{(1)} = \begin{bmatrix} -5{,}9 \\ 0{,}96 \end{bmatrix} \qquad \widehat{\boldsymbol{\mu}}^{(2)} = \begin{bmatrix} 1{,}08 \\ 0{,}03 \end{bmatrix} \qquad \widehat{\boldsymbol{\mu}}^{(3)} = \begin{bmatrix} 10{,}1 \\ 2{,}99 \end{bmatrix}$$

$$\widehat{\boldsymbol{\Sigma}}^{(1)} = \begin{bmatrix} 1{,}72 & -0{,}85 \\ -0{,}85 & 0{,}97 \end{bmatrix} \qquad \widehat{\boldsymbol{\Sigma}}^{(2)} = \begin{bmatrix} 3{,}63 & 2{,}23 \\ 2{,}23 & 4{,}06 \end{bmatrix} \qquad \widehat{\boldsymbol{\Sigma}}^{(3)} = \begin{bmatrix} 1{,}91 & -1{,}06 \\ -1{,}06 & 1{,}97 \end{bmatrix}$$

$$p(G_1) = 1/2 \qquad p(G_2) = 1/4 \qquad p(G_3) = 1/4$$

Estimation des paramètres d'un MDG

Cas 2: Les donnée ne sont pas déjà assignées aux groupes.

- On utilise le maximum de vraisemblance:

$$\ell(\boldsymbol{\varphi}) = \log\left[\prod_{i=1}^{N} p(x^{(i)}|\boldsymbol{\varphi})\right]$$

$$= \sum_{i=1}^{N} \log\left[\sum_{k=1}^{K} p(x^{(i)}|G_k)p(G_k)\right]$$

- On ne peut pas trouver les paramètres $\boldsymbol{\varphi}$ de manière exacte.

Algorithme Espérance-Maximisation (EM)

- Algorithme itératif pour estimer les paramètres d'un MDG.
- Pour chaque itération on doit faire deux étapes:

  ☞ **Étape 1 (Espérance):**

  Pour chaque données $x^{(i)}$, et pour chaque groupe G_k, on calcule la probabilité a posteriori $p(G_k|x^{(i)})$, comme suit:

$$p(G_k|x^{(i)}) = \frac{p(x^{(i)}|G_k)p(G_k)}{\sum_{j=1}^{K} p(x^{(i)}|G_j)p(G_j)}$$

- On dénotera cette probabilité $p(G_k|x^{(i)})$ par: t_{ik}.

Algorithme Espérance-Maximisation (EM)

☞ **Étape 2 (Maximisation):**

Pour chaque groupe G_k, on estime ses paramètres G_k: $\mu^{(k)}$, $\boldsymbol{\Sigma}^{(k)}$, $p(G_k)$ comme suit:

$$p(G_k) = \frac{\sum_{i=1}^{N} t_{ik}}{N}$$

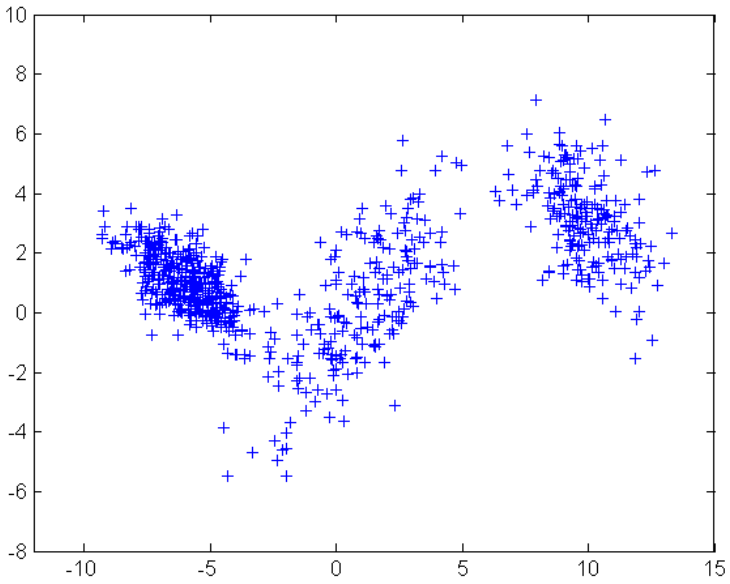
$$\widehat{\boldsymbol{\mu}}^{(k)} = \frac{\sum_{i=1}^{N} t_{ik} x^{(i)}}{\sum_{i=1}^{N} t_{ik}}$$

$$\widehat{\boldsymbol{\Sigma}}^{(k)} = \frac{\sum_{i=1}^{N} t_{ik} \left(x^{(i)} - \widehat{\boldsymbol{\mu}}^{(k)}\right)\left(x^{(i)} - \widehat{\boldsymbol{\mu}}^{(k)}\right)^{T}}{\sum_{i=1}^{N} t_{ik}}$$

Algorithme Espérance-Maximisation (EM)

- On répète les étapes 1 et 2 jusqu'à la convergence de l'algorithme EM.

- **Ex.** Soit notre ensemble $\mathcal{D}$ de 800 données de dimension 2 qui a 3 groupes, mais on ne connait le groupe d'aucune donnée:



Algorithme Espérance-Maximisation (EM)

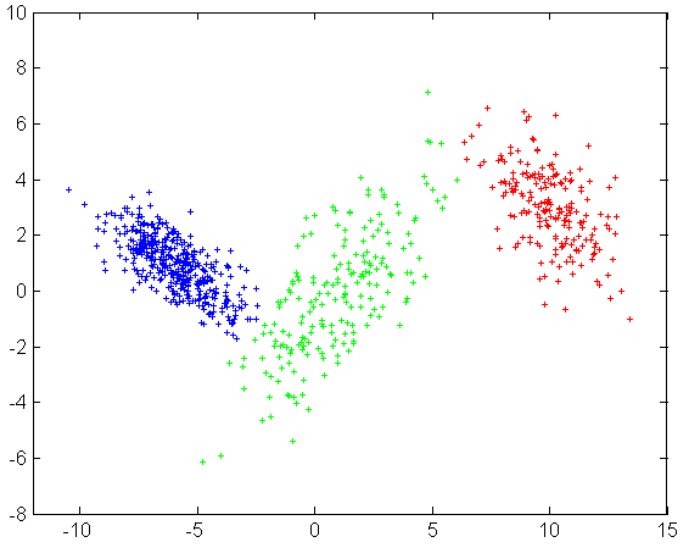
- En appliquant l'algorithme EM, on obtient:

$$\widehat{\boldsymbol{\mu}}^{(1)} = \begin{bmatrix} -6{,}01 \\ 1{,}03 \end{bmatrix} \qquad \widehat{\boldsymbol{\mu}}^{(2)} = \begin{bmatrix} 1{,}14 \\ 0{,}07 \end{bmatrix} \qquad \widehat{\boldsymbol{\mu}}^{(3)} = \begin{bmatrix} 10{,}03 \\ 3{,}3 \end{bmatrix}$$

$$\widehat{\boldsymbol{\Sigma}}^{(1)} = \begin{bmatrix} 1{,}8 & -0{,}92 \\ -0{,}92 & 0{,}93 \end{bmatrix} \qquad \widehat{\boldsymbol{\Sigma}}^{(2)} = \begin{bmatrix} 4{,}04 & 2{,}95 \\ 2{,}95 & 4{,}46 \end{bmatrix} \qquad \widehat{\boldsymbol{\Sigma}}^{(3)} = \begin{bmatrix} 1{,}77 & -1{,}06 \\ -1{,}06 & 1{,}94 \end{bmatrix}$$

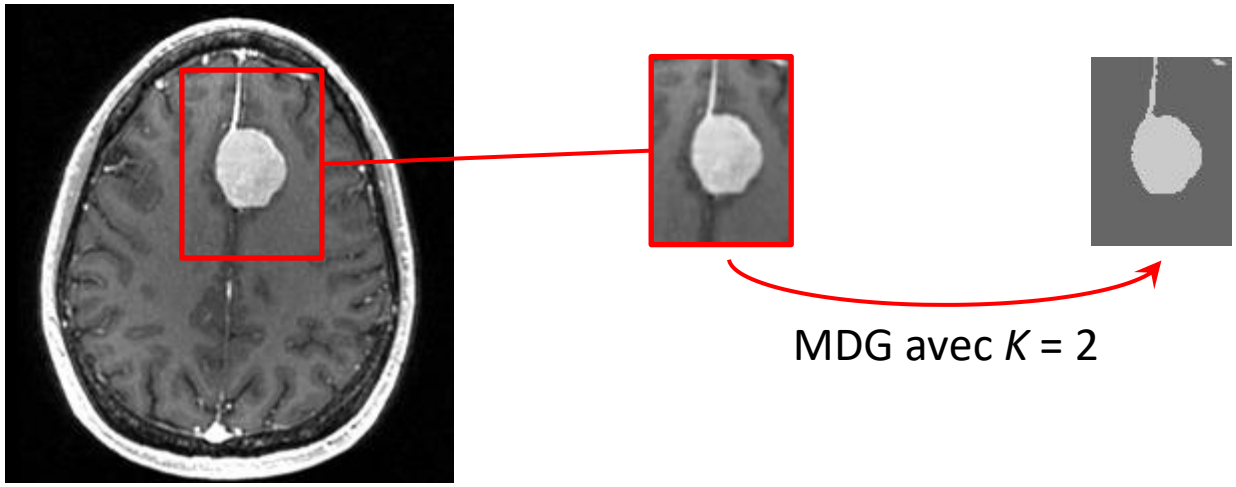
$$p(G_1) = 0{,}499 \qquad p(G_2) = 0{,}249 \qquad p(G_3) = 0{,}25$$

- En classant les données par le MAP, on obtient:



Exemple avec Matlab

- La segmentation d'images est un problème important pour plusieurs applications (ex. imagerie médicale, télédétection, robotique, inspection visuelle, etc.).



- Utiliser la fonction gmdistribution de Matlab.

http://www.mathworks.com/help/stats/gmdistribution.fit.html

Regroupement hiérarchique de données

Principe du groupement hiérarchique (GH)

- Repose uniquement sur l'analyse de la similarité entre les données pour leur groupement.

- Ainsi, le GH trouve des groupes tels que: Les instances appartenant à chaque groupe sont plus similaires que les instances appartenant à des groupes différents.

- La base de calcul de la similarité entre données est le calcul de distances. La plus simple des distances est la distance Euclidienne.

Principe du groupement hiérarchique (GH)

- On envisage une séquence de partitionnements de N données $\{x^{(1)}, x^{(2)},..., x^{(N)}\}$ en groupes.

- Le 1er partitionnement contiendra N groupes (c.-à-d. chaque donnée constituera un groupe).
- Le 2e partitionnement contiendra $N-1$ groupes.
- Le 3e partitionnement $N-2$ groupes

  ⋮

- Le $n^{\text{ième}}$ partitionnement contiendra en $N-n+1$ groupes.

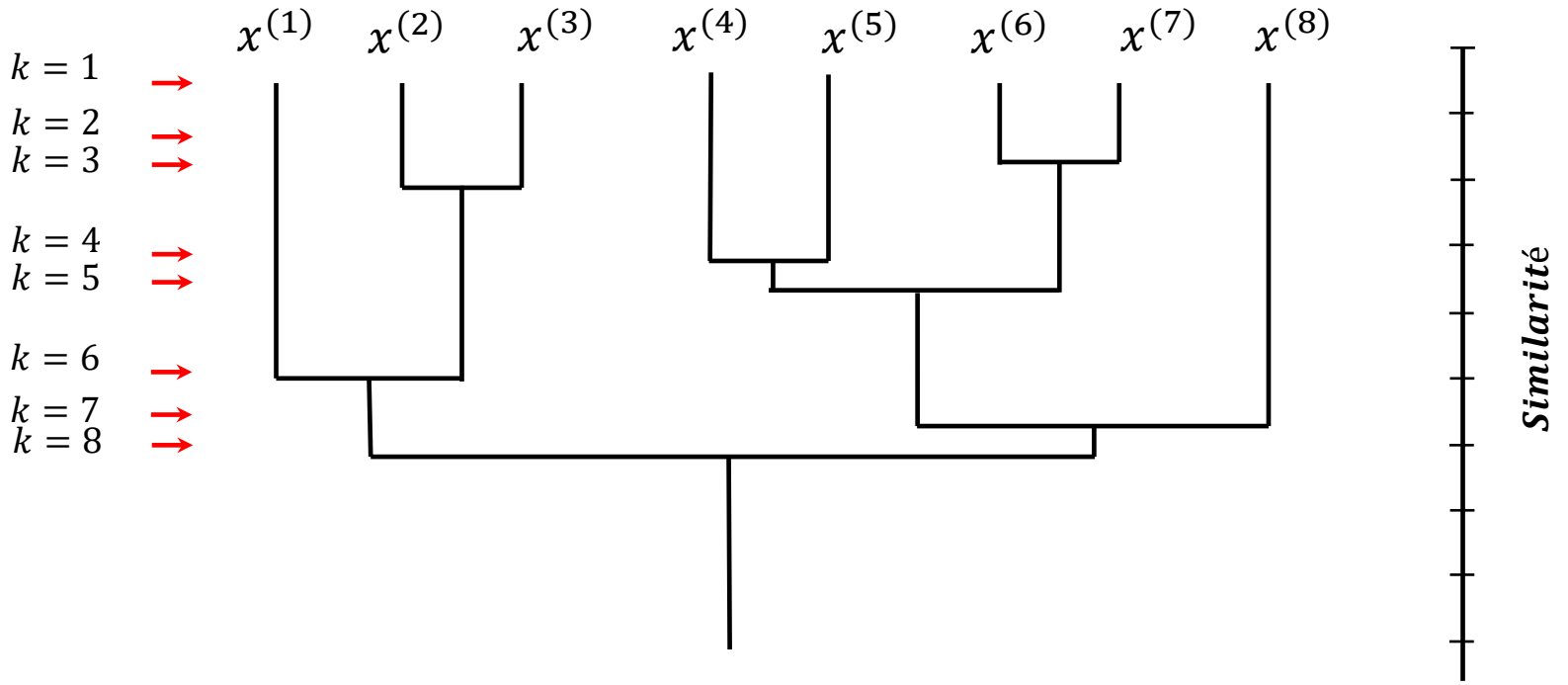
  ⋮

- Le $N^{\text{ième}}$ partitionnement contiendra en 1 groupe.

Principe du groupement hiérarchique (GH)

- La représentation naturelle d'un groupement hiérarchique ressemble à un arbre et est appelé un dendrogramme.

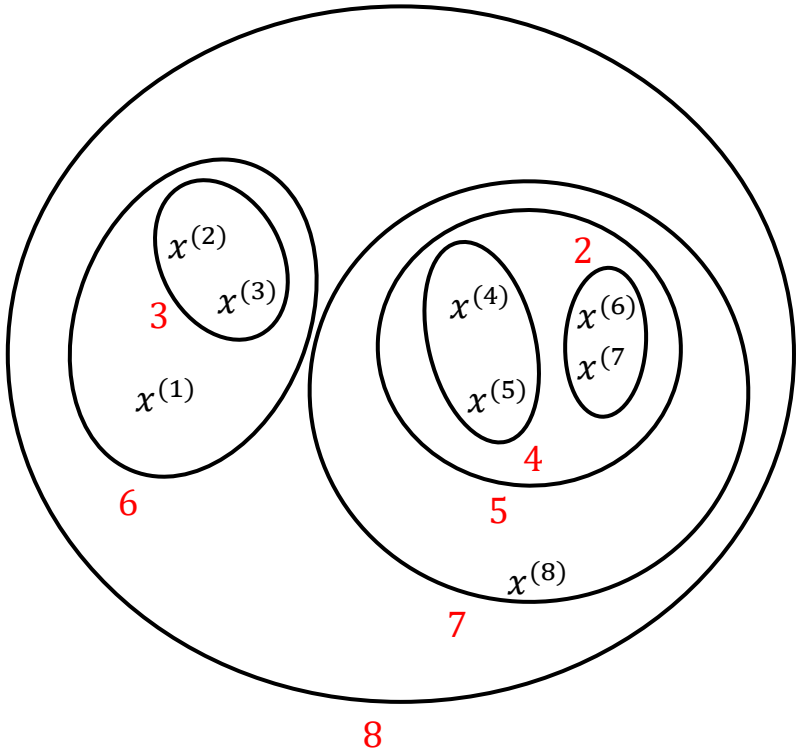
Exemple:



Principe du groupement hiérarchique (GH)

- Une autre représentation peut se baser sur les ensembles, comme montré dans l'exemple précédent .

$$\{\{x^{(1)}, \{x^{(2)}, x^{(3)}\}\}, \{\{\{x^{(4)}, x^{(5)}\}, \{x^{(6)}, x^{(7)}\}\}, x^{(8)}\}\}$$



Principe du groupement hiérarchique (GH)

- Les procédures pour la construction de regroupements hiérarchiques de données peuvent être de deux approches:

  ☞ Construction par agglomération (de bas en haut): commence par N groupes et forme des groupes successifs en fusionnant les groupes.

  ☞ Construction par division (de haut en bas): commence par 1 groupe et forme des groupes successifs en divisant les groupes en sous-groupes.

- La Construction par agglomération est plus simple.

Groupement par agglomération

Algorithme:

Entrées: $K, \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\},$

Sortie: K groupes.

$G_i \leftarrow \{x^{(i)}\}, i = 1, \dots, N;$

$c \leftarrow N;$

Tant que ($c \neq K$)

Trouver les groupes G_j et G_h les plus similaires;

Fusionner G_j et G_h;

$c \leftarrow c - 1;$

Fin

Groupement par agglomération

- Notons que le groupement est imbriqué, c.-à-d. si 2 exemples de données appartiennent au même groupe dans un niveau, il resteront dans un même groupe dans les niveaux supérieurs.

- Cela implique aussi un désavantage: si une erreur de groupement existe dans un niveau inferieur, elle se propagera dans le reste des niveaux supérieurs.

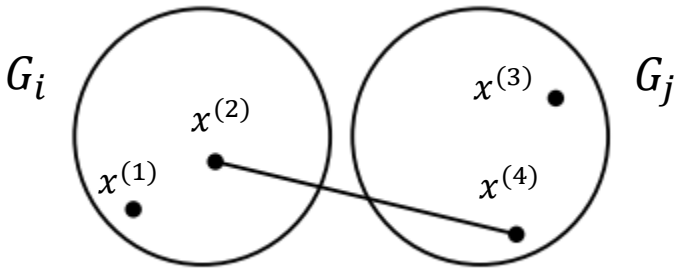
- À un niveau k, il existe $N - k + 1$ groupes. Pour choisir la paire de groupes à fusionner, il faudra faire $\binom{N-k+1}{2}$ calculs de similarité. Le nombre total de calculs exigés pour avoir K groupes est:

$$\sum_{k=1}^{K} \binom{N-k+1}{2}$$

Mesures de similarité

- Les distances de similarité utilisées pour fusionner les groupes G_i et G_j peuvent être de différents types:

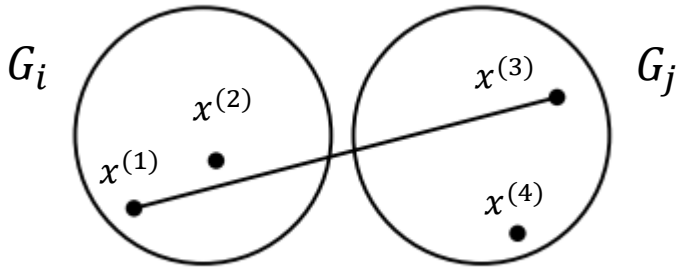
☞ Groupement par points similaires (single Link) mesure la similarité entre deux groupes par la distance entre les points les plus proches appartenant aux deux groupes.



$$dist_{min}(G_i, G_j) = min\left\|x^{(i)} - x^{(j)}\right\|; \qquad x^{(i)} \in G_i, x^{(j)} \in G_j ;$$

Mesures de similarité

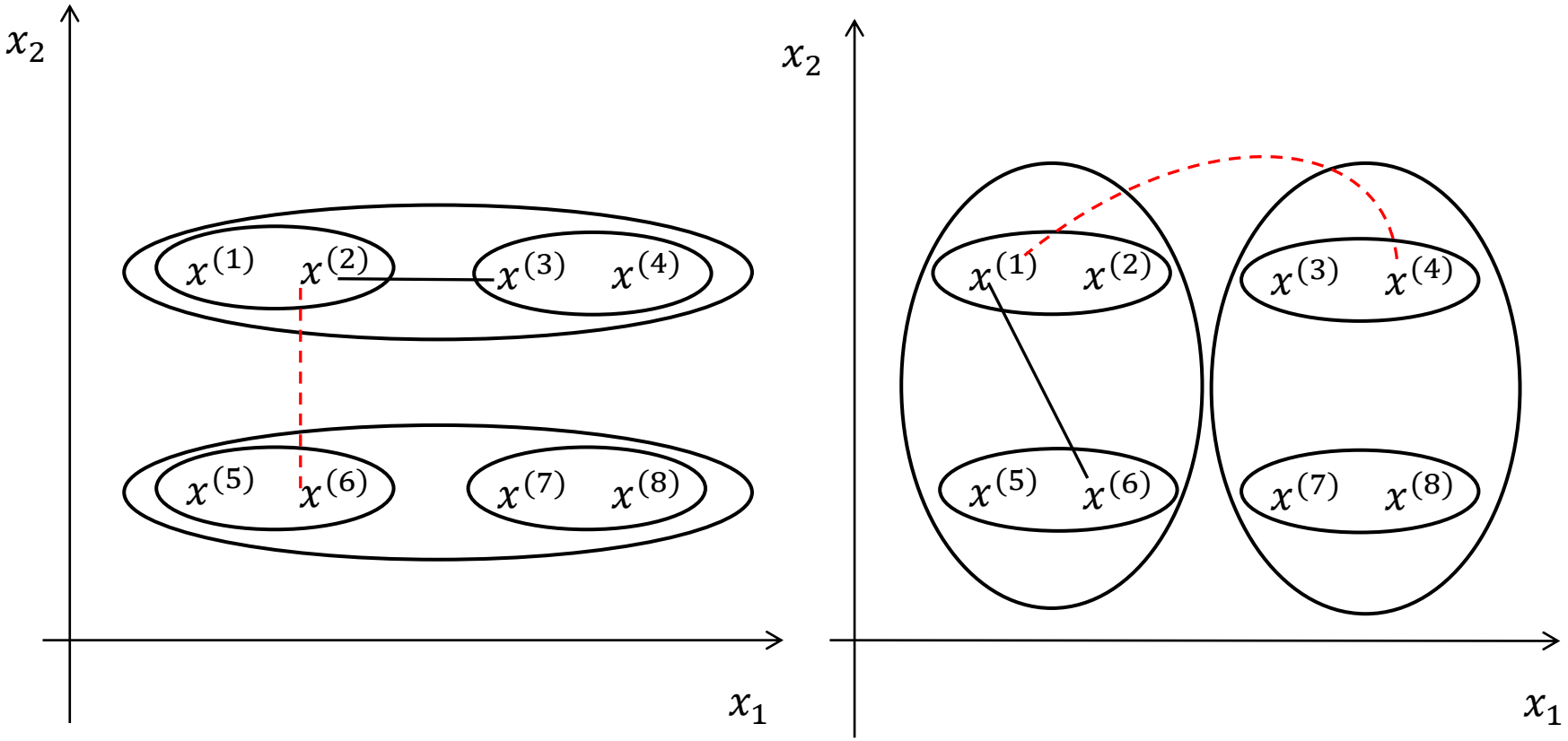
☞ Groupement par points dissimilaires (complete Link) mesure la similarité entre deux groupes par la distance entre les points les plus éloignés appartenant aux deux groupes.



$$dist_{max}(G_i, G_j) = max\left\|x^{(i)} - x^{(j)}\right\|; \qquad x^{(i)} \in G_i, x^{(j)} \in G_j ;$$

Mesures de similarité

Exemple:



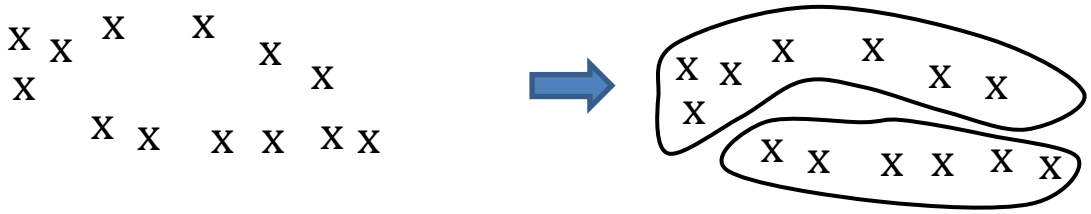
Mesures de similarité

Inconvénients:

- Les mesures signle Link et complete Link réduisent la similarité de deux groupes à une similarité entre points.

- La mesure signle Link peut produire des groupes épars.

Ex.



- La mesure complete Link est sensible aux données aberrantes

Ex.

Mesures de similarité

- D'autres mesures de similarité existes aussi:

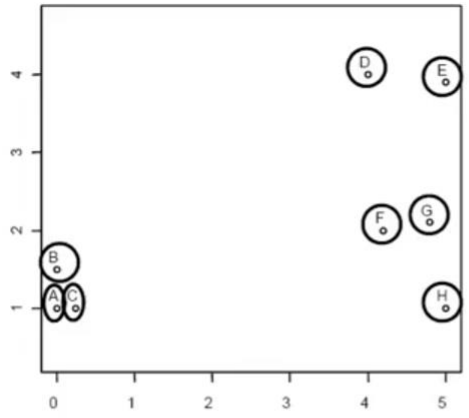
$$dist_{avg}(G_i, G_j) = \frac{1}{N_i N_j} \sum_{x^{(i)} \in G_i} \sum_{x^{(j)} \in G_j} \left\| x^{(i)} - x^{(j)} \right\|;$$

$$dist_{moy}(G_i, G_j) = \left\| \mu^{(i)} - \mu^{(j)} \right\|;$$

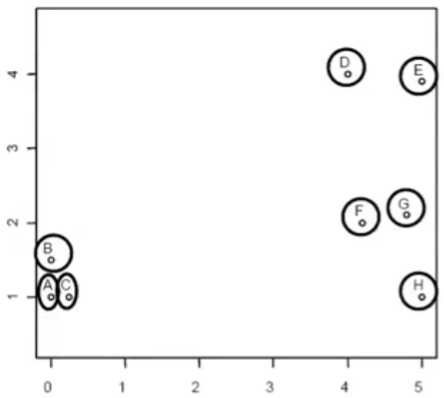
$$dist_{ward}(G_i, G_j) = \frac{N_i N_j}{N_i + N_j} \left\| \mu^{(i)} - \mu^{(j)} \right\|;$$

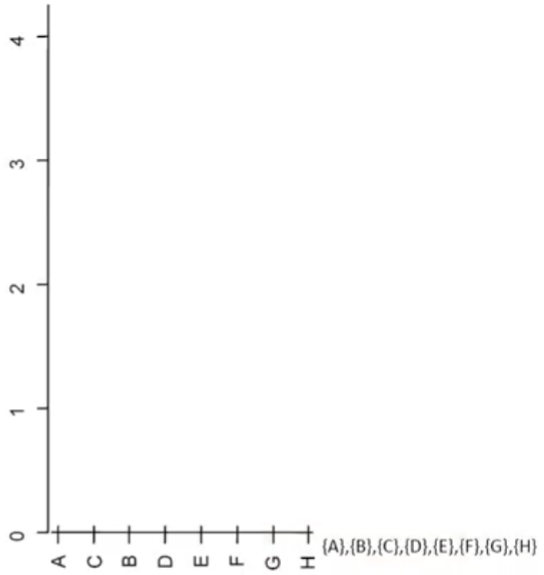
- N_i et N_j sont les cardinalités de G_i et G_j et $\mu^{(i)}$ et $\mu^{(j)}$ les moyennes de leurs données, respectivement.

Algorithme



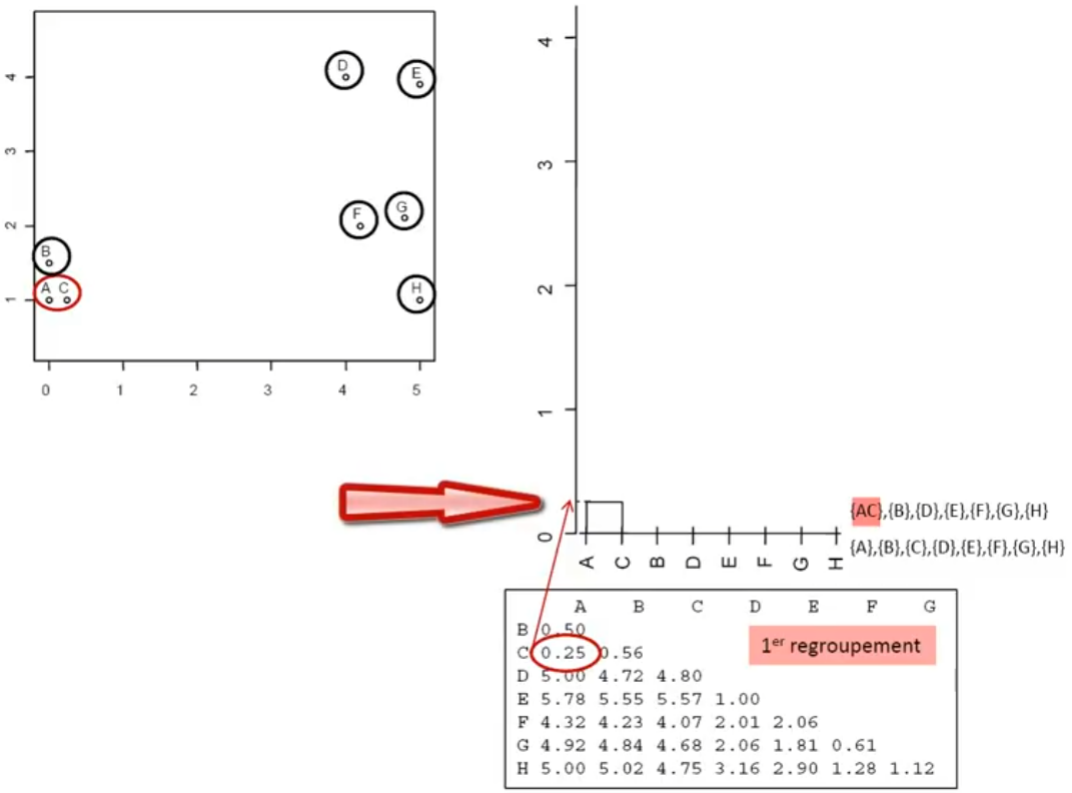
Algorithme



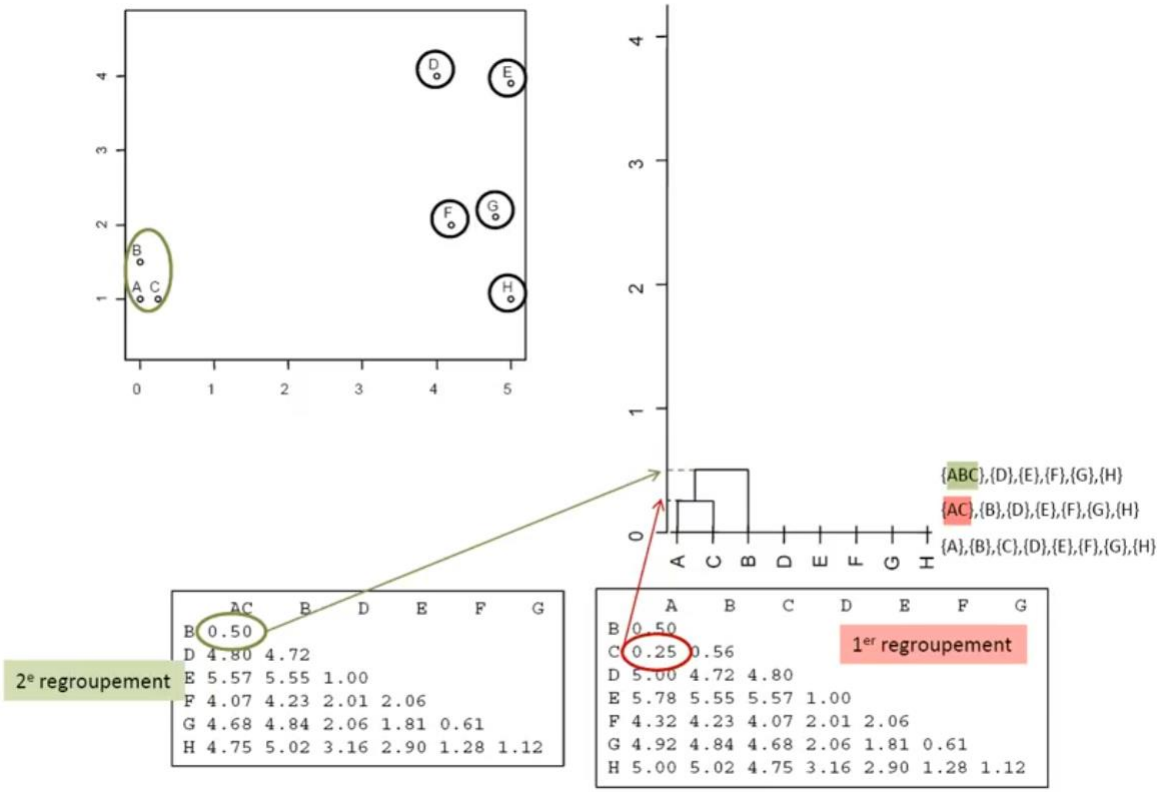


	A	B	C	D	E	F	G
B	0.50						
C	0.25	0.56					
D	5.00	4.72	4.80				
E	5.78	5.55	5.57	1.00			
F	4.32	4.23	4.07	2.01	2.06		
G	4.92	4.84	4.68	2.06	1.81	0.61	
H	5.00	5.02	4.75	3.16	2.90	1.28	1.12

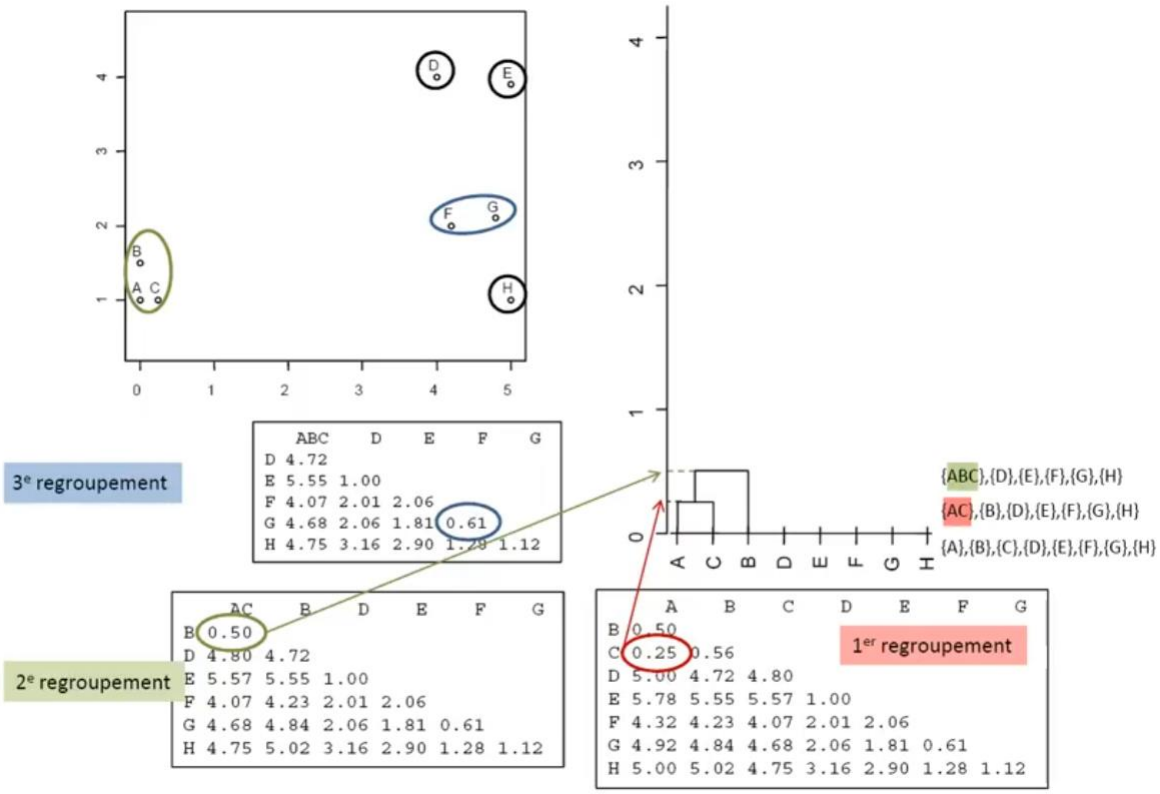
Algorithme


D
E
F
G
B
A C
H
0 1 2 3 4 5
1 2 3 4
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}
A C B D E F G H
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12
1er regroupement

Algorithme


D
E
F
G
B
A C
H
0 1 2 3 4 5
1 2 3 4
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}
A C B D E F G H
2e regroupement
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
1er regroupement
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12

Algorithme

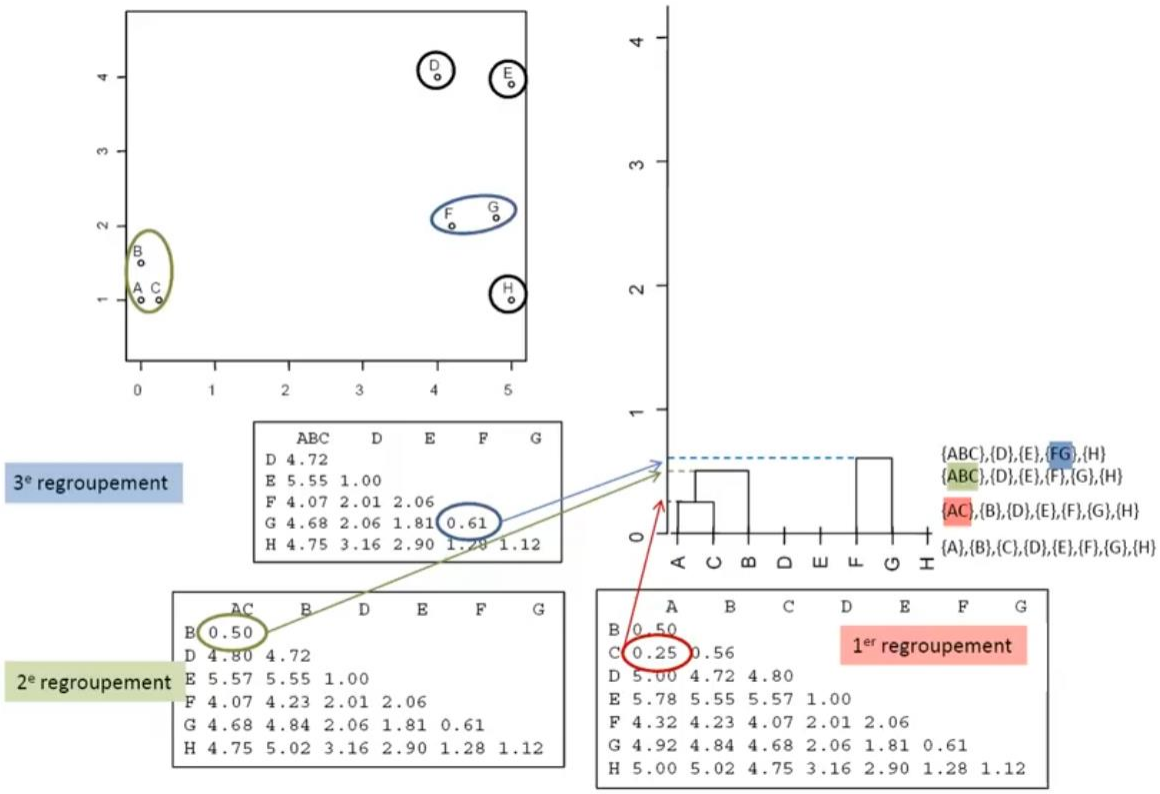


	ABC	D	E	F	G
D	4.72				
E	5.55	1.00			
F	4.07	2.01	2.06		
G	4.68	2.06	1.81	0.61	
H	4.75	3.16	2.90	1.28	1.12

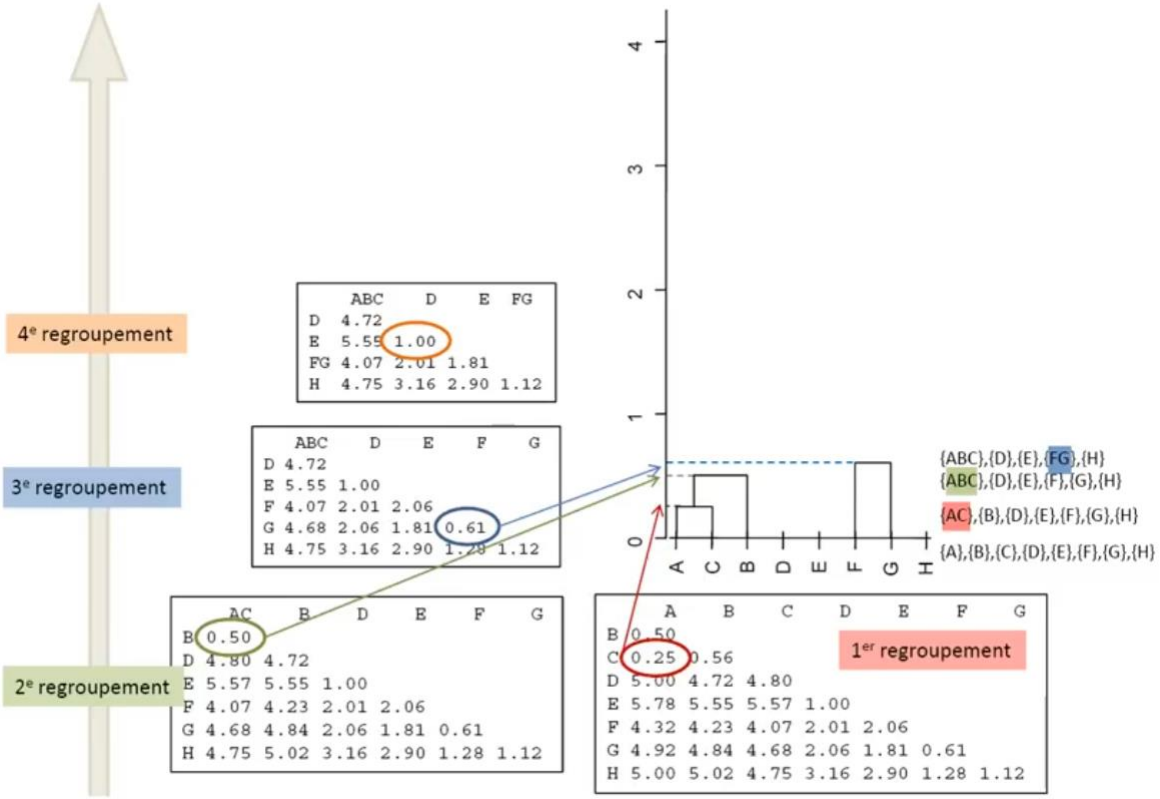
	AC	B	D	E	F	G
B	0.50					
D	4.80	4.72				
E	5.57	5.55	1.00			
F	4.07	4.23	2.01	2.06		
G	4.68	4.84	2.06	1.81	0.61	
H	4.75	5.02	3.16	2.90	1.28	1.12

	A	B	C	D	E	F	G
B	0.50						
C	0.25	0.56					
D	5.00	4.72	4.80				
E	5.78	5.55	5.57	1.00			
F	4.32	4.23	4.07	2.01	2.06		
G	4.92	4.84	4.68	2.06	1.81	0.61	
H	5.00	5.02	4.75	3.16	2.90	1.28	1.12

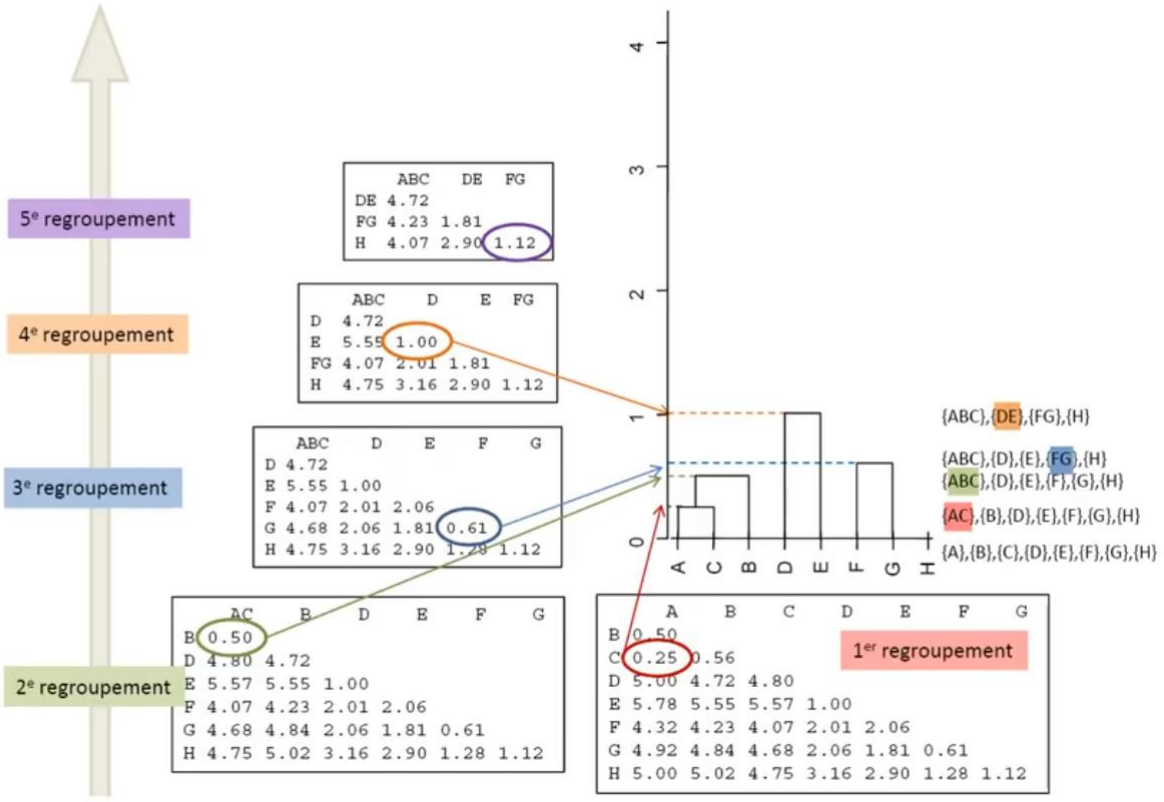
Algorithme


3e regroupement
2e regroupement
1er regroupement
{ABC},{D},{E},{FG},{H}
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}
A C B D E F G H
ABC D E F G
D 4.72
E 5.55 1.00
F 4.07 2.01 2.06
G 4.68 2.06 1.81 0.61
H 4.75 3.16 2.90 1.28 1.12
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12

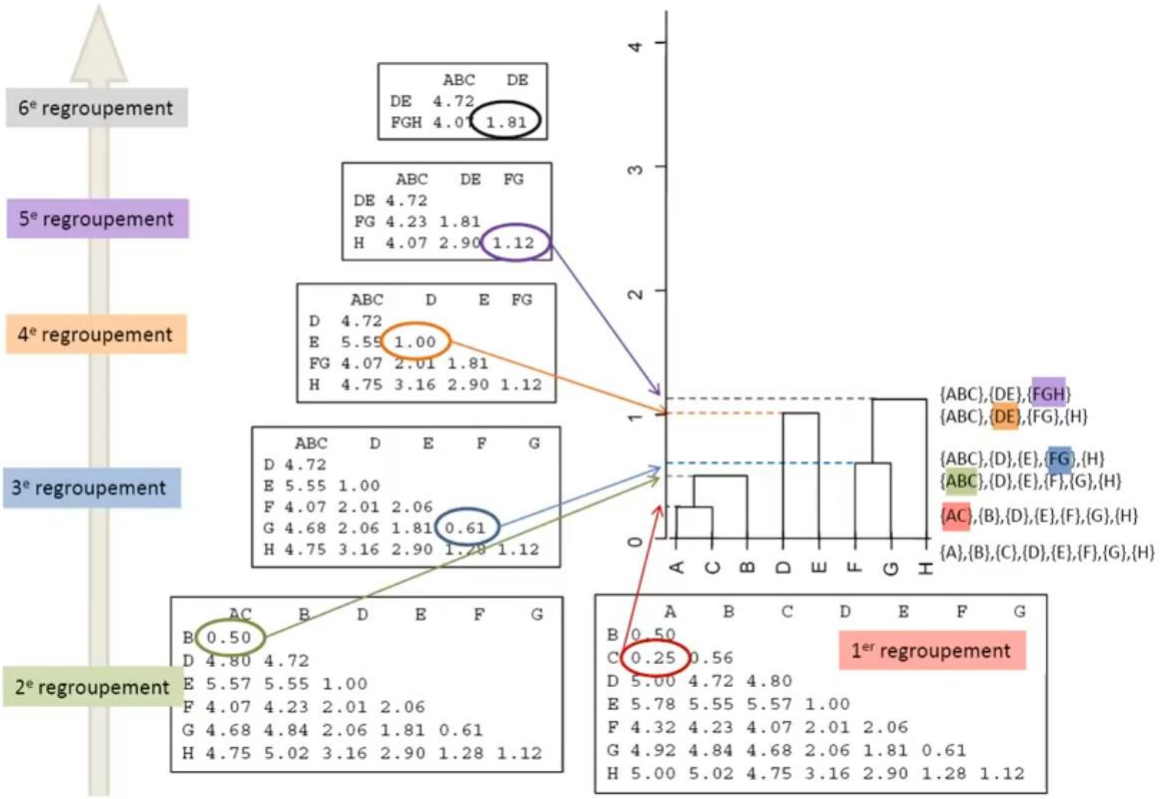
Algorithme


4e regroupement
3e regroupement
2e regroupement
ABC D E FG
D 4.72
E 5.55 1.00
FG 4.07 2.01 1.81
H 4.75 3.16 2.90 1.12
ABC D E F G
D 4.72
E 5.55 1.00
F 4.07 2.01 2.06
G 4.68 2.06 1.81 0.61
H 4.75 3.16 2.90 1.29 1.12
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12
1er regroupement
0 1 2 3 4
A C B D E F G H
{ABC},{D},{E},{FG},{H}
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}

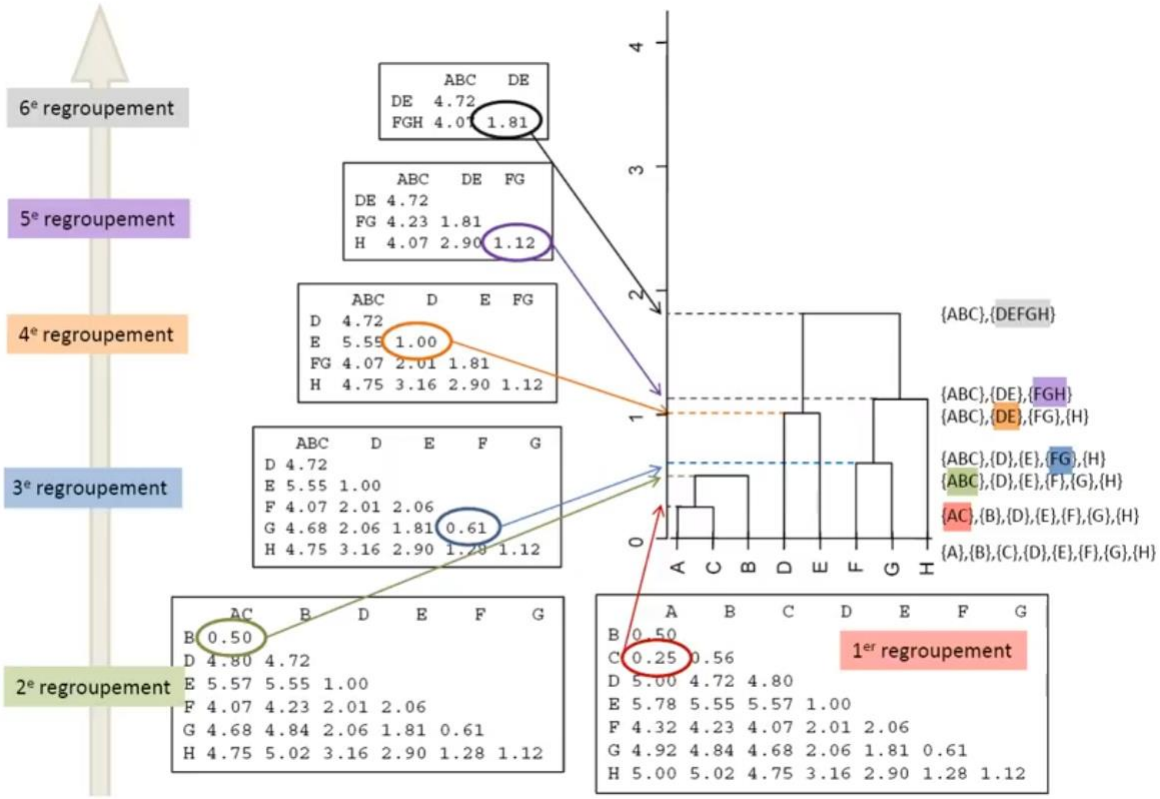
Algorithme


5e regroupement
4e regroupement
3e regroupement
2e regroupement
ABC DE FG
DE 4.72
FG 4.23 1.81
H 4.07 2.90 1.12
ABC D E FG
D 4.72
E 5.55 1.00
FG 4.07 2.01 1.81
H 4.75 3.16 2.90 1.12
ABC D E F G
D 4.72
E 5.55 1.00
F 4.07 2.01 2.06
G 4.68 2.06 1.81 0.61
H 4.75 3.16 2.90 1.28 1.12
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12
1er regroupement
0
1
2
3
4
A C B D E F G H
{ABC},{DE},{FG},{H}
{ABC},{D},{E},{FG},{H}
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}

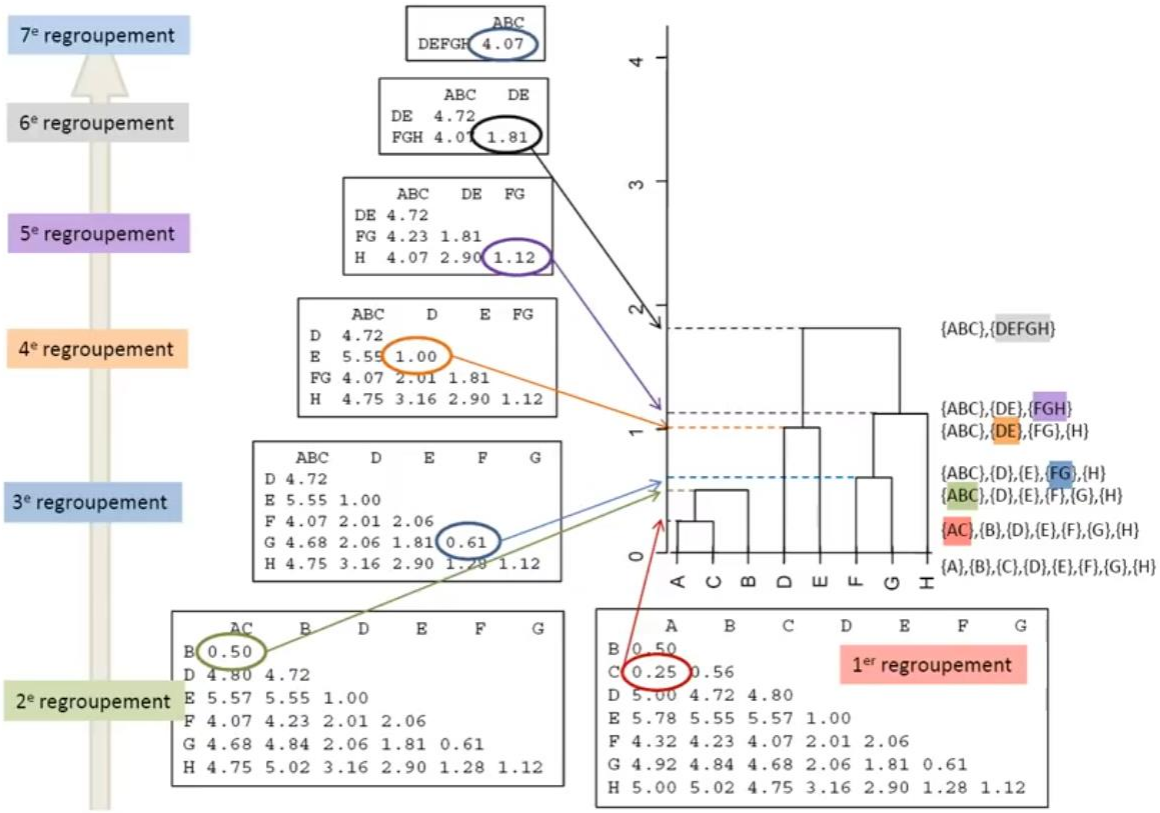
Algorithme


6e regroupement
5e regroupement
4e regroupement
3e regroupement
2e regroupement
ABC DE
DE 4.72
FGH 4.07 1.81
ABC DE FG
DE 4.72
FG 4.23 1.81
H 4.07 2.90 1.12
ABC D E FG
D 4.72
E 5.55 1.00
FG 4.07 2.01 1.81
H 4.75 3.16 2.90 1.12
ABC D E F G
D 4.72
E 5.55 1.00
F 4.07 2.01 2.06
G 4.68 2.06 1.81 0.61
H 4.75 3.16 2.90 1.28 1.12
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12
1er regroupement
0
1
2
3
4
A C B D E F G H
{ABC},{DE},{FGH}
{ABC},{DE},{FG},{H}
{ABC},{D},{E},{FG},{H}
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}

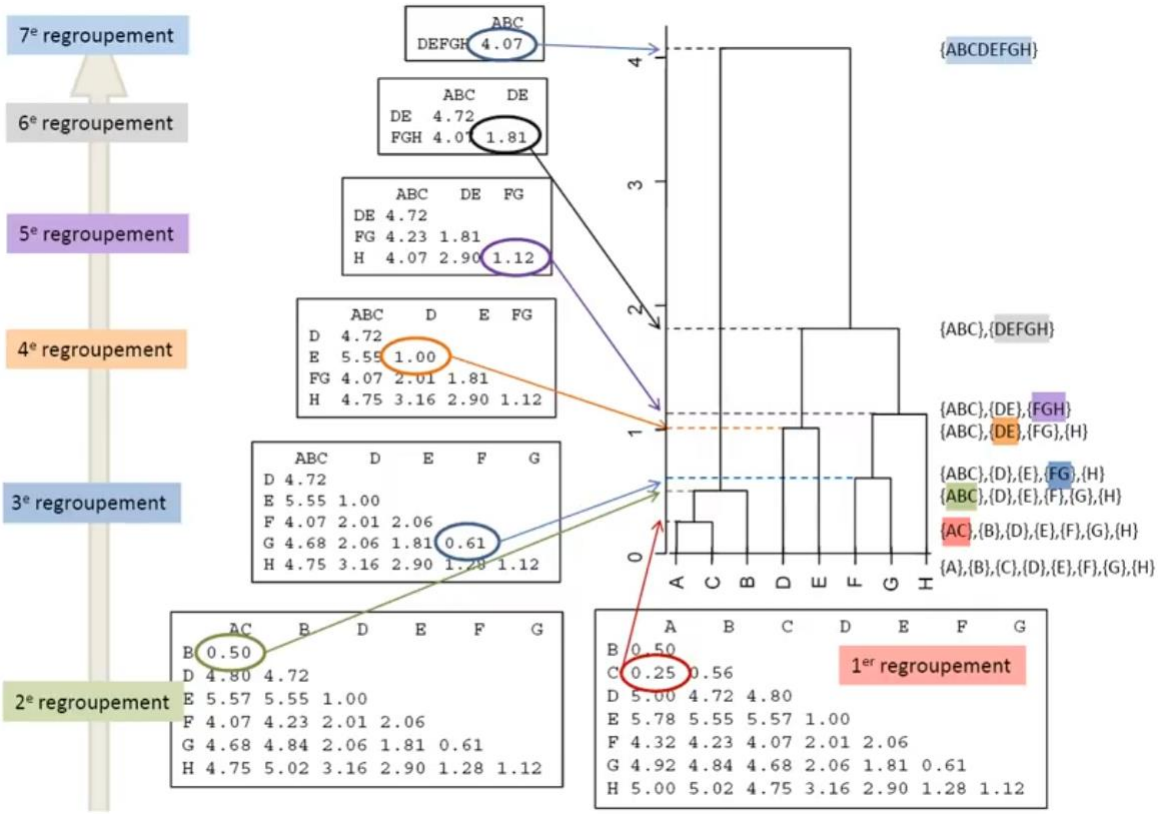
Algorithme


6e regroupement
5e regroupement
4e regroupement
3e regroupement
2e regroupement
1er regroupement
ABC DE
DE 4.72
FGH 4.07 1.81
ABC DE FG
DE 4.72
FG 4.23 1.81
H 4.07 2.90 1.12
ABC D E FG
D 4.72
E 5.55 1.00
FG 4.07 2.01 1.81
H 4.75 3.16 2.90 1.12
ABC D E F G
D 4.72
E 5.55 1.00
F 4.07 2.01 2.06
G 4.68 2.06 1.81 0.61
H 4.75 3.16 2.90 1.79 1.12
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12
0
1
2
3
4
A C B D E F G H
{ABC},{DEFGH}
{ABC},{DE},{FGH}
{ABC},{DE},{FG},{H}
{ABC},{D},{E},{FG},{H}
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}

Algorithme

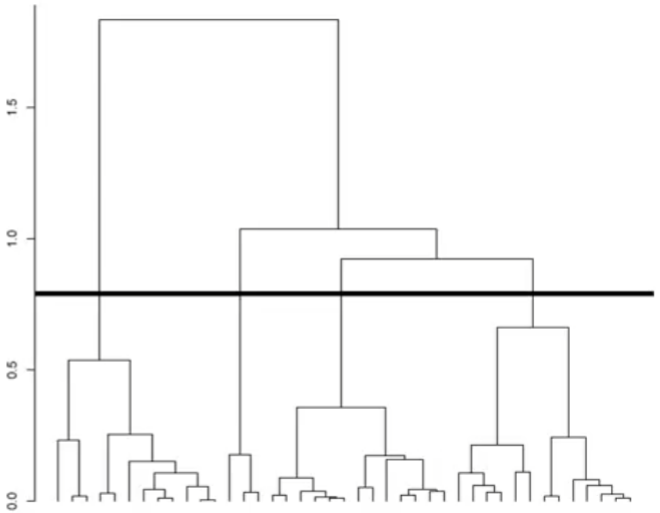

7e regroupement
6e regroupement
5e regroupement
4e regroupement
3e regroupement
2e regroupement
ABC
DEFGH 4.07
ABC DE
DE 4.72
FGH 4.07 1.81
ABC DE FG
DE 4.72
FG 4.23 1.81
H 4.07 2.90 1.12
ABC D E FG
D 4.72
E 5.55 1.00
FG 4.07 2.01 1.81
H 4.75 3.16 2.90 1.12
ABC D E F G
D 4.72
E 5.55 1.00
F 4.07 2.01 2.06
G 4.68 2.06 1.81 0.61
H 4.75 3.16 2.90 1.79 1.12
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12
1er regroupement
0
1
2
3
4
A C B D E F G H
{ABC},{DEFGH}
{ABC},{DE},{FGH}
{ABC},{DE},{FG},{H}
{ABC},{D},{E},{FG},{H}
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}

Algorithme


7e regroupement
6e regroupement
5e regroupement
4e regroupement
3e regroupement
2e regroupement
ABC
DEFGH 4.07
ABC DE
DE 4.72
FGH 4.07 1.81
ABC DE FG
DE 4.72
FG 4.23 1.81
H 4.07 2.90 1.12
ABC D E FG
D 4.72
E 5.55 1.00
FG 4.07 2.01 1.81
H 4.75 3.16 2.90 1.12
ABC D E F G
D 4.72
E 5.55 1.00
F 4.07 2.01 2.06
G 4.68 2.06 1.81 0.61
H 4.75 3.16 2.90 1.29 1.12
AC B D E F G
B 0.50
D 4.80 4.72
E 5.57 5.55 1.00
F 4.07 4.23 2.01 2.06
G 4.68 4.84 2.06 1.81 0.61
H 4.75 5.02 3.16 2.90 1.28 1.12
A B C D E F G
B 0.50
C 0.25 0.56
D 5.00 4.72 4.80
E 5.78 5.55 5.57 1.00
F 4.32 4.23 4.07 2.01 2.06
G 4.92 4.84 4.68 2.06 1.81 0.61
H 5.00 5.02 4.75 3.16 2.90 1.28 1.12
1er regroupement
{ABCDEFGH}
{ABC},{DEFGH}
{ABC},{DE},{FGH}
{ABC},{DE},{FG},{H}
{ABC},{D},{E},{FG},{H}
{ABC},{D},{E},{F},{G},{H}
{AC},{B},{D},{E},{F},{G},{H}
{A},{B},{C},{D},{E},{F},{G},{H}
4
3
2
1
0
A C B D E F G H

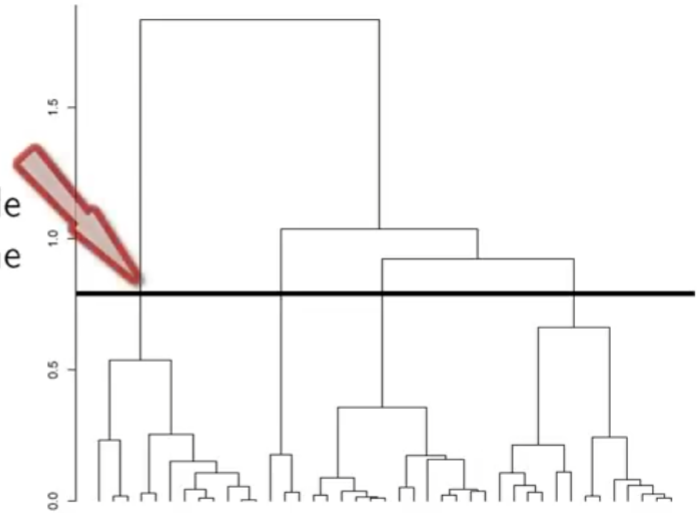
Agglomération et partitions

En définissant un niveau de coupure, on construit une partition



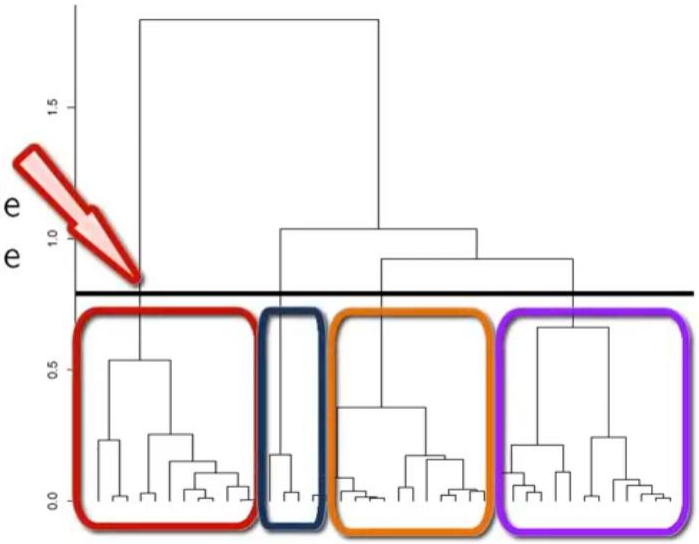
Agglomération et partitions

En définissant un niveau de coupure, on construit une partition



Agglomération et partitions

En définissant un niveau de coupure, on construit une partition



Groupement hiérarchique dans Matlab

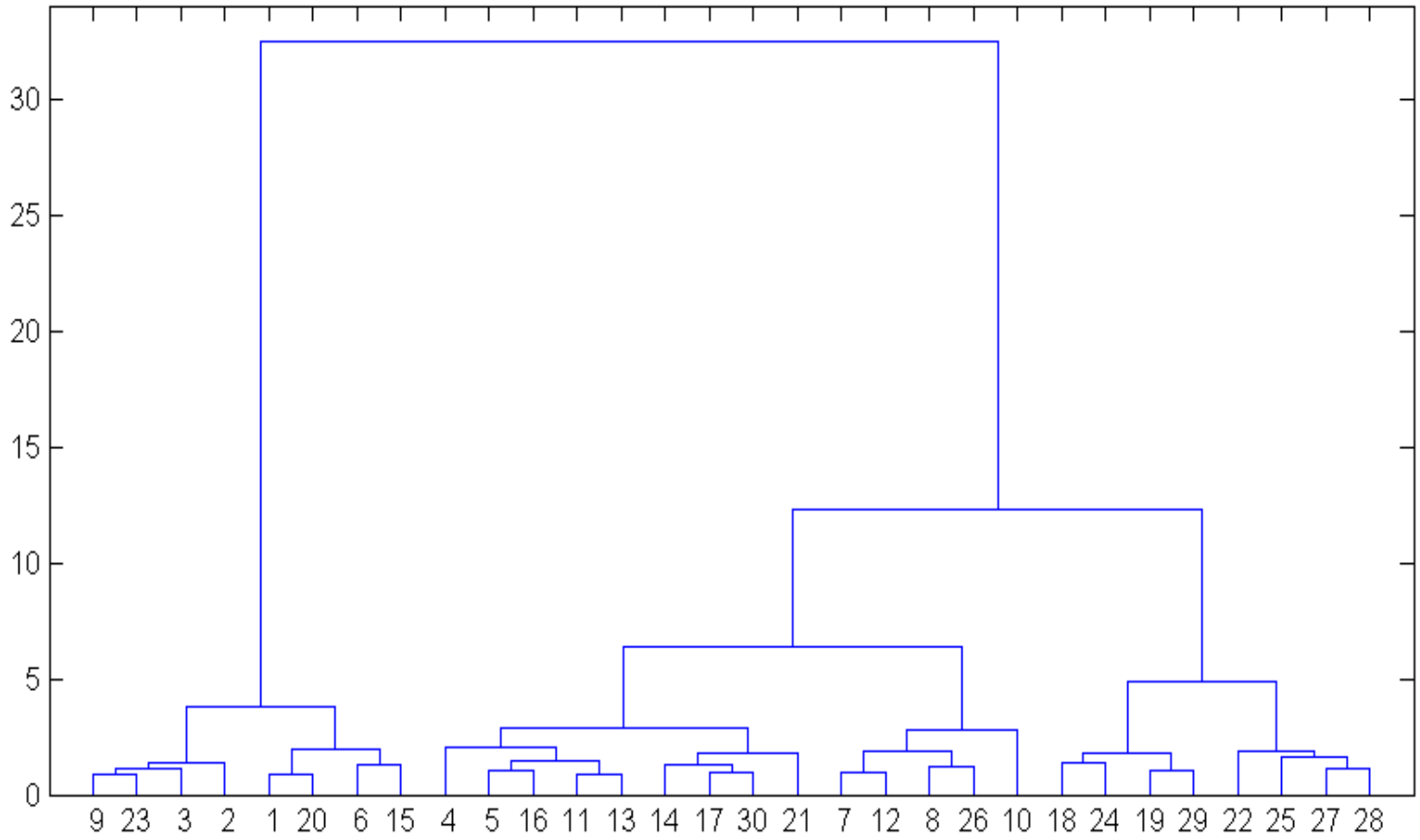
- Matlab offre des fonctions de groupement hiérarchique de données avec la fonction linkage.

  http://www.mathworks.com/help/stats/linkage.html

- En utilisant les données de Iris, par exemple ($N = 150$ et $D = 4$), le groupement hiérarchique obtenu avec la distance $dist_{ward}$ est donné dans la figure ci-après.

- On remarque que puisque Matlab ne peut afficher plus de 30 feuilles par arbre (fonction dendrogram) , il fusionne plusieurs feuilles dans une seule (collapsing).

Groupement hiérarchique dans Matlab



Nombre optimal de groupes

Nombre de groupes

- Dans les algorithmes de groupement présentés, nous avons supposé que le nombre de groupes est connu d'avance.

- Cependant, dans plusieurs situations, on ignore la structure des données et le nombre de groupes est inconnu d'avance.

- L'approche la plus commune est construire une fonction objective $\rho(K)$ et de répéter le groupement pour $K = 1, 2, 3, \dots, etc$.

- La valeur optimale de K serait alors celle qui donnera la meilleure valeur de la fonction objective.

Nombre de groupes pour K-Moyennes

- Soit K groupes $G_1, \ldots, G_K$ résultant d'un algorithme de groupement. Soit N_k le nombre de données du groupe G_k.

- Soit la variable indicatrice r_{ik} pour chaque donnée $x^{(i)}$ et chaque groupe G_k qui possèdera la valeurs suivante:

$$r_{ik} = \begin{cases} 1 & si \quad x^{(i)} \in G_k \\ 0 & si \quad x^{(i)} \notin G_k \end{cases}$$

- Chaque groupe G_k a une moyenne $\mu^{(k)}$ calculée par:

$$\boldsymbol{\mu^{(k)}} = \frac{\sum_{i=1}^{N} r_{ik} x^{(i)}}{\sum_{i=1}^{N} r_{ik}}$$

Nombre de groupes pour K-Moyennes

- Rappelons que pour mesurer le degré de séparation des groupes pour l'algorithme K-moyennes, on minimise la somme des carrés résiduels SCR :

$$SCR = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{ik} \left\| x^{(i)} - \boldsymbol{\mu}^{(k)} \right\|^2$$

- Noter que $SCR \in [0, +\infty]$ et $SCR = 0$ si $K = N$. Donc, la valeur de la SCR n'est pas un bon critère à pour déterminer K.

- On peut construire un critère $\rho(K)$ basé sur SCR comme suit:

Nombre de groupes pour K-Moyennes

$$\rho(K) = SCR + \lambda\, K$$

- La fonction $\rho(K)$ est pénalisée par un grand nombre de groupes K et va s'éloigner de 0.

- Le paramètre λ permet de contrôler le degré de pénalisation des grand nombre de groupes de sorte que.

  ☞ Si λ est grand $\Rightarrow$ on encourage des K petits.

  ☞ Si λ est petit $\Rightarrow$ on encourage des K grands.

- Le choix du K optimal sera établi par: $K^* = argmin_K\, \rho(K)$

Nombre de groupes pour le MDG

- Pour le mélange de distributions Gaussiennes (MDG), le degré d'ajustement du mélange par rapport au données est donné par le maximum der vraisemblance.

$$\ell(\boldsymbol{\varphi}) = \log\left[\prod_{i=1}^{N} p(x^{(i)}|\boldsymbol{\varphi})\right]$$

$$= \sum_{i=1}^{N} \log\left[\sum_{k=1}^{K} p(x^{(i)}|G_k)p(G_k)\right]$$

Où $\boldsymbol{\varphi}$ est l'ensemble de tous les paramètres du mélange.

Nombre de groupes pour le MDG

- Notez que la valeur de $\ell(\boldsymbol{\varphi}) \in [-\infty, 0]$, donc $-\ell(\boldsymbol{\varphi}) \in [0, \infty]$. Comme pour la SCR, la valeur de $-\ell(\boldsymbol{\varphi}) = 0$ indique un ajustement parfait aux données.

- La valeur de $-\ell(\boldsymbol{\varphi})$ pourra être utilisée pour choisir le bon nombre de groupes K.

- Cependant, on peut définir un critère basé sur $-\ell(\boldsymbol{\varphi})$. *Akaike* (1974) a défini un tel critère comme suit:

$$\rho(K) = -2\ell(\boldsymbol{\varphi}) + 2\eta(\boldsymbol{\varphi})$$

- $\eta(\boldsymbol{\varphi})$ est le nombre de paramètres du mélange.

Références

1. M. S. Allili. Techniques d'apprentissage automatique (Cours de 2e cycle). Université du Québec en Outaouais (UQO), Québec, Canada. Hivers 2015.
2. S. Rogers et M Girolami. A first Course in machine learning, CRC press, 2012.
3. C. Bishop. Pattern Recognition and Machine learning. Springer 2006.
4. R. Duda, P. Storck et D. Hart. Pattern Classification. Prentice Hall, 2002.