

COURS MASTER IFI 2010/2011 N°3



JEAN-DANIEL ZUCKER

DR À L'IRD UR GEODES
(MODÉLISATION MATHÉMATIQUES ET INFORMATIQUES DES SYSTÈMES COMPLEXES)
UMMISCO UMI 209



Administratif: 1/2 Module Apprentissage (18ECTS)

- **Séance 1: Algorithme Elimination des candidats** – INTRO GÉNÉRAL
- **Séance 2: Algo. Arbre de décision** – APPRENTISSAGE SUPERVISÉ
- **Séance 3: Lundi 13 Décembre** – APPRENTISSAGE NON-SUPERVISÉ
 - un problème mal posé qui joue un rôle clef en IA
 - Algo CAH : Classification Ascendante hiérarchique
 - Algo Kmeans : k-moyennes
- **Séance 4: Mardi 11 Janvier 2011** – ALGORITHMES ÉVOLUTIONNAIRES
- **Séance 5: Jeudi 14 Janvier 2011** – ALGORITHMES PAR RENFORCEMENT
- **Séance 6: Lundi 17 Janvier 2011** – MINI-PROJET

LE CLUSTERING

- Utilisé dans de très nombreux domaines.
- But : rassembler les éléments (gènes) en groupes :
 - *Homogènes*
 - » Les éléments dans un groupe sont *aussi similaires que possible*
 - *Séparés*
 - » Les éléments de différents groupes sont *aussi différents que possible*

I. LE CLUSTERING

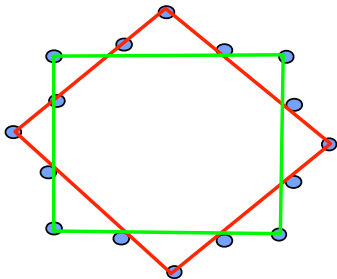
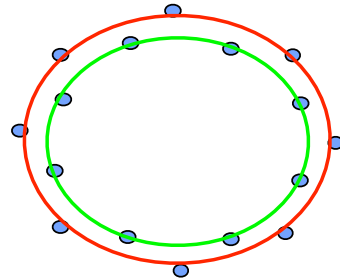
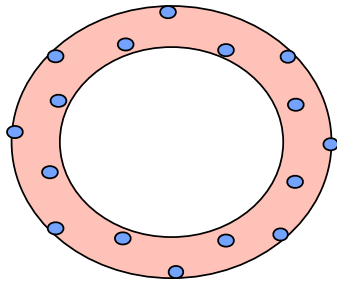
I.1) UN PROBLÈME MAL POSÉ

I.2) FAMILLE DE MÉTHODES

I.3) IMPORTANCE DE LA DISTANCE

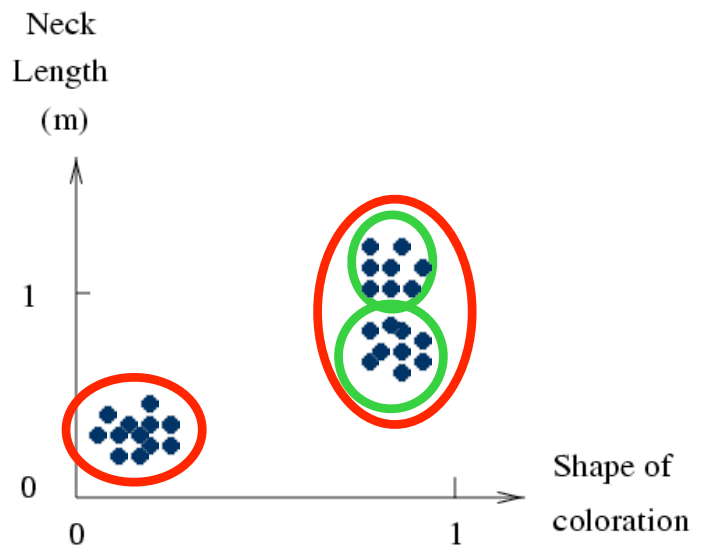
I.4) ALGORITHMES

LE "CLUSTERING" EST UN PROBLÈME MAL POSE...

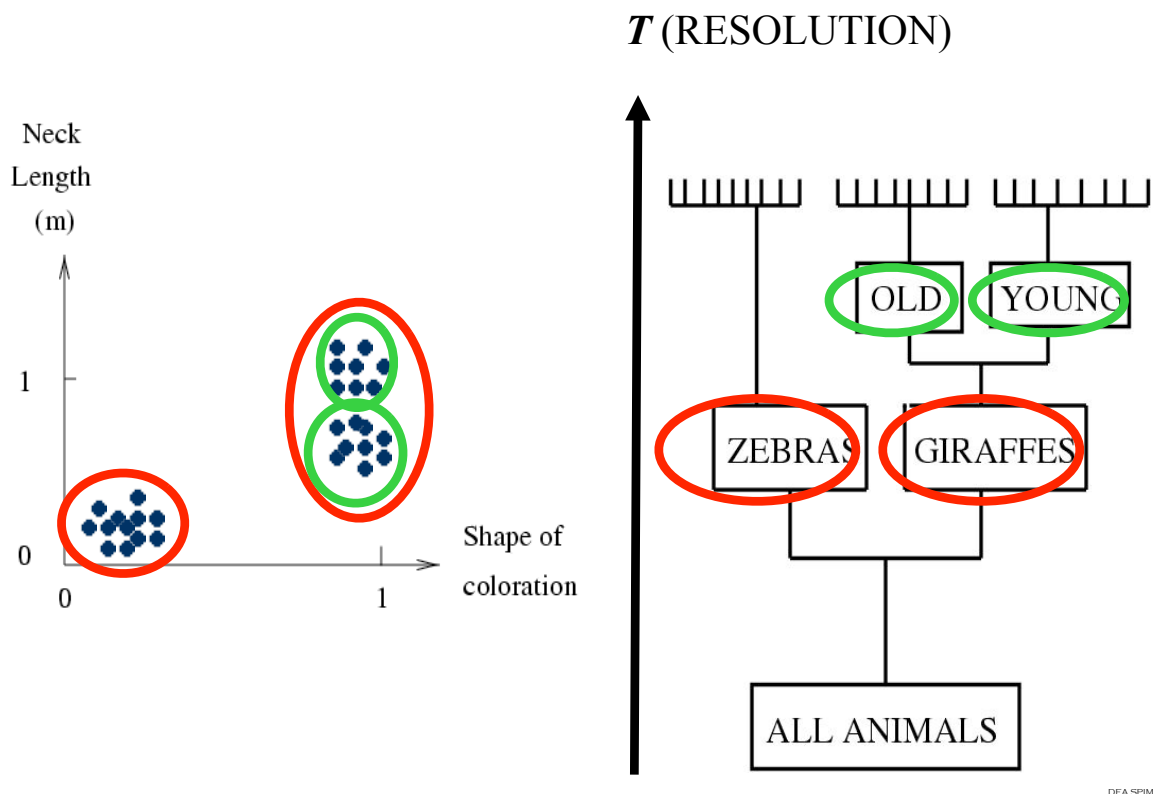


... beaucoup dépend de la représentation du problème et des distances utilisées.

EXEMPLE DE CLUSTERING

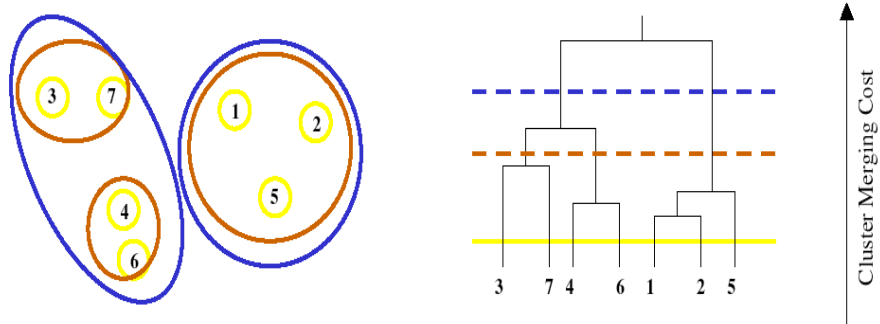


NOTION DE DENDROGRAMME



DENDROGRAMME (2/2)

- Input: Data Points, x_1, x_2, \dots, x_n
- Output: Tree
 - the data points are leaves
 - Branching points indicate similarity between sub-trees
 - Horizontal cut in the tree produces data clusters



(RE)DÉFINITION DU PROBLÈME

- Étant donnée une collection de n “points” $X_i, i=1,2,\dots,n$, définis dans un espace de dimension d , identifier des structures dans les données.
- Exemple : **partitionner** les données en **M groupes** (clusters), tels que les points d’un cluster soient **les plus “similaires”**
 - Identifier des groupes stables
 - Générer des dendrogrammes
- **Problème mal-posé :**
 - Valeur de M ?
 - Notion de similarité ?

I. LE CLUSTERING

I.1) UN PROBLÈME MAL POSÉ

I.2) FAMILLE DE MÉTHODES

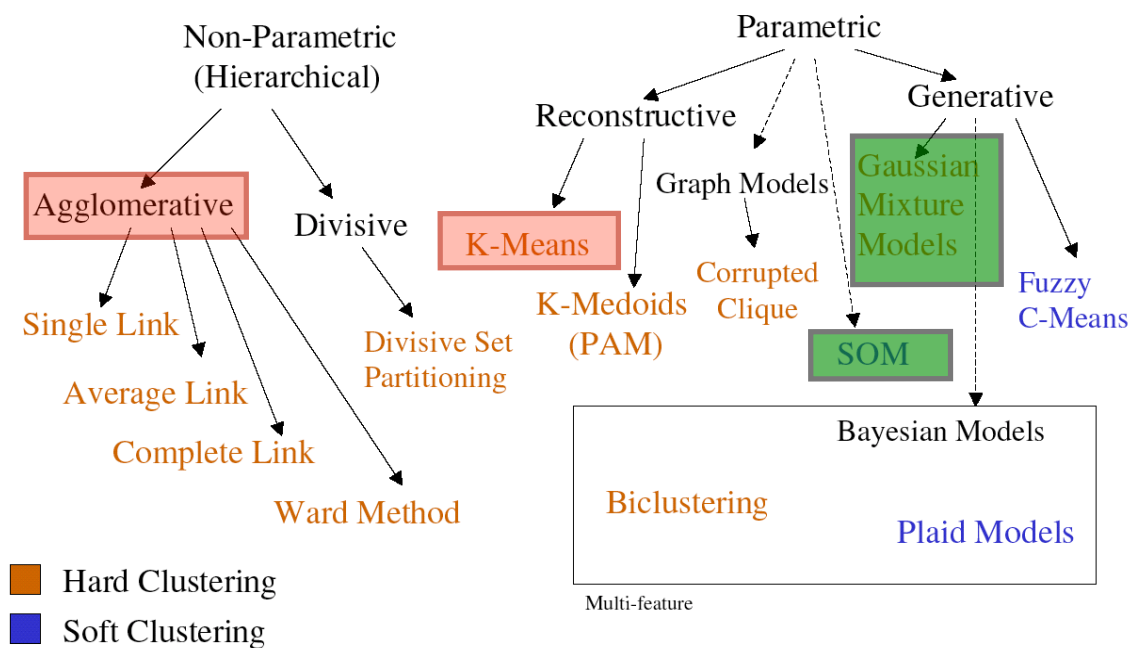
I.3) IMPORTANCE DE LA DISTANCE

I.4) ALGORITHMES

FAMILLES DE MÉTHODES

- **Méthodes de partitionnement**
- **Méthodes de « classification » hiérarchique**
 - *Agglomératives* (ascendantes)
 - » Rassemblement d'éléments d'un niveau pour construire un groupe au niveau supérieur
 - *Par division* (descendantes)
 - » Division des groupes d'un niveau en sous-groupes du niveau inférieur

TAXONOMIE DES MÉTHODES



I. LE CLUSTERING

I.1) UN PROBLÈME MAL POSÉ

I.2) FAMILLE DE MÉTHODES

I.3) IMPORTANCE DE LA DISTANCE

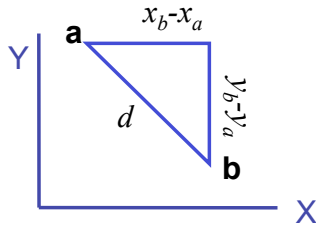
I.4) ALGORITHMES

TYPES DE MESURES DE (DI)SIMILARITÉ

- Un paramètre crucial est la mesure de la similarité ou de dissimilarité entre objets
- Pour en citer quelques unes:
 - **Euclidian distance**
 - Manhattan distance
 - **Pearson's coefficient of correlation**
 - Mahalanobis distance
 - χ^2 distance
- Ce choix dépend des données...

DISTANCE EUCLIDIENNE

- Vous vous souvenez ...:



$$d_E = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

- Cela s'étend naturellement à des espaces de dimensions p

$$d_E = \sqrt{\sum_{i=1}^p (x_{ai} - x_{bi})^2}$$

- Deux applications typiques
 - La distance entre gènes est calculée dans l'espace des conditions (puces)
 - La distance entre types de tissus est calculée dans l'espace des gènes (spot)

LA NORME L_p (GÉNÉRALISE LA DISTANCE EUCLIDIENNE)

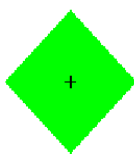
The L_p norm

$$d(x, y) = \sqrt[p]{|x_1 - y_1|^p + \dots + |x_n - y_n|^p}$$

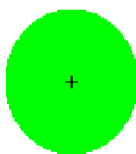
$p = 2$, Euclidean Dist.

$p = \infty$, Manhattan Dist. (downtown Davis distance)

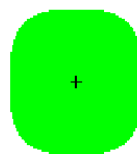
Equidistant points from a center, for different norms



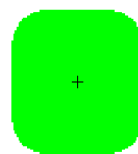
$p=1$



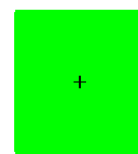
$p=2$



$p=3$



$p=4$



$p=20$

DE LA CORRELATION À UNE DISTANCE

- Soit deux objets :

$$r_p = \frac{\sum_{i=1}^N (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}}$$

Si on pose $\cos(\alpha) = r_p$

Le coefficient de corrélation n'est autre que le cosinus entre les deux vecteurs centrés !

Si $r = 1$, l'angle $\alpha = 0$, les deux vecteurs sont colinéaires (parallèles).

Si $r = 0$, l'angle $\alpha = 90^\circ$, les deux vecteurs sont orthogonaux.

Si $r = -1$, l'angle α vaut 180° , les deux vecteurs sont colinéaires de sens opposé.

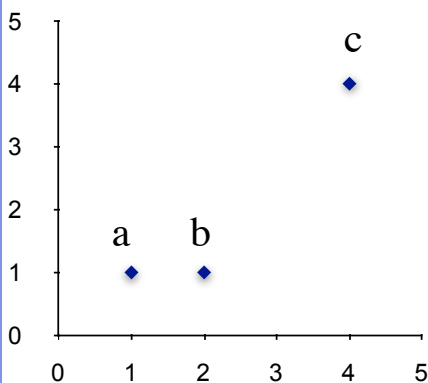
Plus généralement : $\alpha = \arccos(r)$, où \arccos est la réciproque de la fonction cosinus.

- Peut être convertit en une distance :

$$d_p = 1 - c_p$$

IMPACT OF THE DISTANCE METRICS

A



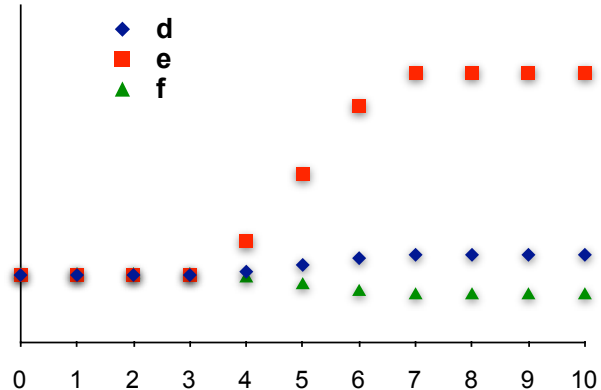
Euclidian distances

- a close to b

Correlation coefficient

- a close to c

B



Euclidian distances

- d close to f

Correlation coefficient

- d close to e

I. LE CLUSTERING

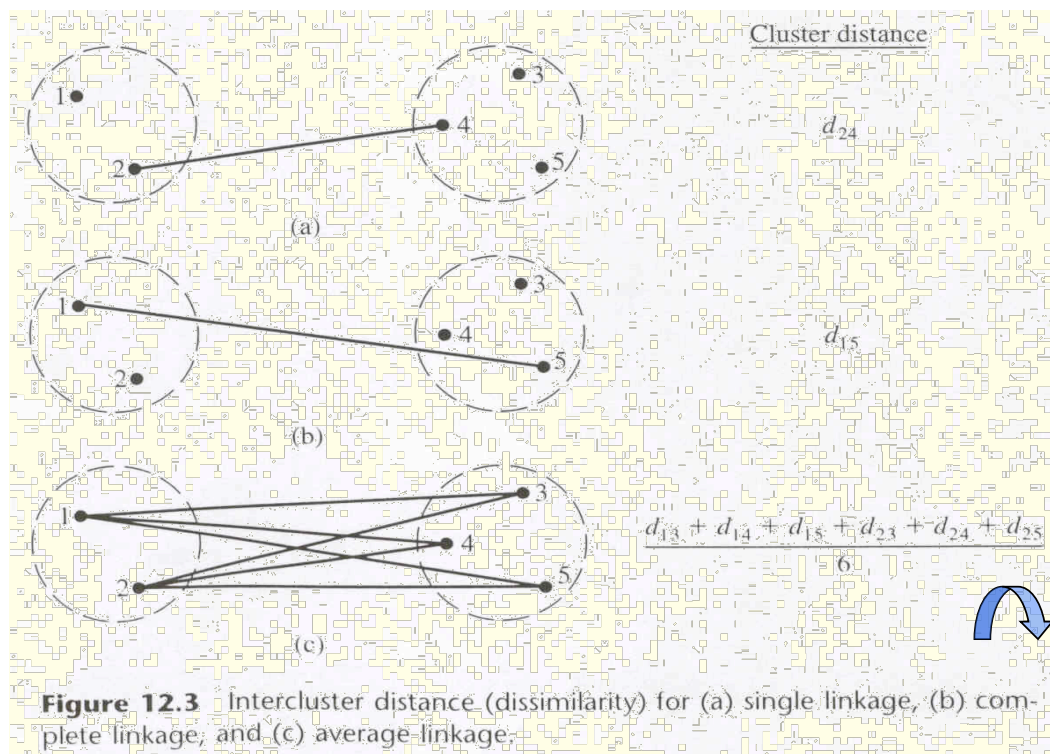
I.1) UN PROBLÈME MAL POSÉ

I.2) FAMILLE DE MÉTHODES

I.3) IMPORTANCE DE LA DISTANCE

I.4) ALGORITHMES : CAH

Algorithme Hierarchique (principe 1/)



Algorithme Hierarchique (principe 2/)

- **Single linkage method**

$$\min(d_{ij})=d_{35}=2$$

$$D = \{d_{ij}\} =$$

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

Algorithme Hierarchique (principe 3/)

- **Single linkage method**

$$d_{(35)1} = \min(d_{31}, d_{51}) = \min(3, 11) = 3$$

$$d_{(35)2} = \min(d_{32}, d_{52}) = \min(7, 10) = 7$$

$$d_{(35)4} = \min(d_{34}, d_{54}) = \min(9, 8) = 8$$

$$d_{(135)2} = \min[d_{(35)2}, d_{12}] = \min(7, 9) = 7$$

$$d_{(135)4} = \min[d_{(35)4}, d_{14}] = \min(8, 6) = 6$$

	(35)	1	2	4
(35)	0			
1	3	0		
2	7	9	0	
4	8	6	5	0

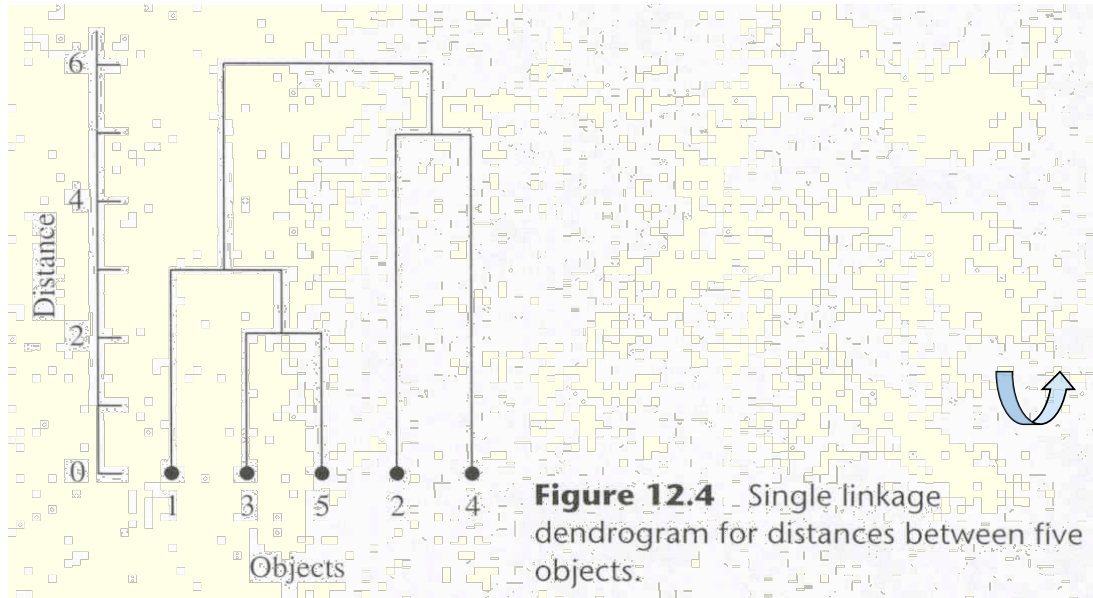
	(135)	2	4
(135)	0		
2	7	0	
4	6	5	0

Algorithme Hierarchique (principe 4/)

- **Single linkage method**

$$d_{(135)(24)} = \min[d_{(135)2}, d_{(135)4}] = \min(7, 6) = 6$$

$$\begin{array}{cc} & (135) & (24) \\ (135) & \begin{bmatrix} 0 & \\ & \end{bmatrix} \\ (24) & \begin{bmatrix} 6 & \\ & 0 \end{bmatrix} \end{array}$$



DIFFÉRENTS TYPES D'ALGORITHMES

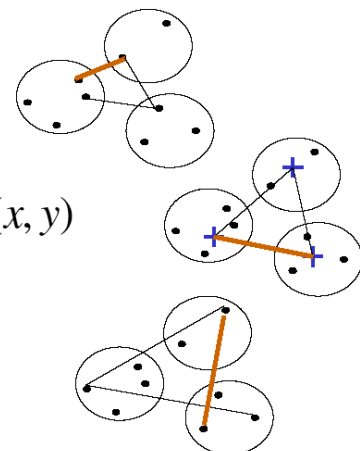
- ... en fonction des coûts d'agglomération

- Single Link, $\min_{x \in C_i, y \in C_j} d(x, y)$

- Average Link, $\frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$

- Complete Link, $\max_{x \in C_i, y \in C_j} d(x, y)$

- Others (Ward method-least squares)



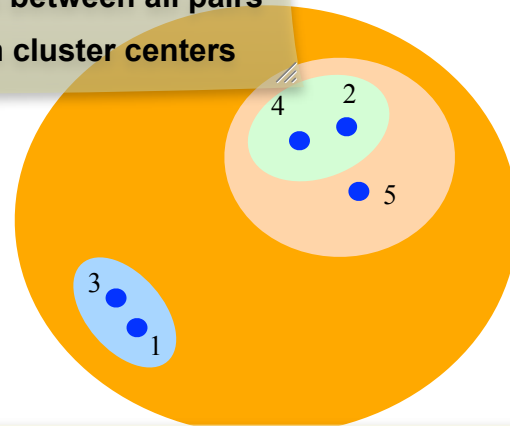
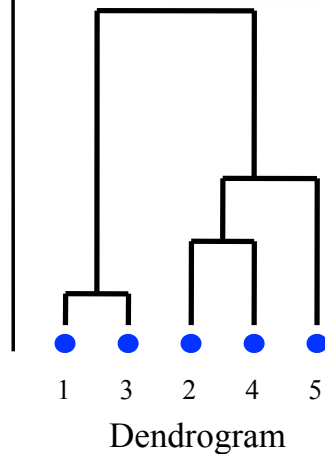
AGGLOMERATIVE HIERARCHICAL CLUSTERING

Need to define the **distance** between the **new cluster** and the **other clusters**.

Single Linkage: distance between closest pair.

Complete Linkage: distance between farthest pair.

Average Linkage: average distance between all pairs
or distance between cluster centers



The dendrogram induces a **linear ordering** of the data points

CLASSIFICATION HIÉRARCHIQUE ET R

```
# Suppression des variables préalablement créées
rm(list=ls(all=TRUE))
# Chargement des données USArrests
data(USArrests)
# Sélection d'une partie de la base
mat <- USArrests[-c(20:50),]

# Calcul de la matrice de distances
dd <- dist(mat)
# Affichage de la matrice de distances
dd

# Application de l'algorithme de classification hiérarchique
hc <- hclust(dd, "average")
# Visualisation du résultat
plot(hc, hang=-1)
```

EVALUATION DE LA QUALITÉ

```
# a 2-dimensional example
x <- rbind(matrix(rnorm(100, mean = 0, sd = 0.5), ncol = 2),
           matrix(rnorm(100, mean = 1, sd = 0.5), ncol = 2))
colnames(x) <- c("x", "y")
# Compute Kmeans with k=3 and 1 iteration
(cl <- kmeans(x, 3, 1))
plot(x, col = cl$cluster)
#Print a star to show centers at iteration 1
points(cl$centers, col = 1:3, pch = 8, cex=2)

# Compute Kmeans with k=3 and 2nd iteration
(cl <- kmeans(x, 3, 2))
#Print a crossed square to show centers at iteration 2
points(cl$centers, col = 1:3, pch = 7, cex=2)
```

HIERARCHICAL CLUSTERING -SUMMARY

- Results depend on distance update method
- Greedy iterative process
- NOT robust against noise
- No inherent measure to identify stable clusters

I. LE CLUSTERING

I.1) UN PROBLÈME MAL POSÉ

I.2) FAMILLE DE MÉTHODES

I.3) IMPORTANCE DE LA DISTANCE

I.4) ALGORITHMES : PARTITIONNEMENT KMEANS

II. LES RÈGLES D'ASSOCIATIONS

MÉTHODES DE PARTITIONNEMENT

Partition the data into a **prespecified number** k of mutually exclusive and exhaustive groups.

Iteratively reallocate the observations to clusters until some criterion is met, e.g. minimize within cluster sums of squares.

Examples :

- k -means, self-organizing maps (SOM), PAM, etc.;
- Fuzzy: needs stochastic model, e.g. Gaussian mixtures.

LA MÉTHODE DES K-MEANS

- **K-means clustering is a variant of clustering around mobile centres**
- **After each assignation of an element to a centre, the position of this centre is re-calculated**
- **The convergence is much faster than with the basic mobile centre algorithm**
 - **after 1 iteration, the result might already be stable**
- **K-means is time- and memory-efficient for very large data sets (e.g. thousands of objects)**

K-MEANS CLUSTERING - SUMMARY

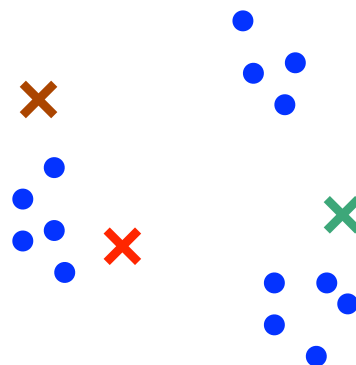
- **Strengths**
 - Simple to use
 - Fast
 - Can be used with very large data sets
- **Weaknesses**
 - The choice of the number of groups is arbitrary
 - The results vary depending on the initial positions of centres
 - The R implementation is based on Euclidian distance, no other metrics are proposed
- **Solutions**
 - Try different values for k and compare the result
 - For each value of k, run repeatedly to sample different initial conditions
- **Weakness of the solution**
 - Instead of one clustering, you obtain hundreds of different clusterings, how to decide among them

CLUSTERING AROUND MOBILE CENTRES

- The number of centres (k) has to be specified a priori
- Algorithm
 - (1) Arbitrarily select k initial centres
 - (2) Assign each element to the closest centre
 - (3) Re-calculate centres (mean position of the assigned elements)
 - (4) Repeat (2) and (3) until one of the stopping conditions is reached
 - » the clusters are the same as in the previous iteration
 - » the difference between two iterations is smaller than a specified threshold
 - » the max number of iterations has been reached

MÉTHODE DU CENTROÏDE

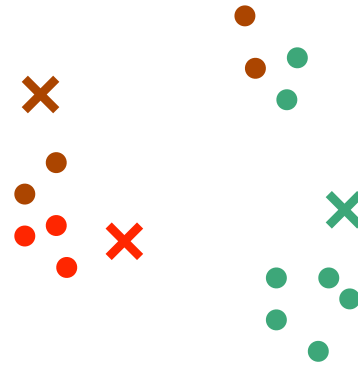
- Start with random positions of centroids.



Iteration = 0

MÉTHODE DU CENTROÏDE

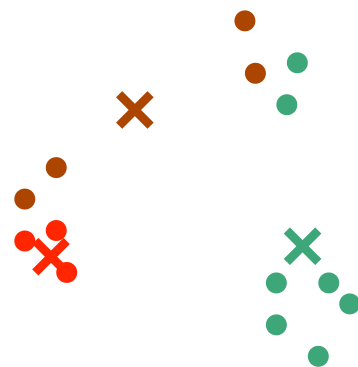
- Start with random positions of centroids.
- Assign data points to centroids



Iteration = 1

MÉTHODE DU CENTROÏDE

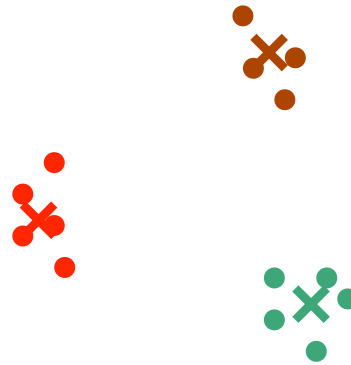
- Start with random positions of centroids.
- Assign data points to centroids
- Move centroids to center of assigned points



Iteration = 1

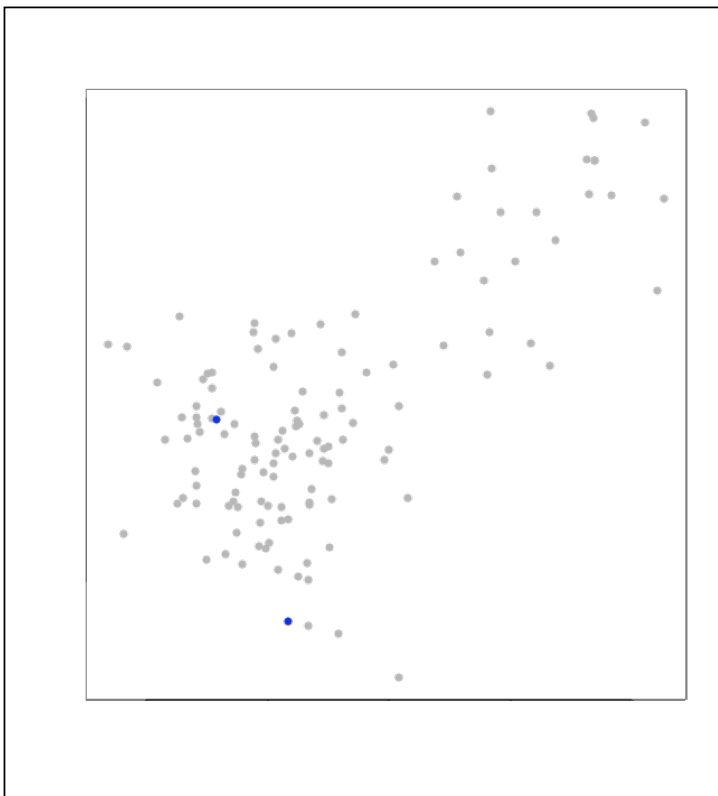
MÉTHODE DU CENTROÏDE

- Start with random positions of centroids.
- Assign data points to centroids
- Move centroids to center of assigned points
- Iterate till minimal **cost**



Iteration = 3

MOBILE CENTRES EXAMPLE - INITIAL CONDITIONS

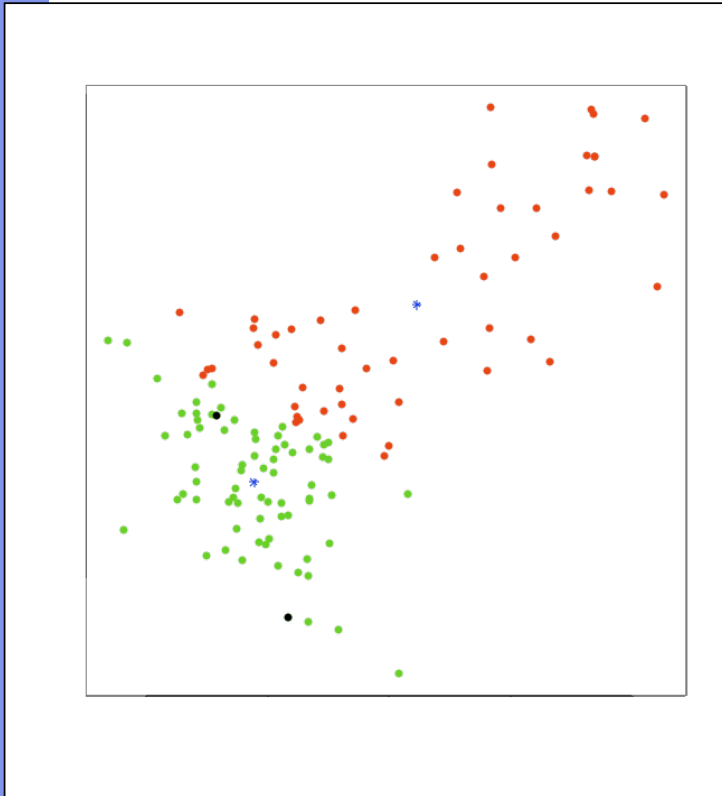


A set of 125 random points was generated

Two points are randomly chosen as seeds (**blue**)

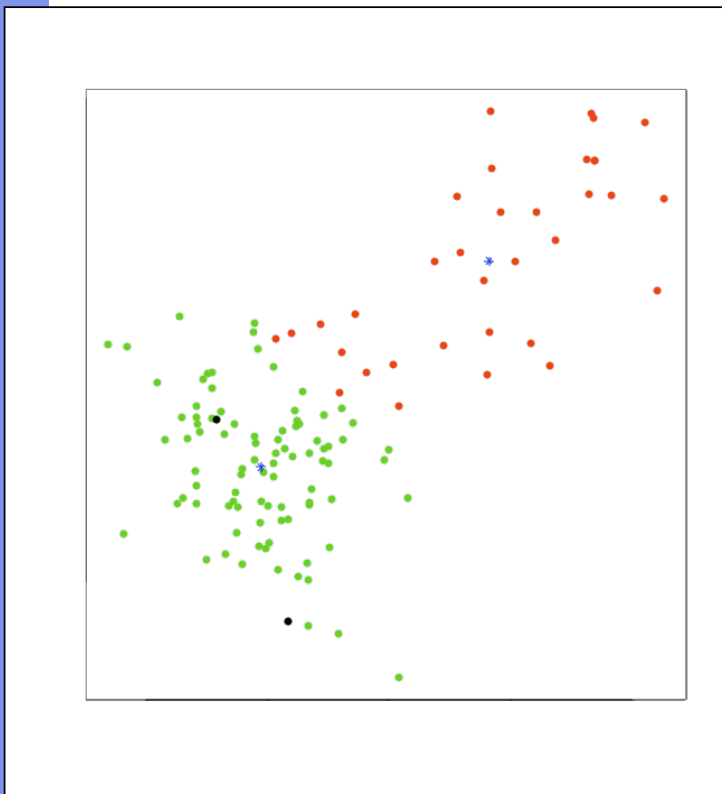
- At each step,
 - points are re-assigned to clusters
 - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

MOBILE CENTRES EXAMPLE - AFTER 3 ITERATIONS



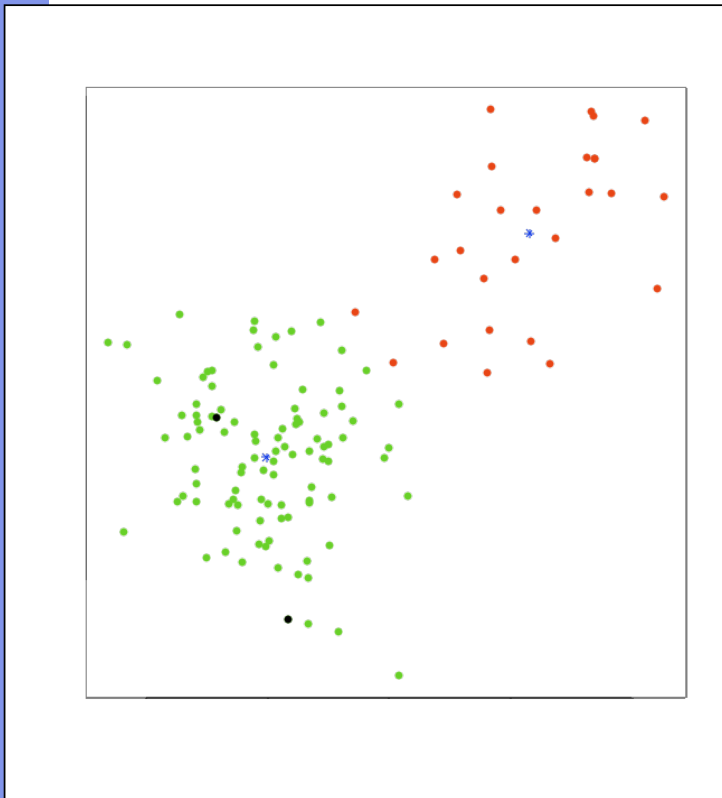
- **At each step,**
 - points are re-assigned to clusters
 - centres are re-calculated
- **Cluster boundaries and centre positions evolve at each iteration**

MOBILE CENTRES EXAMPLE - AFTER 4 ITERATIONS



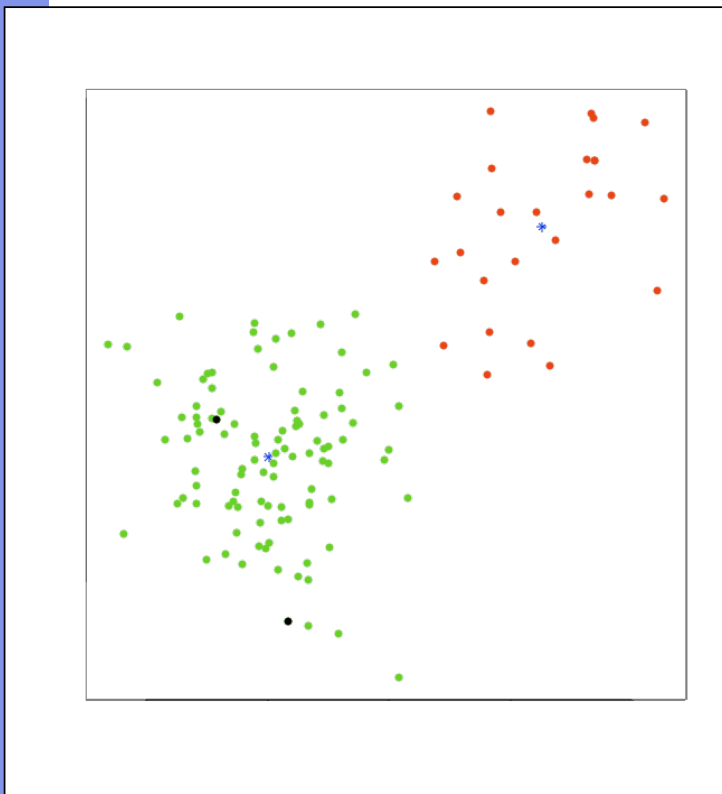
- **At each step,**
 - points are re-assigned to clusters
 - centres are re-calculated
- **Cluster boundaries and centre positions evolve at each iteration**

MOBILE CENTRES EXAMPLE - AFTER 5 ITERATION



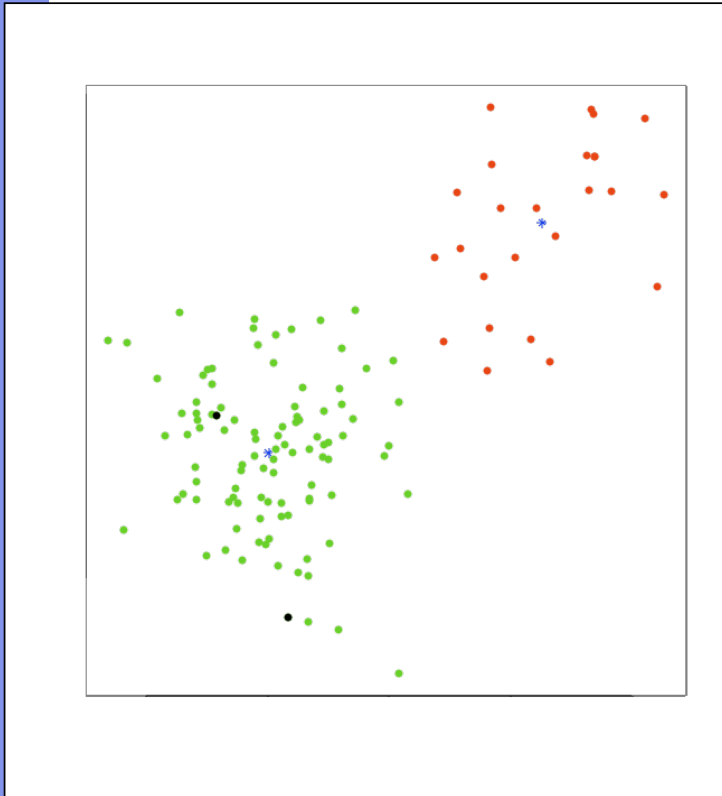
- **At each step,**
 - points are re-assigned to clusters
 - centres are re-calculated
- **Cluster boundaries and centre positions evolve at each iteration**

MOBILE CENTRES EXAMPLE - AFTER 6



- **At each step,**
 - points are re-assigned to clusters
 - centres are re-calculated
- **Cluster boundaries and centre positions evolve at each iteration**

MOBILE CENTRES EXAMPLE - AFTER 7



- After 7 iterations, the clusters and centres are identical to the result from the previous iteration

MÉTHODE DU CENTROÏDE : RÉSUMÉ

- Result depends on **initial** centroids' position
- **Fast** algorithm: compute distances from data points to centroids
- Must preset K
- Fails for non-spherical distributions

K-MEANS : BILAN

- Force des k-means:
 - Relativement efficaces : $O(tkn)$, où n est le nb d'objets, k est le nb de clusters, et t est le nb d'itérations. Normalement, $k, t \ll n$
 - Terminent souvent dans un optimum GLOBAL
- Faiblesse des k-means:
 - Besoin de préciser k à l'avance
 - Sensibles aux données bruitées et aux exceptions, aux cas aberrants
 - Sensibles à l'initialisation
 - » lancer plusieurs exécutions avec différents états initiaux
 - » retenir la configuration jugée la meilleure
 - marche mal lorsque les groupes se chevauchent => variante : K-means flou
- Les variantes des K-Means diffèrent dans :
 - La sélection des k initiaux
 - Calcul de la dissimilarité
 - Stratégies pour calculer la moyenne d'un cluster

K-MEANS ET R

```
# Suppression des variables préalablement créées
rm(list=ls(all=TRUE))
# Création d'une matrice de données artificielles contenant deux sous-populations
c1 <- matrix(rnorm(100, sd = 0.3), ncol = 2)
c2 <- matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2)
mat <- rbind(c1, c2)

# Affichage des points de c1 et c2
plot(c1, col="blue", pch=16, xlim=range(mat[,1]), ylim=range(mat[,2]))
points(c2, col="green", pch=16)

# Application de l'algorithme des kmeans
cl <- kmeans(mat, 2, 20)
# Affichage du résultat de kmeans
cl

# Visualisation du résultat des kmeans sur le graphique
points(cl$centers, col=1:2, pch = 7, lwd=3)
segments( mat[cl$cluster==1,][,1], mat[cl$cluster==1,][,2],
          cl$centers[1,1], cl$centers[1,2], col=1)
segments( mat[cl$cluster==2,][,1], mat[cl$cluster==2,][,2],
          cl$centers[2,1], cl$centers[2,2], col=2)
```

Conclusions

- Deux grandes familles d'approches

- Non-paramétriques : ascendantes (CAH) ou divisives.
- Paramétriques : K-Means, SOM, ...

Hierarchical (CAH)

Advantages

- Faster computation.

Disadvantages

- Rigid;
- Cannot correct later for erroneous decisions made earlier.

Partitioning (K-means):

Advantages

- Optimal for certain criteria.

Disadvantages

- Need initial k ;
- Often require long computation times.