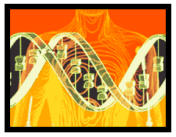
MASTER IBM 2010

INTRODUCTION À LA FOUILLE DE DONNÉES BIOMÉDICALES

COURS MASTER IBM 2010



JEAN-DANIEL ZUCKER

DR À L'IRD UR UMMISCO
(MODÈLISATION MATHÉMATIQUES ET INFORMATIQUES DES SYSTÈMES COMPLEXES)
UMR U755 INSERM/PARIS 6 ET LIM&BIO/PARIS 13

UPMC PARIS UNIVERSITAS — IRD Institut de recherche pour le développement — Inserm Institut national de la santé et de la recherche médicale — Lim&Bio

MODULE FOUILLE DE DONNÉES : DIDACTIQUE

- **Objectifs:**
  - COMPRENDRE LE DOMAINE DE LA FOUILLE DE DONNÉES
  - LE RÔLE DES ANALYSES PRÉDICTIVES DANS L'INFORMATIQUE MÉDICALE ET LA BIOINFO.
  - COMPRENDRE LES ALGORITHMES DE BASE DE LA FOUILLE DE DONNÉES BIOMÉDICALES
  - UTILISER UN ENVIRONNEMENT DE FOUILLE DE DONNÉES : CLEMENTINE (ET R)
- **Examen: projet personnel + soutenance**
- **Lecture: articles clefs à lire.**

ADMINISTRATIF: MODULE IF-FD MASTER IBM

Vendredi 13 Novembre 2009 – INTRO GÉNÉRALE / ANALYSE PRÉDICTIVE

- **La fouille de données**
- **Les données biomédicales**
- **Algorithmes pour la classification supervisée : Arbre de décision, SVM, Réseaux de Neurones, etc.**

Lundi 16 Novembre – FOUILLE DE DONNÉES (FIN)

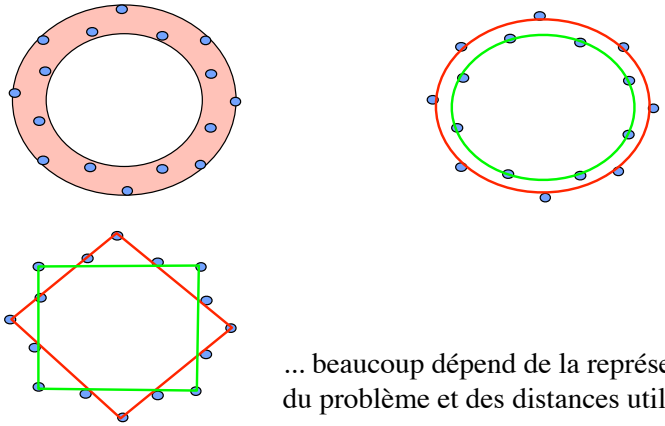
- **Réduction de dimension (t-test)**
- **Evaluation (Validation croisée, rééchantillonage)**
- **Le clustering (Dendrogramme, Algos CAH, KMeans, EM)**
- **Les règles d'associations (Algo Apriori)**

PLAN

Le clustering

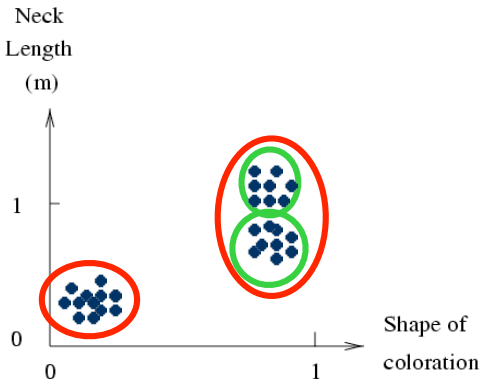
- **Utilisé dans de très nombreux domaines.**
- **But : rassembler les éléments (gènes) en groupes :**
  - ***Homogènes***
    - » **Les éléments dans un groupe sont *aussi similaires que possibl***
  - ***Séparés***
    - » **Les éléments de différents groupes sont *aussi différents que possible***

Le "clustering" est un problème mal pose...

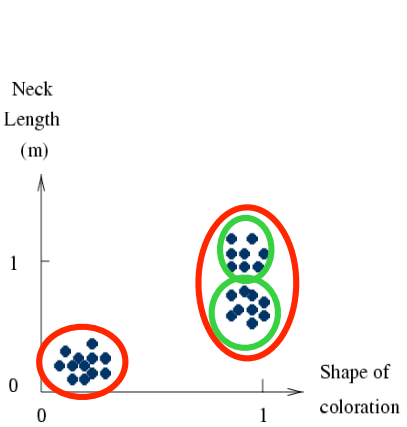


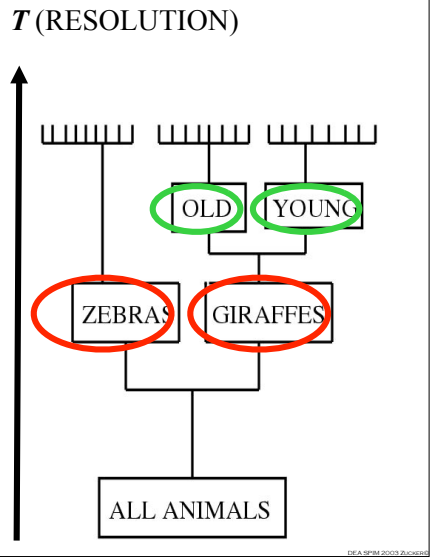
... beaucoup dépend de la représentation du problème et des distances utilisées.

Exemple de clustering



Notion de Dendrogramme





(RE)DÉFINITION DU PROBLÈME

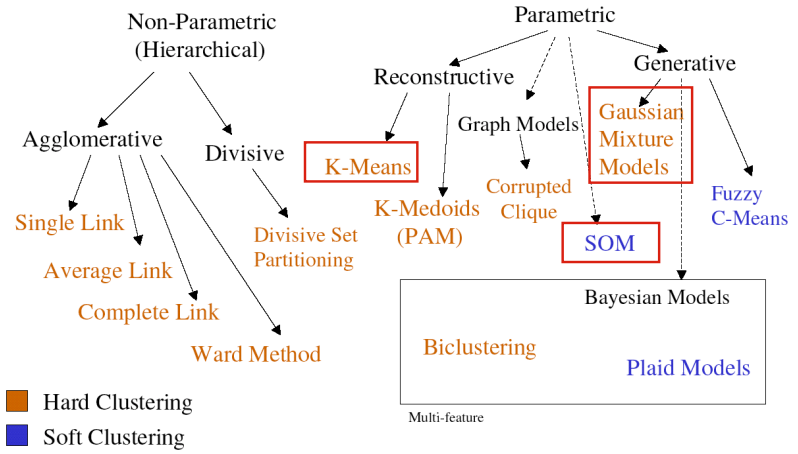
- Étant donnée une collection de n "points" $\boldsymbol{X}_i$, i=1,2,...n, définis dans un espace de dimension d, identifier des structures dans les données.
- Exemple : **partitioner** les données en ***M* groupes** (clusters), tels que les points d'un cluster soient **les plus "similaires"**
  - **Identifier des groupes stables**
  - **Générer des dendrogrammes**
- **Problème mal-posé :**
  - **Valeur de M ?**
  - **Notion de similarité ?**

PLAN

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 divisision*** **(descendantes)**
    - » **Division des groupes d'un niveau en sous-groupes du niveau inférieur**

TAXONOMIE DES MÉTHODES



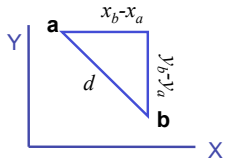
PLAN

Types de mesures de (di)similarité

- **Un paramètre crucial est la mesure the 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 Lp (généralise la distance euclidienne)

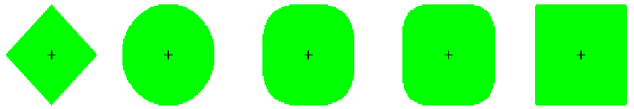
The L_p norm

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

p = 2, Euclidean Dist.

p = ∞, Manhattan Dist.(downtown Davis distance)

Equidistant points from a center, for different norms



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 $\cos(\alpha) = rp$

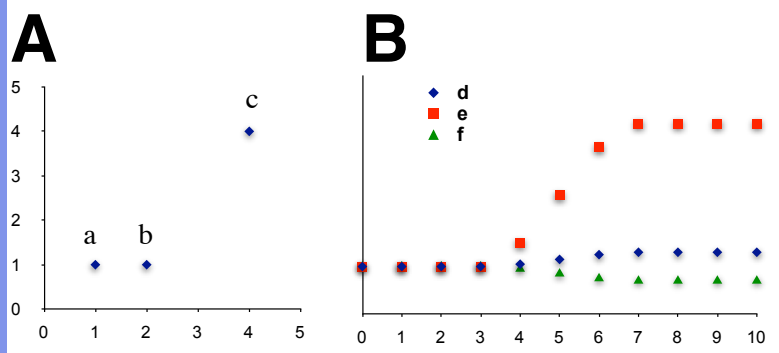
Le coefficient de corrélation n'est autre que le cosinus entre les deux vecteurs centrés !

Si r = 1, l'angle α = 0, les deux vecteurs sont colinéaires (parallèles).
Si r = 0, l'angle α = 90°, 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



Euclidian distances
• **a** close to **b**
Correlation coefficient
• **a** close to **c**

Euclidian distances
• **d** close to **f**
Correlation coefficient
• **d** close to **e**

Plan

I. Le Clustering

I.1) Un problème mal posé

I.2) Famille de méthodes

I.3) Importance de la distance

I.4) Algorithmes : CAH

II. Les règles d'associations

Algorithme Hierarchique (principe 1/)

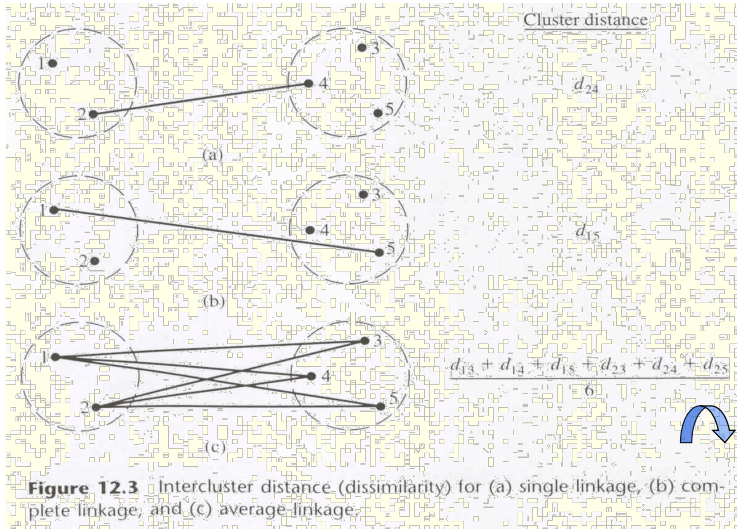


Figure 12.3 Intercluster distance (dissimilarity) for (a) single linkage, (b) complete linkage, and (c) average linkage.

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**

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

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

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

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

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

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

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

Algorithme Hierarchique (principe 4/)

- **Single linkage method**

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

	(135)	(24)
(135)	0	
(24)	6	0

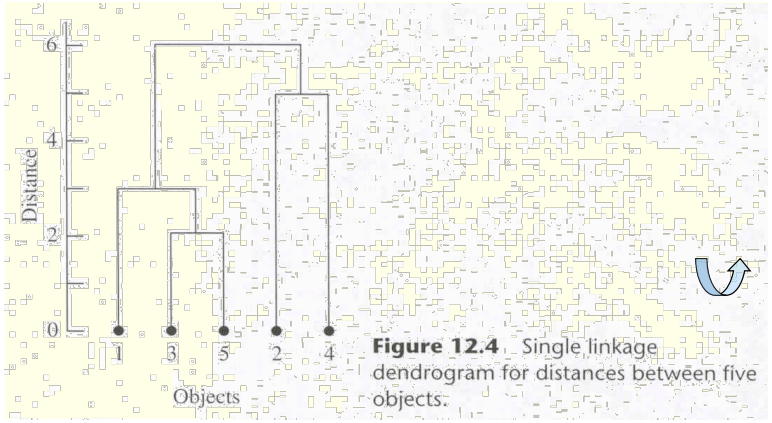


Figure 12.4 Single linkage dendrogram for distances between five objects.

HIERARCHICAL METHODS

Hierarchical clustering methods produce a tree or dendrogram.

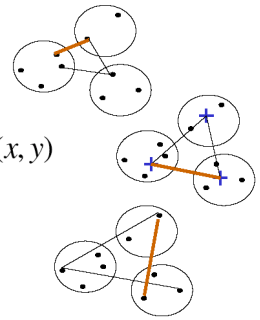
They avoid specifying how many clusters are appropriate by providing a partition for each k obtained from cutting the tree at some level.

The tree can be built in two distinct ways

- **bottom-up: agglomerative clustering;**
- **top-down: divisive clustering.**

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, $\frac{1}{|Ci||Cj|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$
  - Complete Link, $\max_{x \in C_i, y \in C_j} d(x, y)$
  - Others (Ward method-least squares)



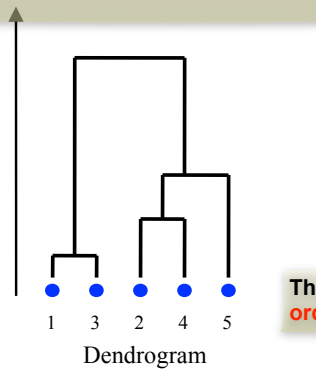
Agglomerative Hierarchical Clustering

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

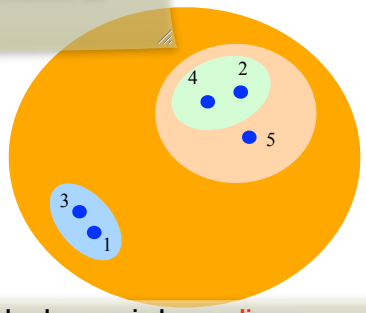
Single Linkage: distance between closest pair.

Complete Linkage: distance between farthest pair.

Average Linkage: average distance between all pairs



Dendrogram



The dendrogram induces a linear ordering of the data points

Divisive methods

- **Start with only one cluster.**

At each step, split clusters into two parts.

- ***Advantages*. Obtain the main structure of the data, i.e. focus on upper levels of dendogram.**
- ***Disadvantages*. Computational difficulties when considering all possible divisions into two groups.**

Hierarchical Clustering -Summary

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

Partitioning vs. hierarchical

Partitioning:

Advantages

- **Optimal for certain criteria.**

Disadvantages

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

Hierarchical

Advantages

- **Faster computation.**

Disadvantages

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

PLAN

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 du 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

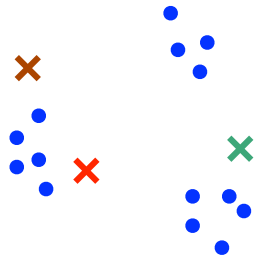
- Strengths
  - Simple to use
  - Fast
  - Can be used with very large data sets
- Weaknesses
  - The choice of the number of groups is arbitrary
  - The results vary depending on the initial positions of centres
  - The R implementation is based on Euclidian distance, no other metrics are proposed
- 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.

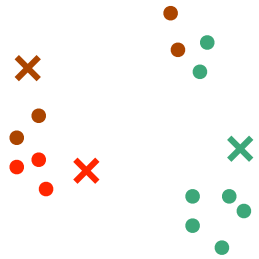


Iteration = 0

MÉTHODE DU CENTROÏDE

•Start with random positions of centroids.

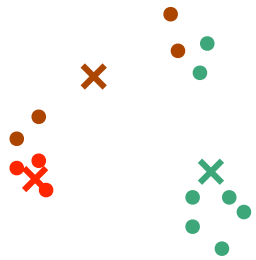
•Assign data points to centroids



Iteration = 1

Méthode du centroïde

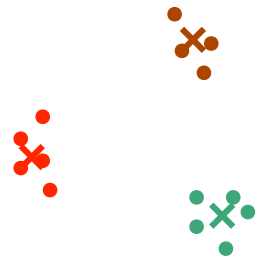
•Start with random positions of centroids.
•Assign data points to centroids
•Move centroids to center of assigned points



Iteration = 1

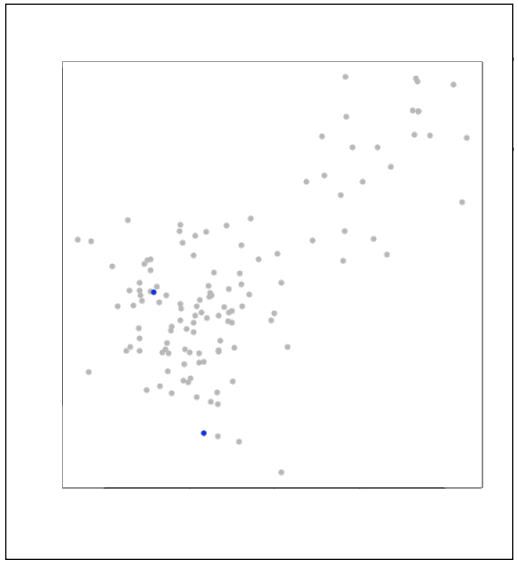
Méthode du centroïde

•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

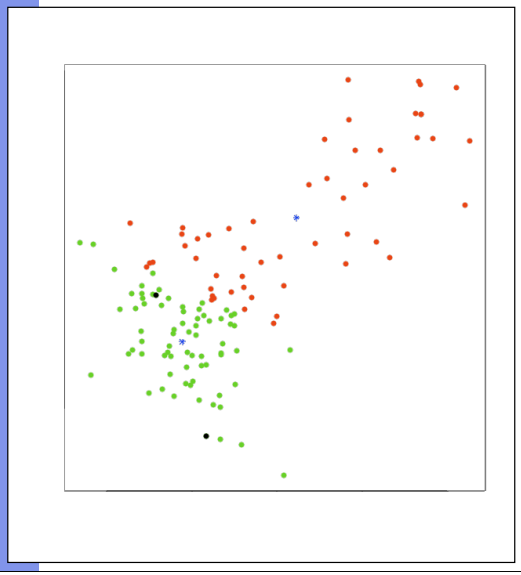


A set of 125 random points was generated

Two points are randomly chosen as seeds (blue)

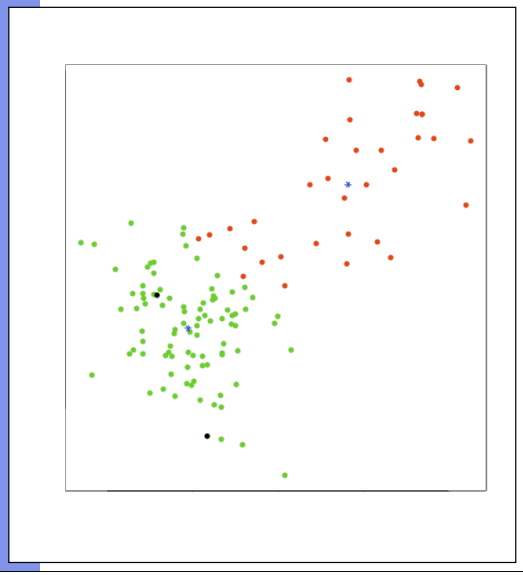
- **At each step,**
  - **points are re-assigned to clusters**
  - **centres are re-calculated**
- **Cluster boundaries and centre positions evolve at each iteration**

Mobile centres example - after 3 iterations



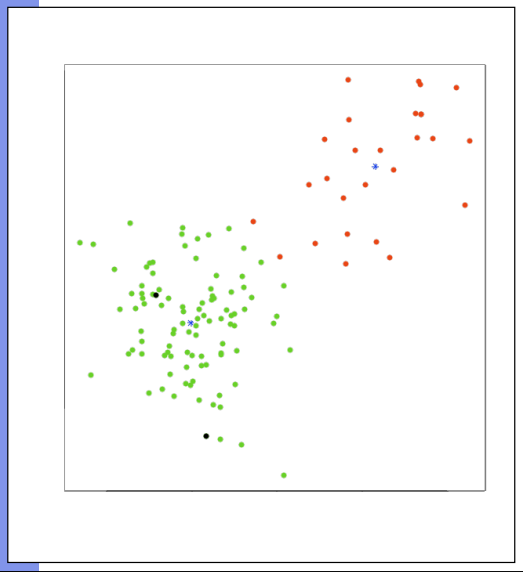
- **At each step,**
  - **points are re-assigned to clusters**
  - **centres are re-calculated**
- **Cluster boundaries and centre positions evolve at each iteration**

Mobile centres example - after 4 iterations



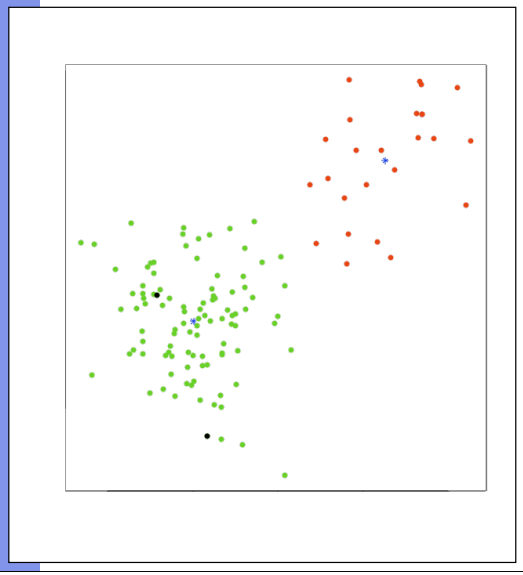
- At each step,
  - points are re-assigned to clusters
  - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

Mobile centres example - after 5 iteration



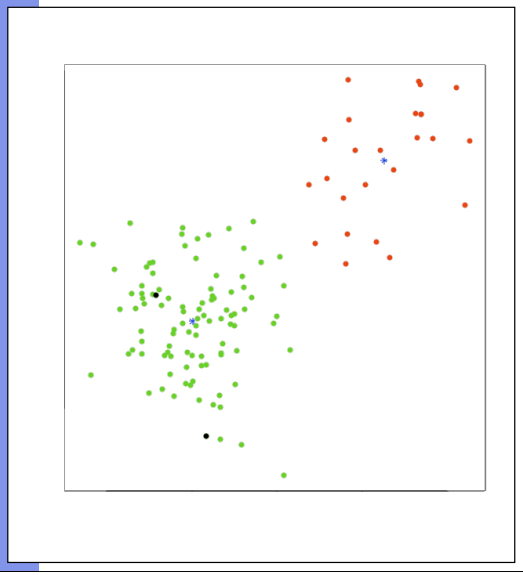
- At each step,
  - points are re-assigned to clusters
  - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

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 << n
  - Terminent souvent dans un optimum local
- 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 éxecutions 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
```


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)
```

PLAN

I. Le Clustering

I.1) Un problème mal posé

I.2) Famille de méthodes

I.3) Importance de la distance

I.4) Algorithmes : SOM

II. Les règles d'associations

Les règles d'association

- **Association rule mining:**
  - » **Finding frequent patterns, associations, correlations, structures among sets of items or objects in transaction databases, relational databases, and other information repositories.**
  - » **Find all rules that correlate the presence of one set of items with another set of items**
- **Applications:**
  - » **Analyse du panier de la ménagère, cross-marketing**
  - » **Clustering, classification**
  - » **etc.**
- **Example :**
  - » **"If a customer buys pizzas, in 60% of cases, he/she also buys beers. This happens in 3% of all transactions**

Les règles d'association [Agrawal 91,93]

"If a customer buys pizzas, in 60% of cases, he/she also buys beers. This happens in 3% of all transactions"

Pizzas ⇒ beers

- ***Règle d'association* : *A* ⇒ *B***
  où *A* ⊂ I, *B* ⊂ I et *A*∩*B* = ∅.

Confidence: 60%

- **La règle *A* ⇒ *B* a une confiance *c* dans D**
  si *c*% des transactions de D contenant *A* contiennent aussi *B*.
- **La règle *A* ⇒ *B* a un support *s* dans D**
  si *s*% des transactions de D contiennent *A* ∪ *B*.

Support: 3%

Tâche : découvrir toutes les règles d'association de D
Avec support ≥ *minsup* and confidence ≥ *minconf*

- **Algorithmes : Apriori [Agrawal 91] , AprioriTID [Agrawal 93]**

Des mesures sur les règles

[Cherfi 02]

- **Support index**

$$Sup(A \Rightarrow B) = \frac{\| \{T \in D \mid A \cup B \subseteq T\} \|}{\| D \|} = p(A,B)$$

- **Confidence index**

$$Conf(A \Rightarrow B) = p(B/A) = \frac{p(A,B)}{p(A)}$$

- **Interest index [Brin 97]**

$$Int(A \Rightarrow B) = \frac{p(A,B)}{p(A) * p(B)}$$

- **Conviction index [Brin 97]**

$$Conv(A \Rightarrow B) = \frac{p(A) * p(\neg B)}{p(A, \neg B)}$$

- **Dependency index**

$$Dep\ (A \Rightarrow B) = |p(B/A) - p(B)|$$

- **Novelty index [Piatetsky-Sha**

$$Nov\ (A \Rightarrow B) = p(A,B) - p(A) * p(B)$$

- **Satisfaction index**

$$Sat(A \Rightarrow B) = \frac{p(\neg B) - p(\neg B/A)}{p(\neg B)}$$

- **Surprise index [Azé 02**

$$Spr\ (A \Rightarrow B) = \frac{p(A,B) - p(A,\neg B)}{p(B)}$$