

Fouille de données

Classification non supervisée

Sylvain Meignier – sylvain.meignier@univ-lemans.fr

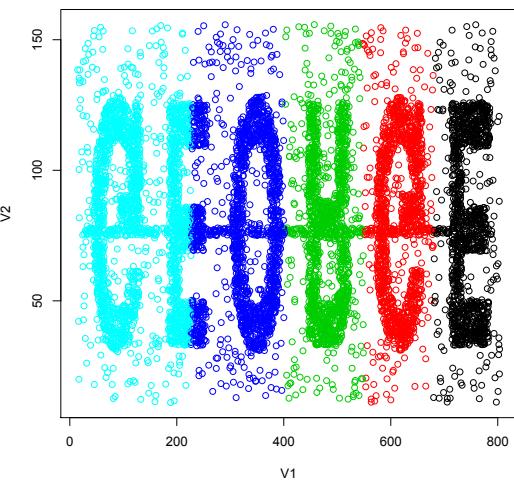
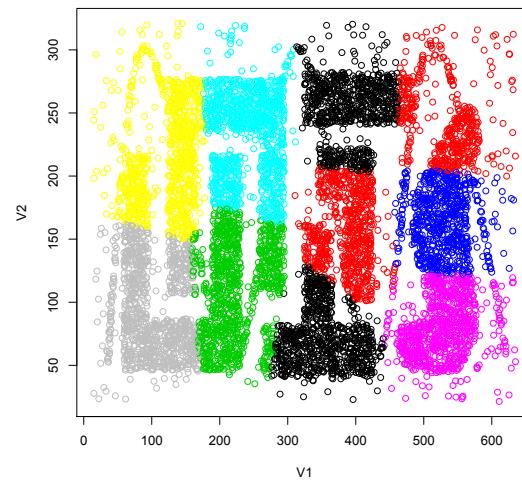
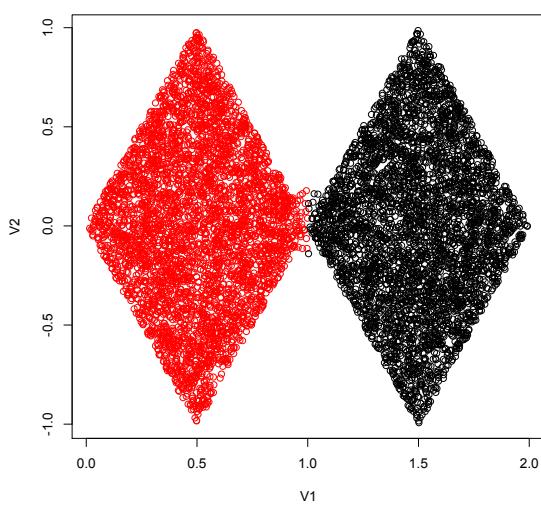
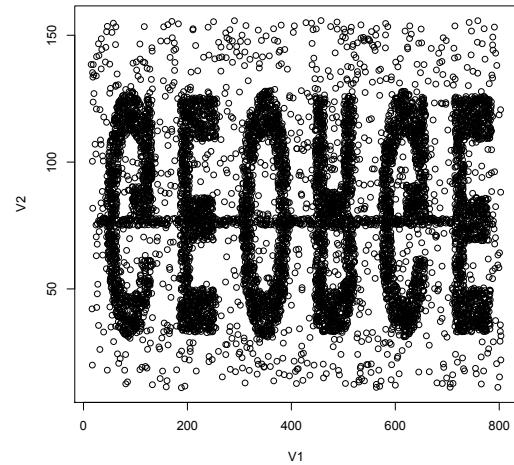
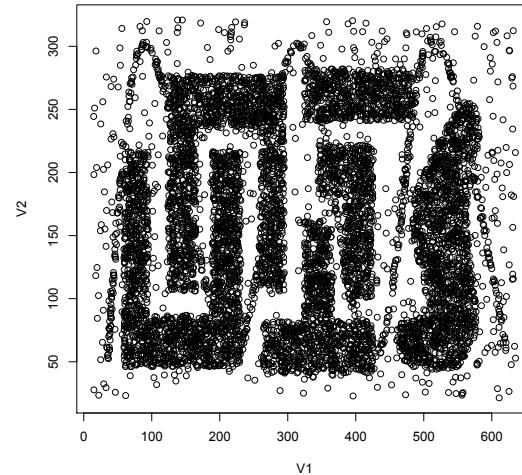
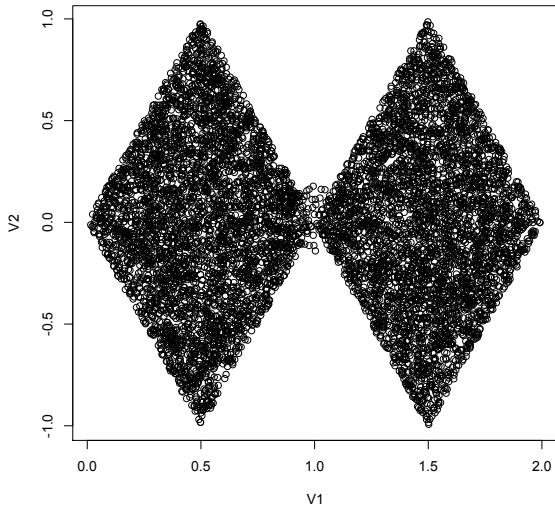
Bibliographie

- ❑ <https://moodle.insa-rouen.fr/course/view.php?id=92§ion=2>
- ❑ <http://www.grappa.univ-lille3.fr/~ppreux/Documents/notes-de-cours-de-fouille-de-donnees.pdf>

Classification non supervisée

- Objectif
 - On dispose d'individus non étiquetés
 - On souhaite les regrouper en fonction de leurs ressemblances
 - = les structurer en des classes homogènes
- Difficultés :
 - Notion de ressemblance entre deux individus
 - Organiser les individus sans disposer d'information sur les groupes à former.
 - Qu'est-ce qu'une classe ?
- Construire des groupes de cet ensemble tel que :
 - les individus qui se ressemblent appartiennent au même groupe
 - les individus peu ressemblants appartiennent à des groupes différents

Exemple



Individus

- Soit un ensemble $X = \{x_i^k\}$ individus décrits par P attributs
 - i = indice des individus
 - k = indice des attributs
 - Généralement on aura un tableau de N lignes et P colonnes
- Nature des attributs
 - quantitatifs = un nombre décimal
 - Poids, taille, ...
 - ordinal = un nombre entier
 - Le nombre de membres, couleur
 - catégorie = une valeur parmi un ensemble fini de valeurs non ordonnées
 - Couleur, genre

Catégorie

- Méthode
 - Pas d'apprentissage
 - 1er classe de méthodes : Non hiérarchique
 - on décompose l'ensemble d'individus en k groupe
 - = découpage à plat
 - 2^e classe de méthodes : Hiérarchique
 - on décompose l'ensemble d'individus en une arborescence de groupes
 - = construire un arbre

Catégorie

- ❑ Séparation entre les individus
 - ❑ Exclusif
 - ❑ Chaque individu appartient à un seul groupe
 - ❑ partition au sens mathématique
 - ❑ Non exclusif
 - ❑ Chaque individu peut appartenir à plusieurs groupes
 - ❑ Chaque individu appartient est associés à chaque groupe avec une probabilité
 - ❑ Degré d'appartenance

Partition

- Partition : soit un ensemble X quelconque, un ensemble C de sous-ensembles de X est une partition de X si :
 - aucun élément de C n'est vide ;
 - l'union des éléments de C est égale à X ;
 - les éléments de C sont deux à deux disjoints.

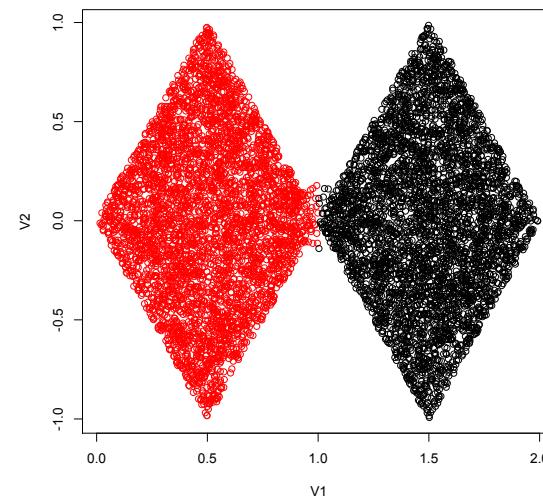
$$X = x_1, \dots, x_n$$

$$C = c_1, \dots, c_k \text{ avec :}$$

$$c_i \neq \{\}$$

$$\cup_{i=1}^k c_i = X$$

$$c_i \cap c_j = \{\}, \forall i \neq j$$



Considération algorithmique

- ❑ La partition optimale d'un ensemble de n individus est un problème NP-Complet
 - ❑ NP-Complet = un problème de décision vérifiant les propriétés suivantes :
 - ❑ Il est possible de vérifier une solution efficacement en temps polynomial mais nous ne disposons pas d'algorithme en temps polynomial pour trouver une solution
 - ❑ Tous les problèmes de la classe NP se ramènent à celui-ci via une réduction polynomiale ; cela signifie que le problème est au moins aussi difficile que tous les autres problèmes de la classe NP.
- ❑ Énoncer toutes les solutions n'est valable que pour des problèmes de petite taille (et encore...)
- ❑ Utiliser des algorithmes gloutons, des heuristiques, etc. en espérant s'approcher de la meilleure solution
- ❑ →Optimalité d'une partition ?

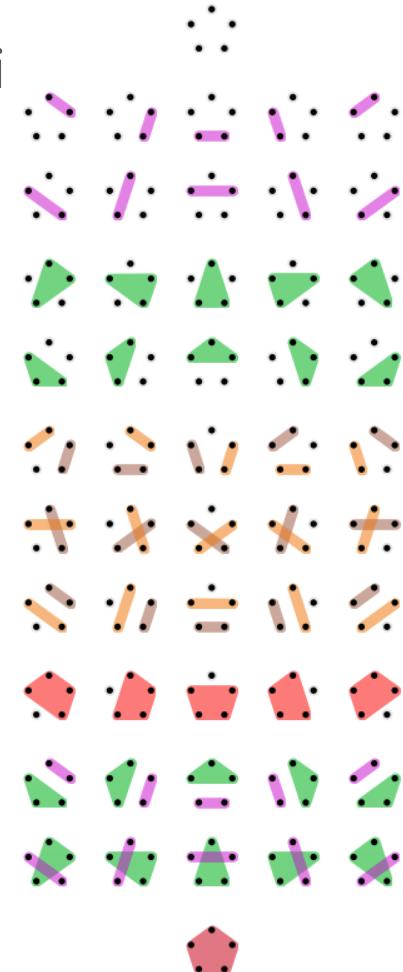
Nombre de partitions différentes

- Nombre de partitions distinct d'un ensemble fini : nombre de Bell

$$B_{n+1} = \sum_{k=0}^c C_n^k B_n$$

$$C_n^k = \frac{n!}{k!(n-k)!}$$

- Pour $n = 30$, B_{30} environ $8,40 \times 10^{23}$



Source Wikipedia

Ressemblance entre individus

- On a deux individus, on veut mesurer leur ressemblance
- Solution 1 : distance
 - Distance de Manhattan
 - Distance euclidienne
 - Distance de Tchebychev

$$d : X \times X \rightarrow R^+$$

$$d(x_i, x_j) = 0 \Leftrightarrow i = j$$

$$d(x_i, x_j) = d(x_j, x_i)$$

$$d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$$

$$d_1(x_i, x_j) = \sum_{k=1}^P |x_i^k - x_j^k|$$

$$d_2(x_i, x_j) = \sqrt{\sum_{k=1}^P (x_i^k - x_j^k)^2}$$

$$d_\infty(x_i, x_j) = \sup_{1 \leq k \leq P} |x_i^k - x_j^k|$$

Distance entre individus

- On a deux individus, on veut mesurer leur "ressemblance"

- Distance de Minkowski ($p > 0$)

- les attributs sont traités indépendamment et de la même façon

$$d_p(x_i, x_j) = \sqrt[p]{\sum_{k=1}^M |x_i^k - x_j^k|^p}$$

- Cas $p=2$: distance euclidienne

$$d_2(x_i, x_j) = \sqrt{\sum_{k=1}^M (x_i^k - x_j^k)^2}$$

$$d_2(x_i, x_j) = \sqrt{(x_i - x_j)^T (x_i - x_j)}$$

- Cas $p=1$: distance de Manhattan

- Déplacement uniquement sur les axes

- Plusieurs chemin

$$d_1(x_i, x_j) = \sum_{k=1}^M |x_i^k - x_j^k|$$

Distance entre individus

- Cas $p = \infty$:

$$d_{\infty}(x_i, x_j) = \max_{k=1, \dots, M} |x_i^k - x_j^k|$$

- Cas $p = 0$:

- Si vecteur de bits \rightarrow distance de Hamming

- Le nombre de bits à changer dans x_i pour obtenir x_j

$$d_0(x_i, x_j) = \sum_{k=1}^M \delta(x_i^k \neq x_j^k)$$

- Distance d'édition = distance de Levenshtein

- Nombre de caractères à échanger, ajouter ou supprimer dans x_i pour obtenir x_j

Distance entre individus

- Définition :

$$d : X \times X \rightarrow R^+$$

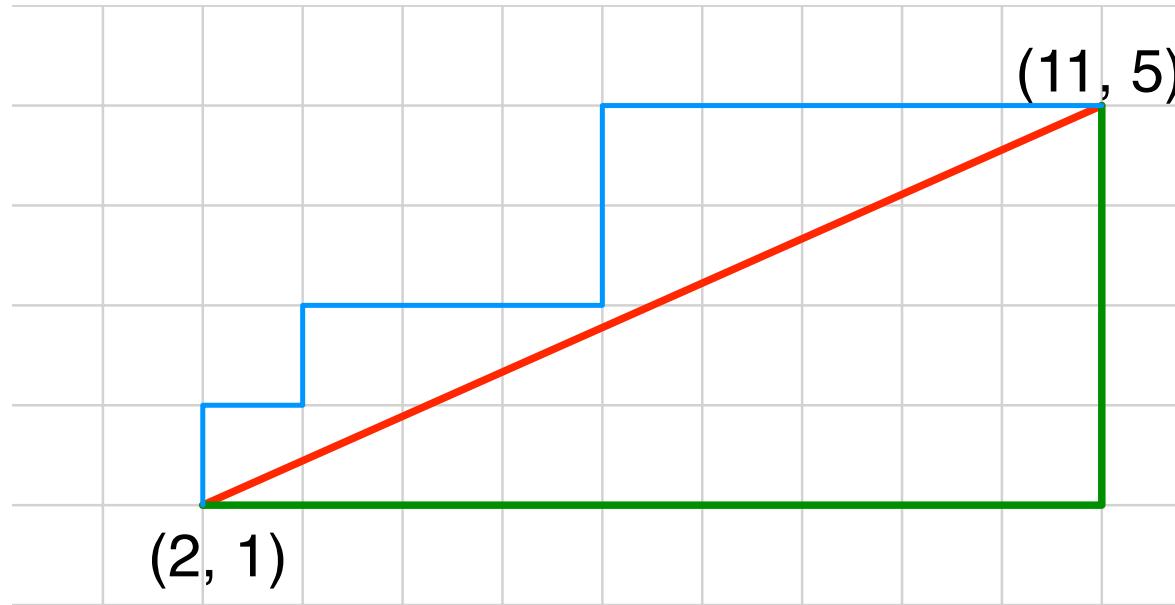
$$d(x_i, x_j) = 0 \Leftrightarrow i = j$$

$$d(x_i, x_j) = d(x_j, x_i)$$

$$d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$$

Exemple

- Distance de Manhattan : chemins vert et bleu
 - $D1 = \text{abs}(2-11) + \text{abs}(1-5) = 13$
- Distance euclidienne : chemin en rouge
 - $D2 = \sqrt{(2-11)^2 + (1-5)^2} = 9.8$



Distance

- Distance de Mahalanobis
 - Accorde un poids moins important aux attributs les plus dispersés

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T \Sigma^{-1} (x_i - x_j)}$$

- La matrice Σ est une matrice de covariance qui a pour effet de décolérer et de normaliser les données
- Si $\Sigma = I$, on a la distance euclidienne
- Si Σ est diagonale on a la distance euclidienne normalisée

$$d_M(x_i, x_j) = \sqrt{\frac{(x_i - x_j)^2}{\sigma}}$$

Discussion

- ❑ Toutes ces distances donnent les mêmes résultats si les classes sont compactes et bien séparées
- ❑ On obtient des résultats différents lorsque les classes sont proches ou si leur forme n'est pas hypersphériques
- ❑ En utilisant le saut minimal, l'arbre créer en théorie des graphes couvrant (spanning tree), c'est-à-dire un arbre avec un chemin permettant de se déplace de n'importe quel noeud vers n'importe quel autre.
- ❑ En utilisant le saut maximal, la distance entre les classes est la distance la plus grande parmi tous les individus. En théorie des graphes, les individus de la classe forment un graphe complet
- ❑ En utilisant la moyenne, nous croisions un compromis entre les deux situations précédentes. Cependant, cette méthode ne peut pas être utilisée avec des similarités. La similarité entre des moyennes est généralement impossible à définir (ou difficile)

Caractéristique d'une classe

- Chaque classe contient de 1 à n individus
- Pour la classe C_l à $|C_l|$ individus
 - Centre de gravité :

$$G_l = \frac{1}{|C_l|} \sum_{x_i \in C_l} x_i$$

- Inertie = mesure le degré de concentration des points autour du centre de gravité. On cherche à avoir l'inertie la plus faible.

$$\mathcal{I}_l = \sum_{x_i \in C_l} d(x_i, G_l)^2$$

Qualité d'une partition

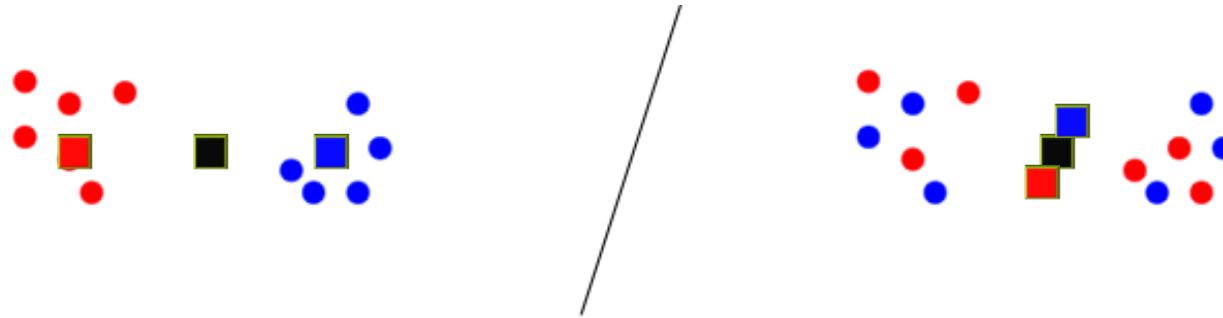
- Centre de gravité = la moyenne des individus

$$G = \frac{1}{N} \sum_{i=1}^N x_i \quad \mathcal{I} = \sum_{i=1}^N d(x_i, G)^2$$

- Inertie intraclasse et l'inertie interclasse
 - W = within = intraclasse
 - B = between = interclasse
 - L'inertie inter-classe mesure l'éloignement des centres des classes entre elles. On cherche à maximiser cette valeur.

$$\mathcal{I}^B = \sum_{l=1}^{|C|} \frac{1}{|C_l|} d(G_l, G)^2 \quad \mathcal{I}^W = \sum_{l=1}^{|C|} \mathcal{I}_l$$

Qualité d'une partition

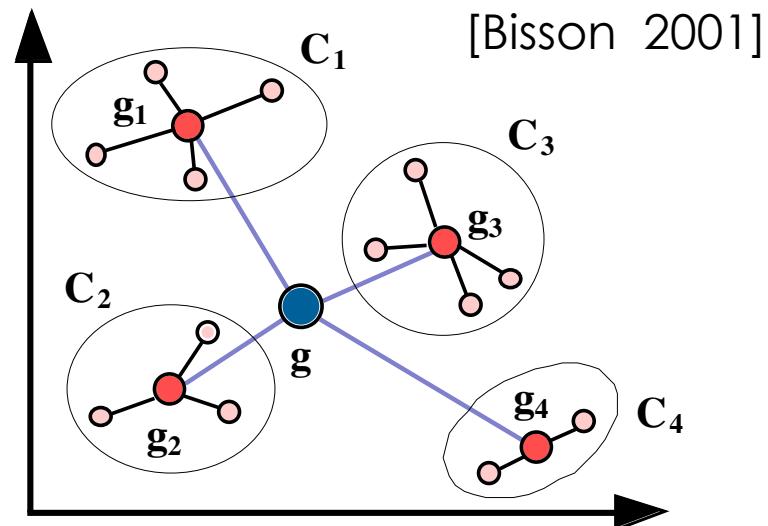
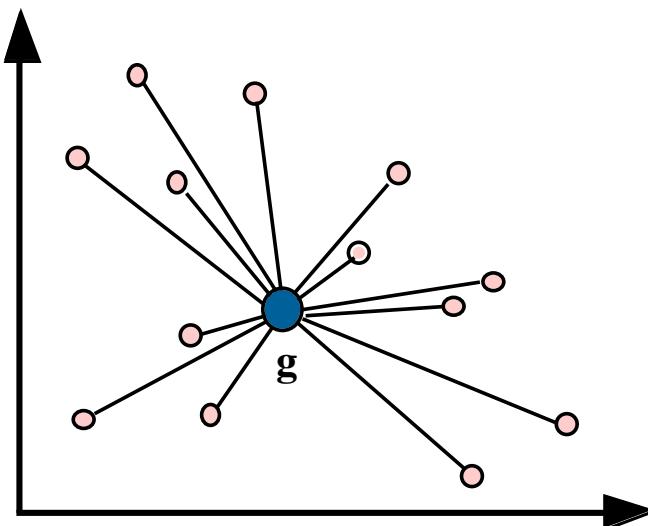


- Inerties des deux cas identiques
 - La position des individus est identique
- Cas de gauche
 - Inertie intraclasse \mathcal{I}^W faible, inertie interclasse grande \mathcal{I}^B
- Cas de droite
 - Inertie intraclasse \mathcal{I}^W grande, inertie interclasse faible \mathcal{I}^B

Qualité d'une partition

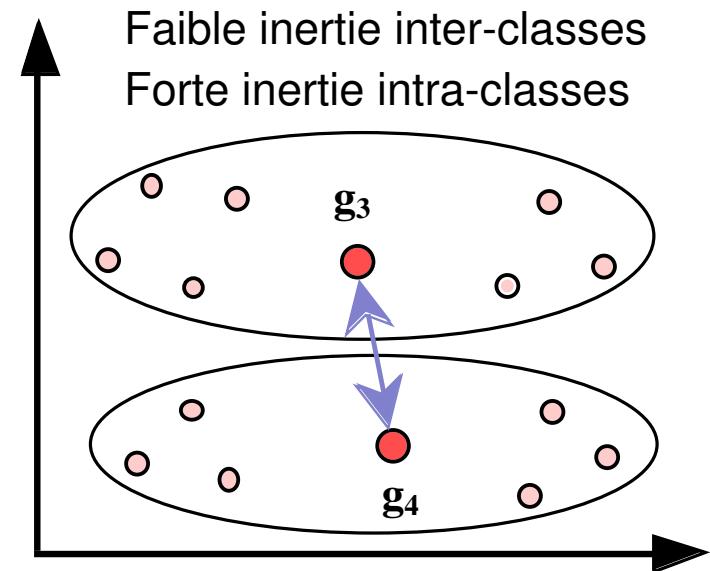
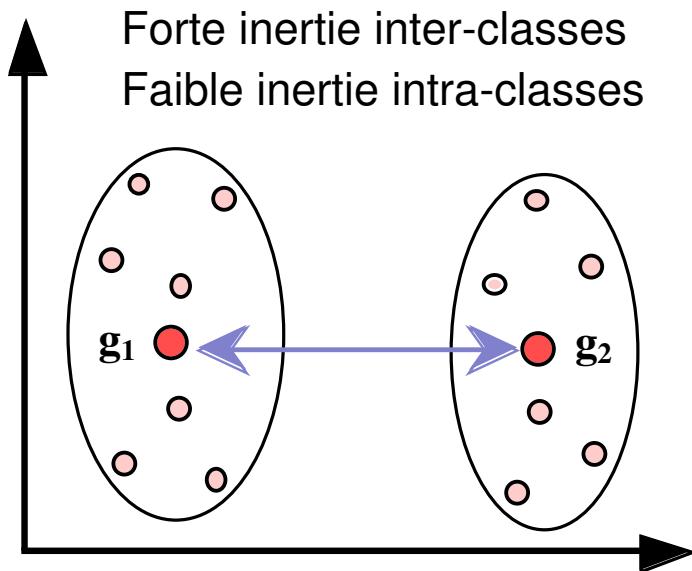
- Inertie globale = Inertie Intraclasse+ Inertie Interclasse
- Bonne partition = minimiser l'inertie intraclasse et maximiser l'inertie interclasse
- Théorème de Huggens
 - I est constant, donc il suffit s'intéresser à l'inertie intra ou inter classe

$$I = I^W + I^B$$



Qualité d'une partition

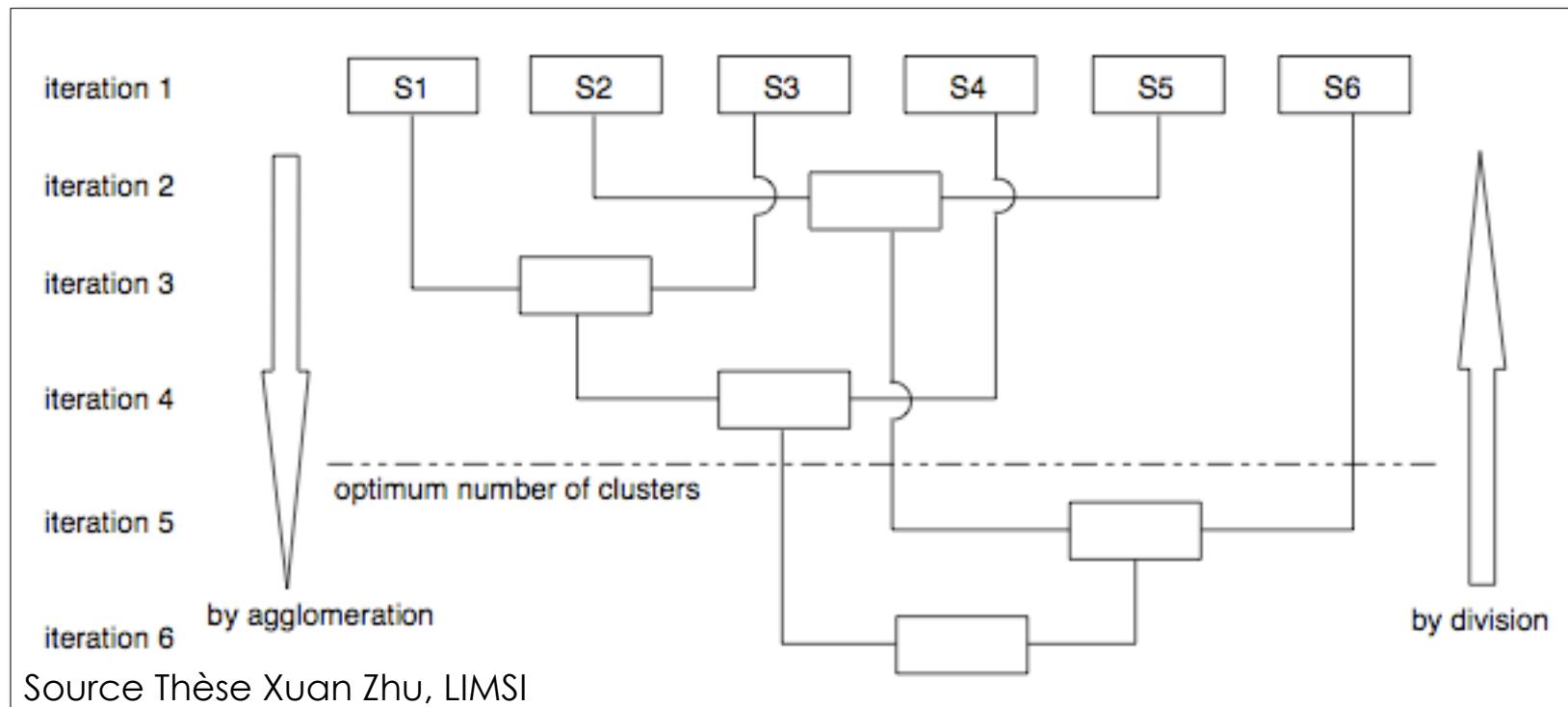
- Source [Bisson 2001]



Classification hiérarchique

Classification hiérarchique

- ❑ Ascendante : par agglomération,
 - ❑ N classes vers une classe
- ❑ Descendante : par division
 - ❑ Une classe vers N classes



Classification hiérarchique ascendante

- ❑ Algorithme itératif
 - ❑ À chaque itération, les deux classes les plus « proches » sont groupées
 - ❑ Produire une **séquence** de groupes imbriqués les uns dans les autres
- ❑ Besoin
 - ❑ d'une mesure entre les classes pour grouper les classes les plus proches
 - ❑ d'une méthode pour mettre à jour les mesures après chaque regroupement
 - ❑ d'un critère d'arrêt

Algorithme

- Initialisation
 - Chaque individu est placé dans une classe formant n classes
$$C_O = \{c_1, \dots, c_j, \dots, c_k, \dots, c_n\}$$
 - Calculer la matrice des mesures entre chaque paire de classes
$$[d(c_i, c_j)]_{i,j}$$
- Itération i, (i > 1 et i < n-1)
 - Étape 1 : grouper la paire de classes avec la mesure la plus proche
 - Pour i=1, on passe de n à n-1 classes...
 - La nouvelle classification est $C_i = \{c_1, \dots, c_{n-i}\}$
 - Étape 2 :
 - Calculer les mesures associées à la nouvelle classe
 - Étapes 3 : test d'arrêt



Mesure entre classes

- ❑ Distance
- ❑ Mesure entre 2 classes : une mesure de dissimilarité

$$d : C \times C \rightarrow R^+$$

$$d(c_i, c_j) = 0 \Leftrightarrow i = j$$

$$d(c_i, c_j) = d(c_j, c_i), \forall i, j$$

- ❑ Mais en fait, on peut juste avoir un degré d'association ou de ségrégation entre les classes

$$d : C \times C \rightarrow R$$

$$d(c_i, c_j) = d(c_j, c_i), \forall i, j$$

Critère d'arrêt

- Quand arrêter le regroupement des classes ?
- Comparer les classifications

$$C_{i-1} = \{c_1, \dots, c_{n-i}, c_{n-(i-1)}\}$$

$$C_i = \{c_1, \dots, c_{n-i}\}$$

- La différence entre C_{i-1} et C_i : les classes c_j et c_k ont été groupées dans C_k
- Mesure entre 2 classifications revient à s'arrêter si

$$d(c_j, c_k) > \alpha$$

Mise à jour des distances

- Après le regroupement des classes c_j et c_k dans c_k
 - Supprimer la ligne et la colonne $d(c_j, c.)$ $d(c., c_j)$
 - Mettre à jour $d(c_k, c.)$ $d(c., c_k)$

$d(c_1, c_1)$...	$d(c_1, c_k)$...	$d(c_1, c_j)$...	$d(c_1, c_i)$
\vdots	\ddots				\vdots	
$d(c_k, c_1)$...	$d(c_k, c_k)$...	$d(c_k, c_j)$...	$d(c_1, c_i)$
\vdots			\ddots			\vdots
$d(c_j, c_1)$...	$d(c_j, c_k)$...	$d(c_j, c_j)$...	$d(c_j, c_i)$
\vdots					\ddots	\vdots
$d(c_i, c_1)$...	$d(c_i, c_k)$...	$d(c_i, c_j)$...	$d(c_i, c_i)$

Mise à jour des distances

- Mise à jour :
 - En calculant les distances de la nouvelle classe avec les autres
 - Estimation complète
 - À partir des distances entre les éléments des classes

saut minimum :

$$d(c_k, c.) = \min_{x \in c_k, y \in c.} d(x, y)$$

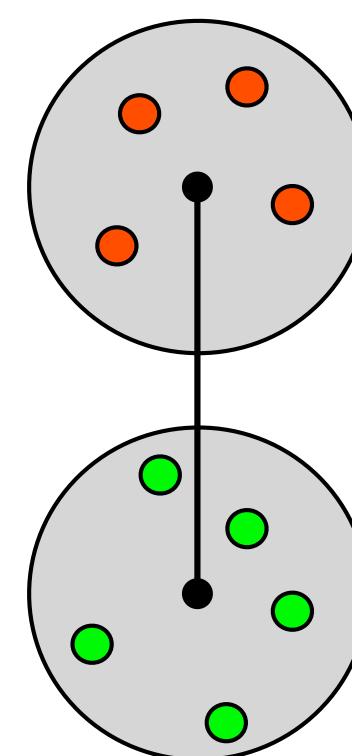
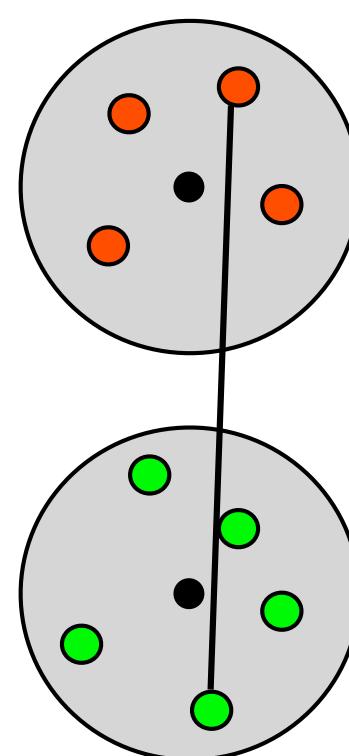
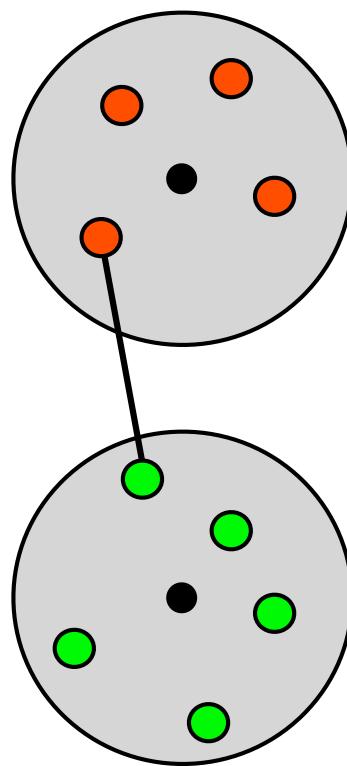
saut maximum :

$$d(c_k, c.) = \max_{x \in c_k, y \in c.} d(x, y)$$

lien moyen :

$$d(c_k, c.) = \frac{\sum_{x \in c_k, y \in c.}}{|c_k| \times |c.|}$$

- Minimal, maximal, moyen

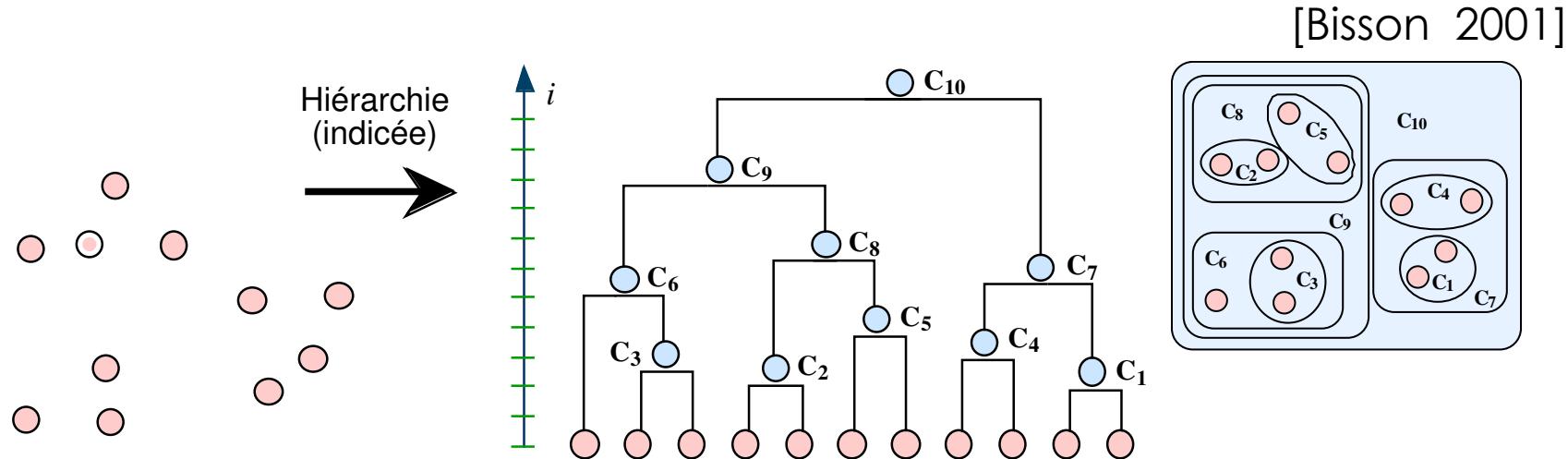


Mise à jour des distances

- ❑ Saut minimal (single linkage)
 - ❑ tendance à produire des classes générales (par effet de chaînage)
 - ❑ sensibilité aux individus bruités
- ❑ Saut maximal (complete linkage)
 - ❑ tendance à produire des classes spécifiques (on ne regroupe que des classes très proches)
 - ❑ sensibilité aux individus bruités
- ❑ Lien moyen
 - ❑ tendance à produire des classes de variance proche

Graphique : dendrogramme

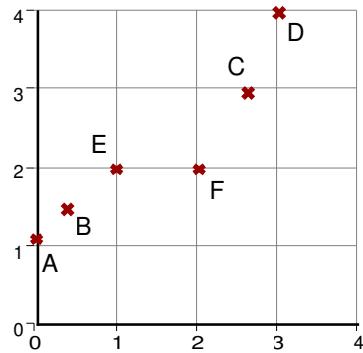
- Un dendrogramme du Grec dendron "arbre", -gramma "dessiner"
 - représentation des fusions successives
 - Hauteur d'une classe dans le dendrogramme = distance entre les 2 classes avant fusion



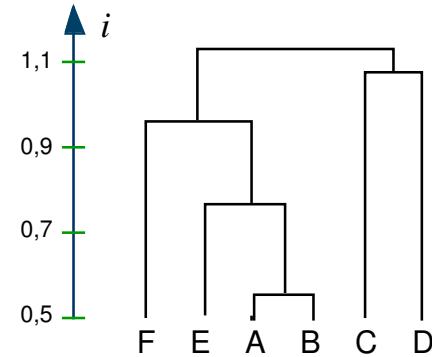
Différences

- Suivant la méthode de mise à jour des distances, on obtient un résultat différent

Données (métrique : dist. Eucl.)

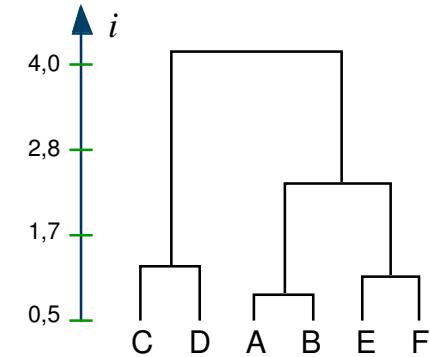


Saut minimal



[Bisson 2001]

Saut maximal



Exemple

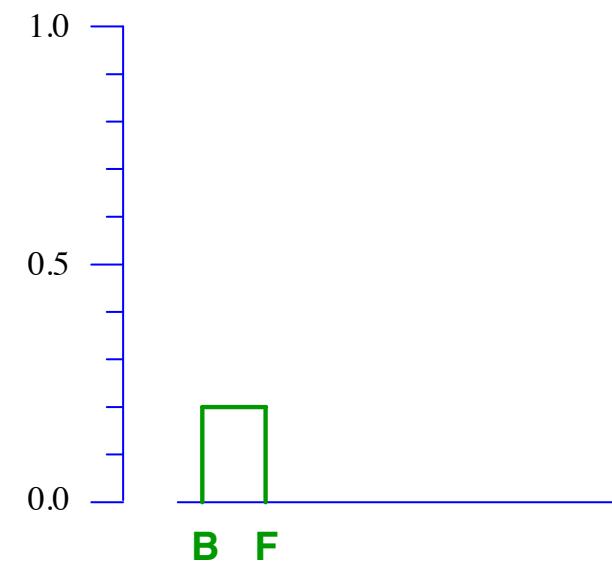
- Une matrice de distances

samples	A	B	C	D	E	F	G
A	0	0.5000	0.4286	1.0000	0.2500	0.6250	0.3750
B	0.5000	0	0.7143	0.8333	0.6667	0.2000	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.6667	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8000	0.8571
E	0.2500	0.6667	0.4286	1.0000	0	0.7778	0.3750
F	0.6250	0.2000	0.6667	0.8000	0.7778	0	0.7500
G	0.3750	0.7778	0.3333	0.8571	0.3750	0.7500	0

▣ Saut maximum

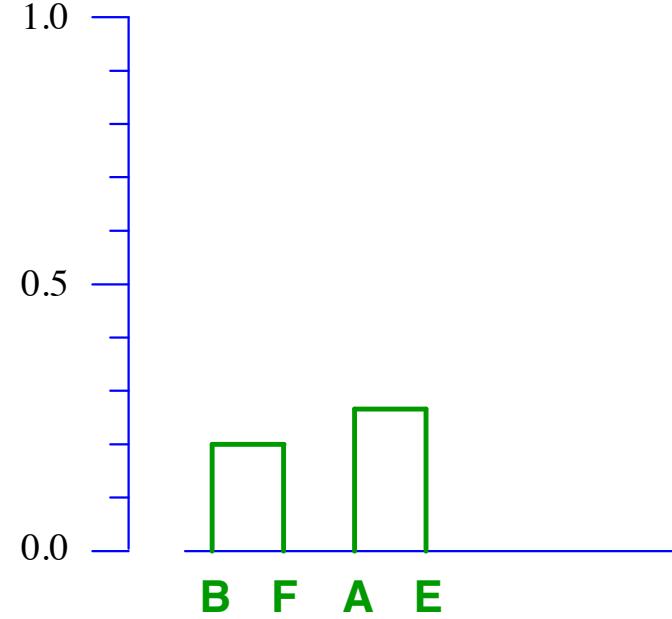
samples	A	B	C	D	E	F	G
A	0	0.5000	0.4286	1.0000	0.2500	0.6250	0.3750
B	0.5000	0	0.7143	0.8333	0.6667	0.2000	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.6667	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8000	0.8571
E	0.2500	0.6667	0.4286	1.0000	0	0.7778	0.3750
F	0.6250	0.2000	0.6667	0.8000	0.7778	0	0.7500
G	0.3750	0.7778	0.3333	0.8571	0.3750	0.7500	0

samples	A	(B,F)	C	D	E	G
A	0	0.6250	0.4286	1.0000	0.2500	0.3750
(B,F)	0.6250	0	0.7143	0.8333	0.7778	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8571
E	0.2500	0.7778	0.4286	1.0000	0	0.3750
G	0.3750	0.7778	0.3333	0.8571	0.3750	0

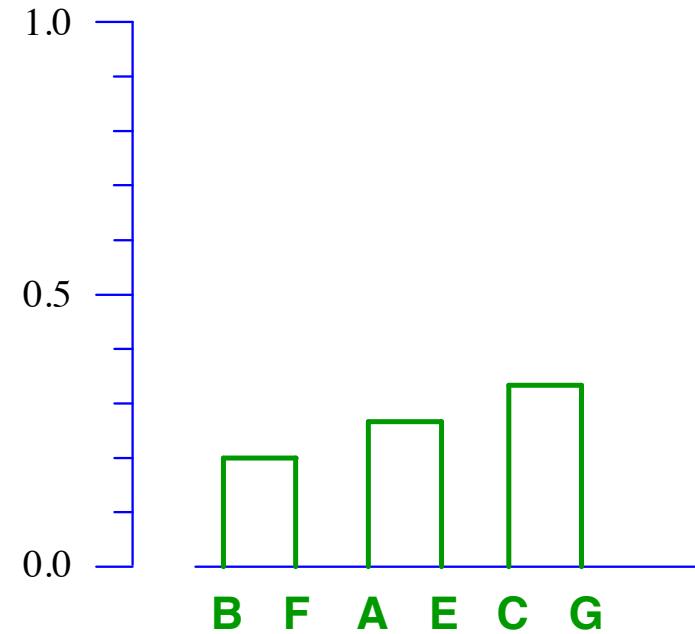


samples	A	(B,F)	C	D	E	G
A	0	0.6250	0.4286	1.0000	0.2500	0.3750
(B,F)	0.6250	0	0.7143	0.8333	0.7778	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8571
E	0.2500	0.7778	0.4286	1.0000	0	0.3750
G	0.3750	0.7778	0.3333	0.8571	0.3750	0

samples	(A,E)	(B,F)	C	D	G
(A,E)	0	0.7778	0.4286	1.0000	0.3750
(B,F)	0.7778	0	0.7143	0.8333	0.7778
C	0.4286	0.7143	0	1.0000	0.3333
D	1.0000	0.8333	1.0000	0	0.8571
G	0.3750	0.7778	0.3333	0.8571	0

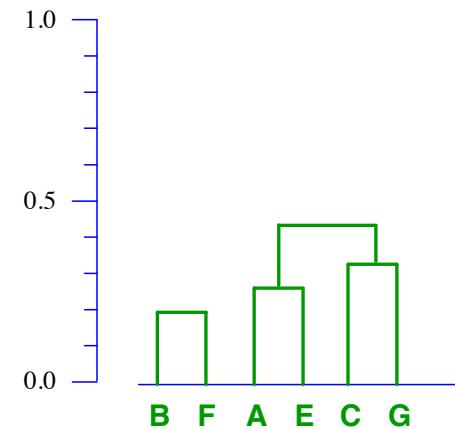


samples	(A,E)	(B,F)	C	D	G
(A,E)	0	0.7778	0.4286	1.0000	0.3750
(B,F)	0.7778	0	0.7143	0.8333	0.7778
C	0.4286	0.7143	0	1.0000	0.3333
D	1.0000	0.8333	1.0000	0	0.8571
G	0.3750	0.7778	0.3333	0.8571	0

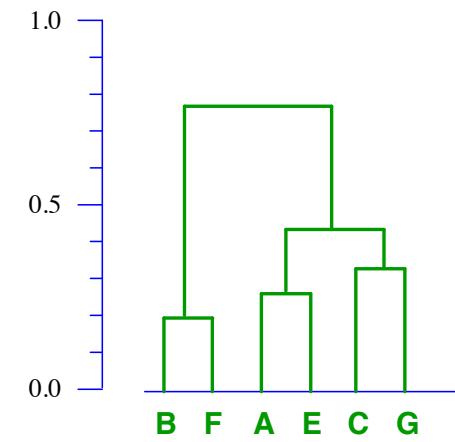


samples	(A,E)	(B,F)	(C,G)	D
(A,E)	0	0.7778	0.4286	1.0000
(B,F)	0.7778	0	0.7778	0.8333
(C,G)	0.4286	0.7778	0	1.0000
D	1.0000	0.8333	1.0000	0

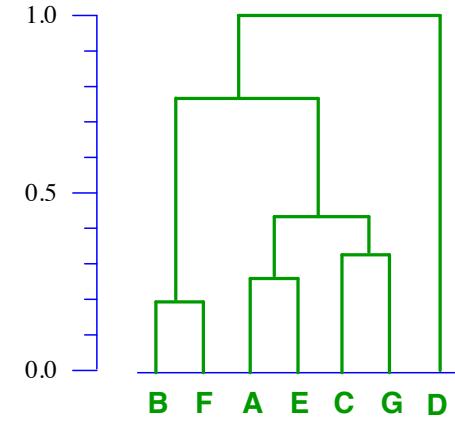
samples	(A,E)	(B,F)	(C,G)	D
(A,E)	0	0.7778	0.4286	1.0000
(B,F)	0.7778	0	0.7778	0.8333
(C,G)	0.4286	0.7778	0	1.0000
D	1.0000	0.8333	1.0000	0



samples	(A,E,C,G)	(B,F)	D
(A,E,C,G)	0	0.7778	1.0000
(B,F)	0.7778	0	0.8333
D	1.0000	0.8333	0



samples	(A,E,C,G,B,F)	D
(A,E,C,G,B,F)	0	1.0000
D	1.0000	0



Classification K plus proche voisin

K plus proche voisin

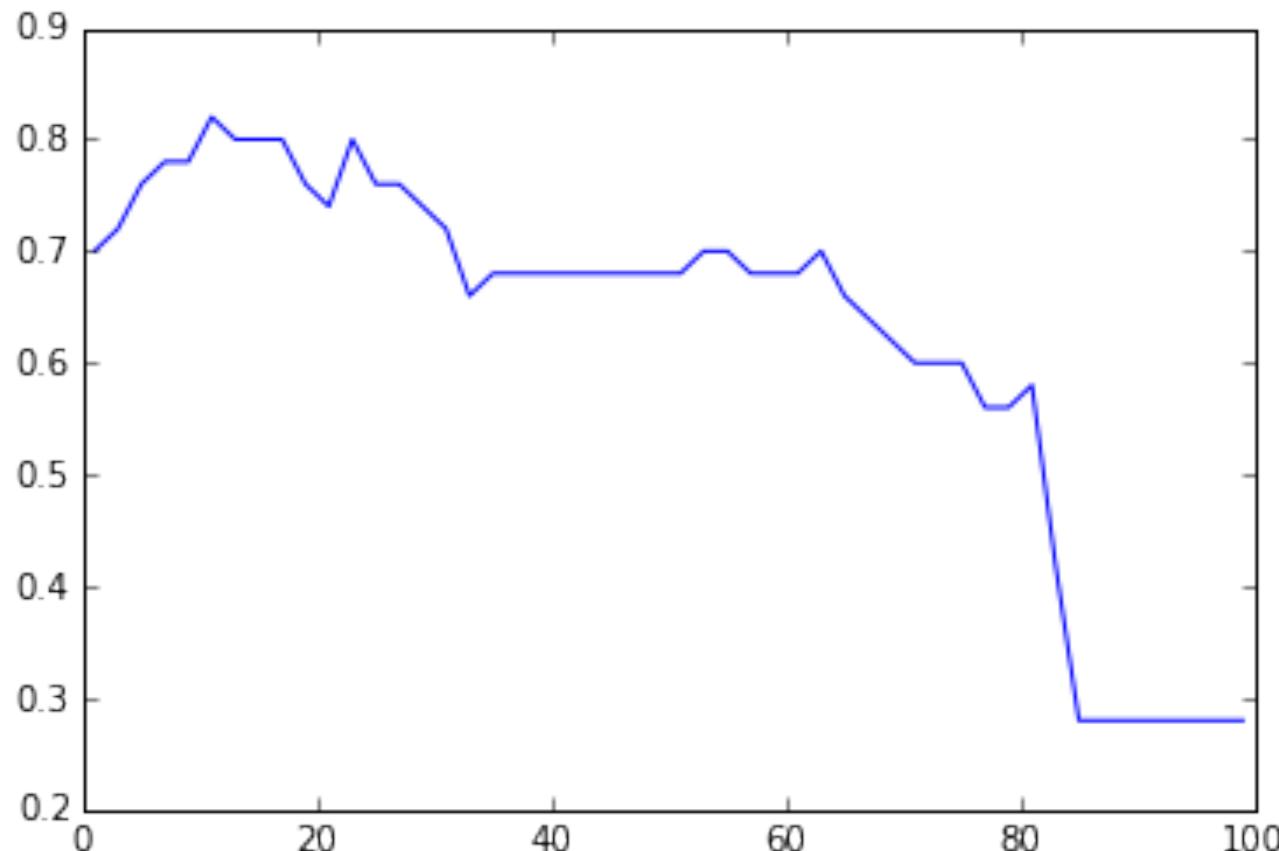
- ❑ Un des algorithmes les plus simples en IA
- ❑ Données :
 - ❑ un corpus étiqueté
 - ❑ Une distance entre individus
- ❑ Pas d'apprentissage
- ❑ En test
 - ❑ Chercher les k plus proches voisins de l'individu x à classer
 - ❑ K=1,3,...
 - ❑ Choisir la classe majoritaire parmi ces k voisins
- ❑ Méthode
 - ❑ non-paramétrique
 - ❑ qui utilise les exemples déjà connus
 - ❑ En $O(n)$

1-NN

- ❑ 1-NN sépare parfaitement le corpus, on recherche l'individu le plus proche
- ❑ Typiquement, k est un nombre impair choisi entre 3 et 99
- ❑ Lorsque $k = N$, la classe majoritaire est choisie indépendamment de l'observation x
- ❑ Choix de k :
 - ❑ À partir du corpus d'apprentissage en validation croisée
 - ❑ Mesurer le taux de bonnes prédictions
- ❑ Attributs
 - ❑ Besoin d'être normalisés, attention à l'attribut dominant
 - ❑ Nombre d'attributs trop grand, besoin de réduire le nombre
 - ❑ PCA par exemple

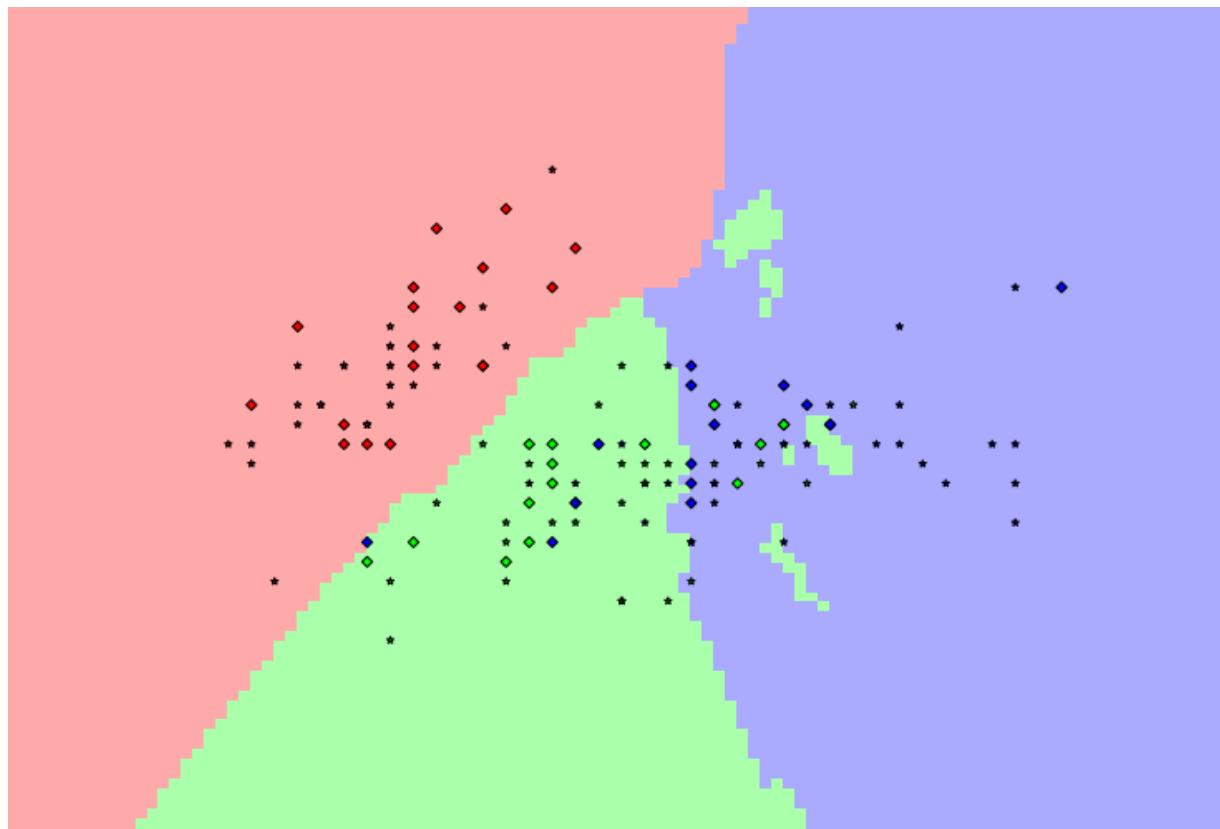
Exemple : iris

- Prédiction avec k de 1 à 100 avec un pas de deux



Exemple : iris

- Prédiction avec k de 1 à 100 avec un pas de deux



Classification en K classes

Critère de classification

- ❑ la séparation
 - ❑ maximiser les écarts entre classes, qui sont fonctions des distances interclasses
- ❑ l'homogénéité
 - ❑ les plus concises possible, on cherche à minimiser le *diamètre*
 - ❑ distances intraclasses
- ❑ la dispersion
 - ❑ minimiser une *fonction d'inertie*, la somme des carrés des écarts à un centre, qu'il soit réel ou virtuel

K-Means

- On a N individus représentés par P attributs
- On recherche une partition en K classes ($K < N$)
- Solution : recherche exhaustive
 - Générer toutes les partitions en K classes
 - Évaluer la qualité de chaque partition, retenir la meilleure partition
 - C'est qui minimise l'inertie intraclasse
- Même en se limitant aux partitions en K classes, ce nombre est très grand (nombre de Stirling de seconde espèce)
 - Pour $N=9$ et $K=4$, on a 7770 partitions $\frac{1}{K!} \sum_{k=1}^K (-1)^{K-k} C_k^K k^N$

K-means (Hartigan and Wong)

- ❑ Solution 2 : minimiser l'inertie interclasse
 - ❑ Trouver une heuristique qui évite d'énumérer toutes les solutions
 - ❑ On ne peut pas garantir d'avoir la partition optimale (au sens de l'inertie intraclasse)
 - ❑ Algorithme de k-means
 - ❑ Algorithme très connu, il a de nombreuses variantes

K-means (Hartigan and Wong)

■ Principe

- Si on connaît k centres de gravité, chaque individu peut être affecté au centre le plus proche formant ainsi une classe
- Connaissant une classe, on sait calculer son centre de gravité

$$G_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

- En alternant, les étapes d'affectation et de calcul de centre, itérer jusqu'à convergence

Algorithme

$I \leftarrow \infty$

prendre K centres arbitraires $c_k \in \mathcal{D}$

répéter

pour $k \in \{1, \dots, K\}$ **faire**

$G_k \leftarrow \emptyset$

fin pour

pour $i \in \{1, \dots, N\}$ **faire**

$k^* \leftarrow \arg \min_{k \in \{1, \dots, K\}} d(x_i, c_k)$

$G_{k^*} \leftarrow G_{k^*} \cup \{x_i\}$

fin pour

pour $k \in \{1, \dots, K\}$ **faire**

$c_k \leftarrow$ centre de gravité de G_k

fin pour

$I \leftarrow \mathcal{I}_W$

 calculer \mathcal{I}_W

jusque $I - \mathcal{I}_W <$ seuil

$$G_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

$$SSE = \sum_{l=1}^{|C|} \sum_{i \in C_l} d(x_i, g_l)^2$$

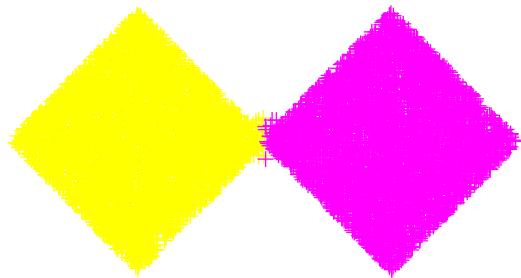
Exemple



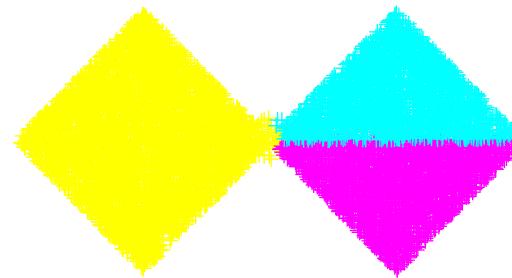
<http://shabal.in/visuals.html>

Exemple

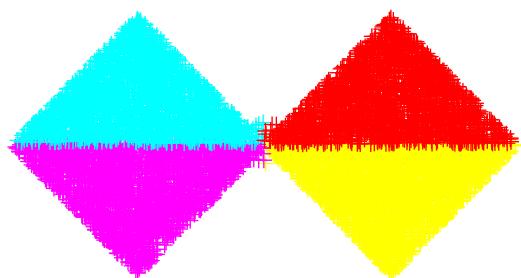
$K = 2$



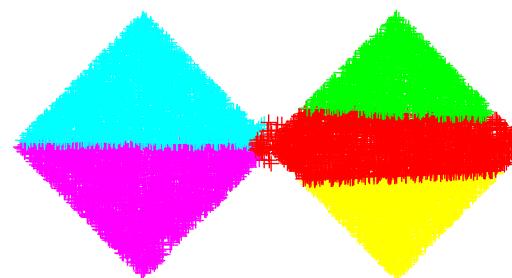
$K = 3$



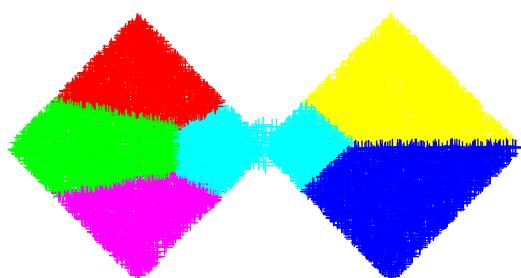
$K = 4$



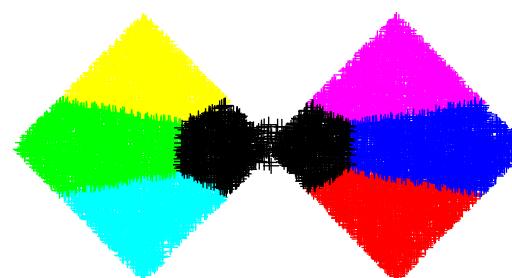
$K = 5$



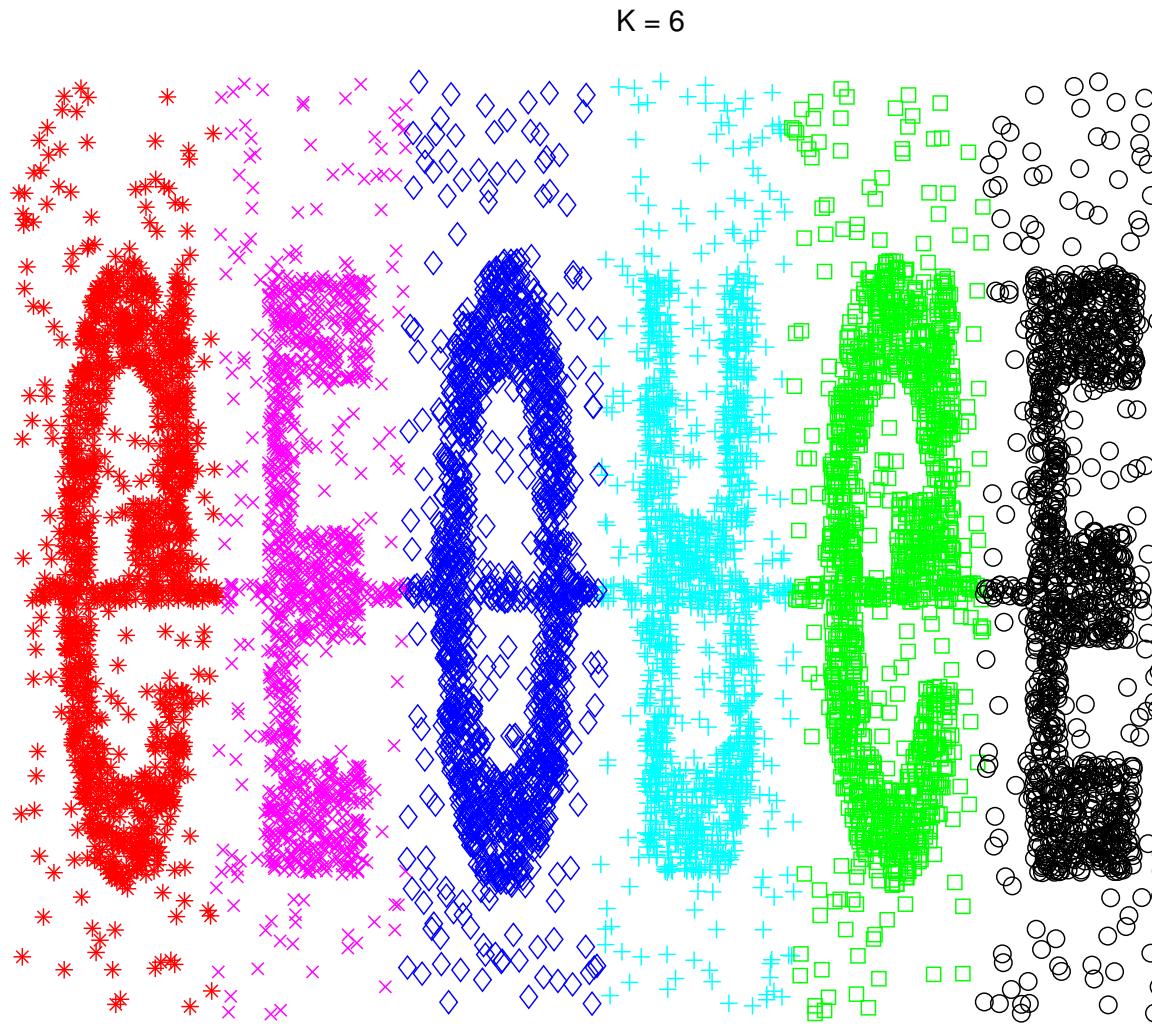
$K = 6$



$K = 7$

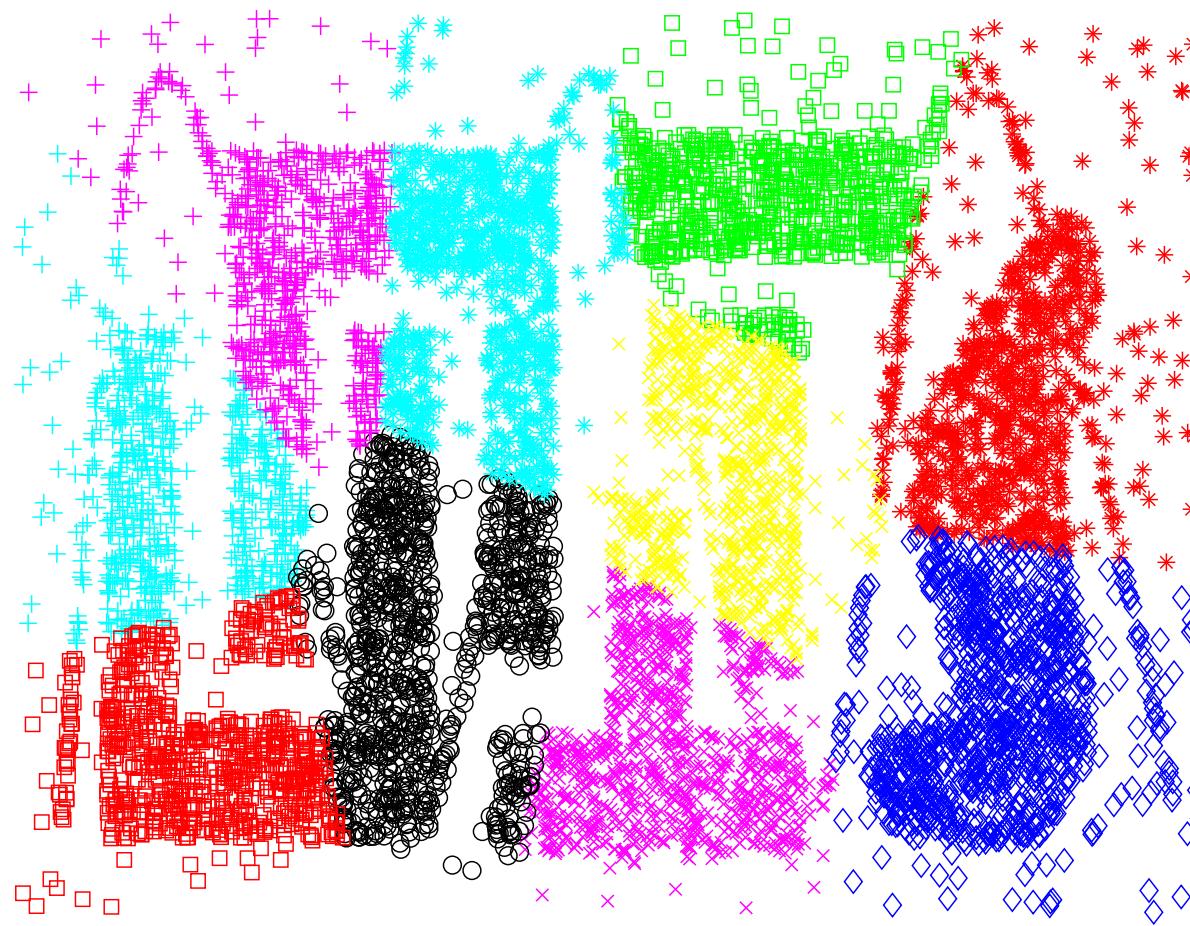


Exemple



Exemple

$K = 10$

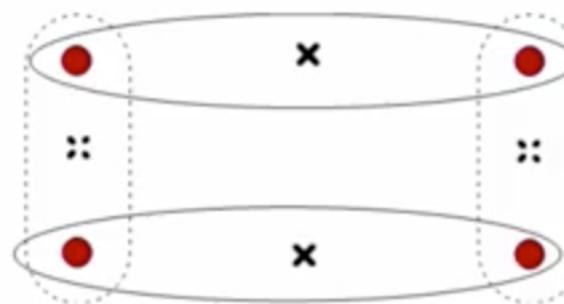


K-means

- À chaque itération l'inertie intraclasse diminue
- K-means converge vers un minimum local
 - La convergence est rapide

K-means

- Mais l'initialisation des centres est problématique
 - Choisir des individus aléatoirement (Mac Queen 67)
 - Choisir aléatoirement dans l'ensemble de définitions des individus
 - Deux initialisations peuvent donner des résultats différents



K-means

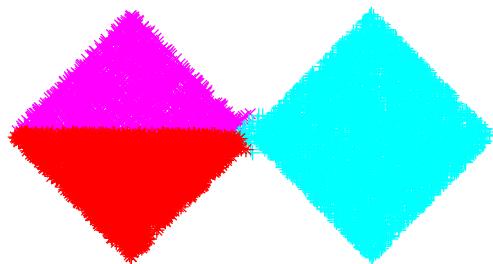
- ❑ Problème d'initialisation
 - ❑ Exécuter plusieurs fois K-means avec des jeux d'initialisation différents
 - ❑ Puis Sélectionner classes qui apparaissent majoritairement dans les différentes partitions
 - ❑ Kmean++ (Arthur et al. 07)
 - ❑ Sélectionner un centre aléatoirement parmi les individus
 - ❑ Le 2e centre sera l'individu le plus éloigné du premier centre
 - ❑ Continuer jusqu'à obtenir les k centres

K-means

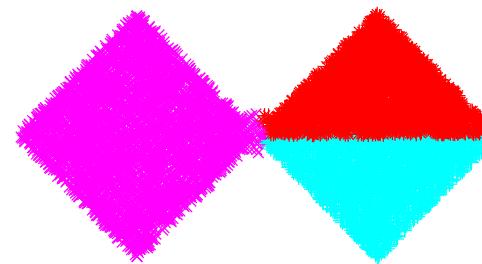
- ❑ K doit être connu
 - ❑ On peut tester différentes valeurs de K, mais il est alors difficile de comparer les partitions
 - ❑ L'inertie intraclasse n'est pas le bon critère, elle diminue lorsque le nombre de classes augmente
 - ❑ Se fonder sur la différence entre les inerties
 - ❑ Imposer des contraintes sur le volume, le nombre d'individus dans une classe...

Différentes initialisations

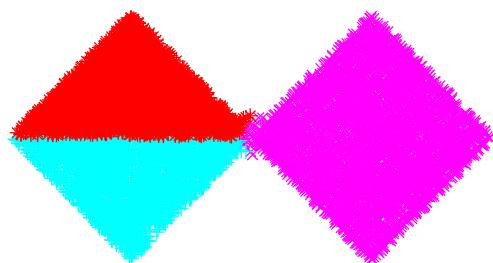
$K = 3$



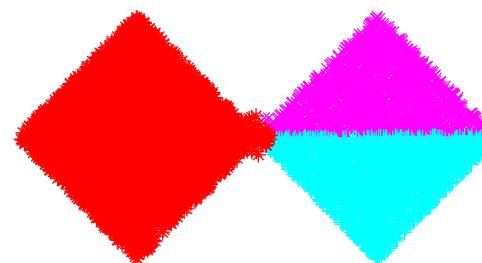
$K = 3$



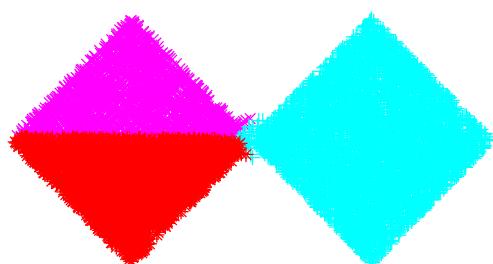
$K = 3$



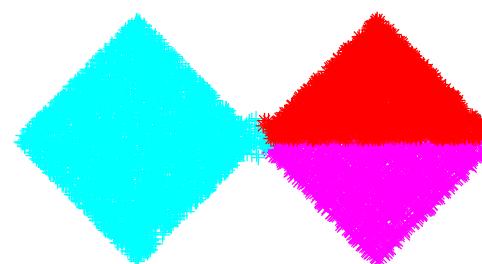
$K = 3$



$K = 3$

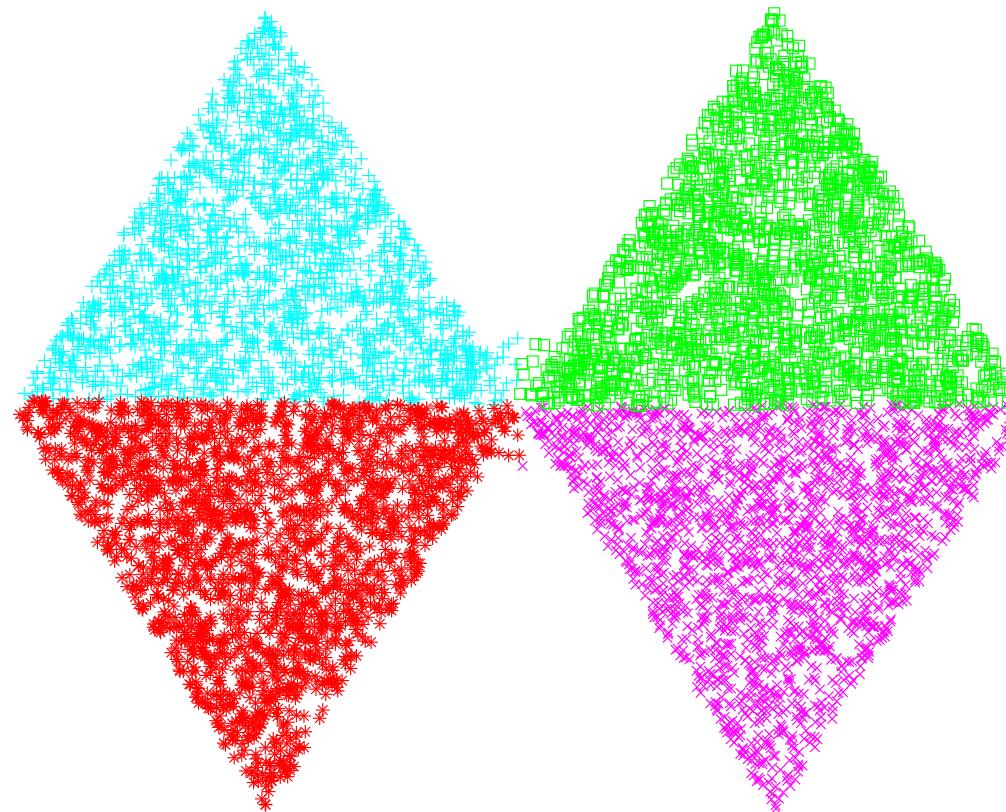


$K = 3$

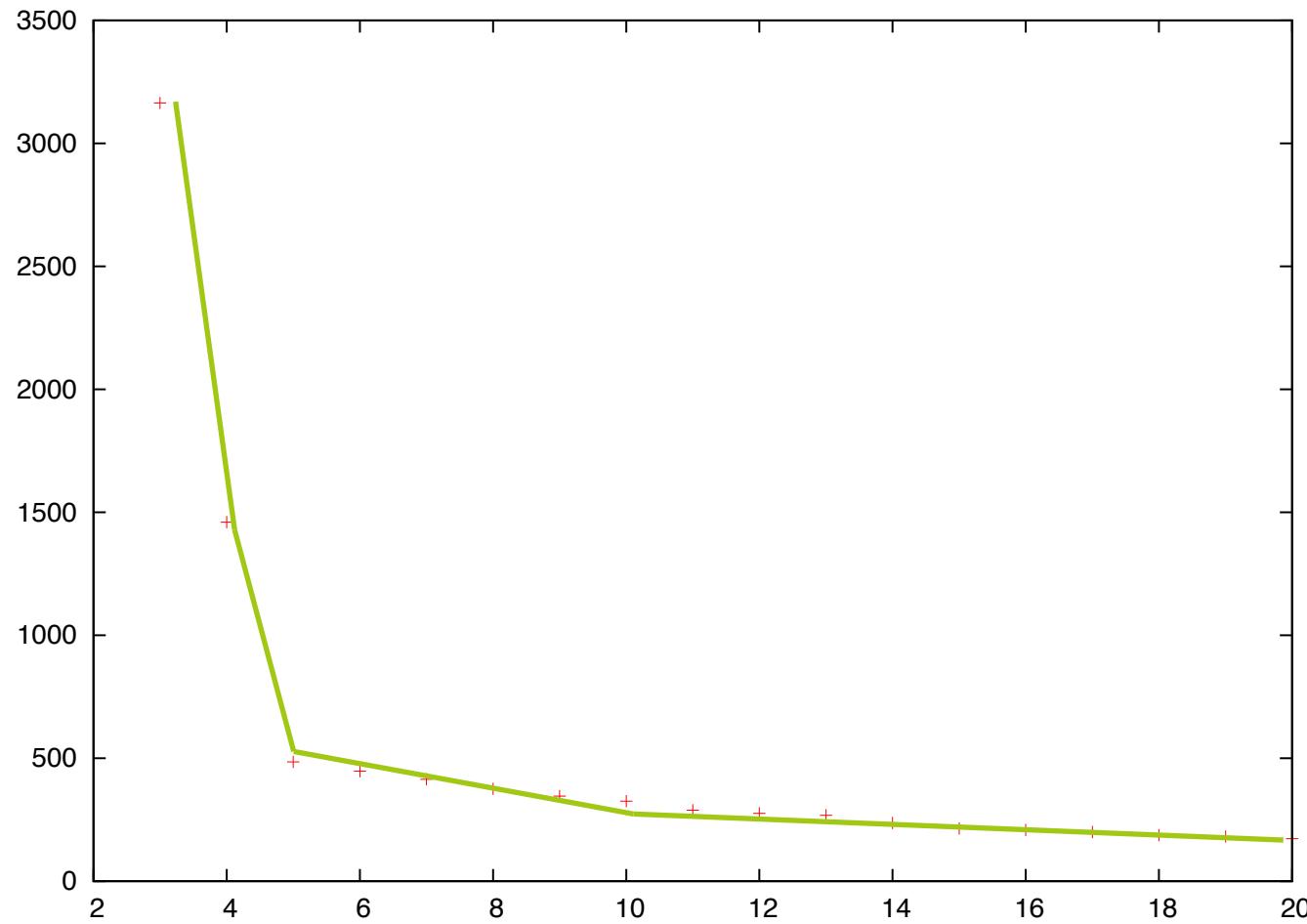


Classes majoritaires

4 Formes fortes pour K = 3



K et inertie intraclasse



Complexité

- Soit
 - T le nombre d'itérations
 - K le nombre de classes
 - N le nombre d'individus
- La complexité est de : $O(T K N)$
 - Complexité linéaire
 - Rappel CHA : $O(N^3)$

Variantes

- ❑ Mc Queen (67)
 - ❑ Recalculer les centres de gravité après chaque affectation
 - ❑ Convergence plus rapide
 - ❑ Mais l'ordre de traitement des individus n'est pas neutre

Variantes

- ❑ Mc Queen (67)
 - ❑ Recalculer les centres de gravité après chaque affectation
 - ❑ Convergence plus rapide
 - ❑ Mais l'ordre de traitement des individus n'est pas neutre
- ❑ K-médoïde, PAM (partition around medoids)
 - ❑ Le centre de gravité est remplacé par l'individu le plus central
 - ❑ Plus robustes au bruit et aux individus aberrants
- ❑ Nuées dynamiques (Diday 1972, 1974)
 - ❑ Le centre de gravité est remplacé par un ensemble d'individus

Variantes

- ❑ K-médoïde, PAM (partition around medoids)
 - ❑ Le centre de gravité est remplacé par l'individu le plus central
 - ❑ Plus robustes au bruit et aux individus aberrants
 - ❑ Mais couteux: $O(k(n-k)^2)$
- ❑ Algorithme
 - ❑ Initialize: randomly select k of the n data points as the medoids
 - ❑ Associate each data point to the closest medoid.
 - ❑ While the cost of the configuration decreases:
 - ❑ For each medoid m
 - ❑ for each non-medoid data point o:
 - ❑ Swap m and o, recompute the cost (sum of distances of points to their medoid)
 - ❑ If the total cost of the configuration increased in the previous step, undo the swap

Variantes

- K-median
 - La moyenne est remplacée par la médian
 - Moins sensible ou élém
 - Médian : valeur m qui permet de couper l'ensemble des valeurs en deux parties égales
 - Recherche de la median en triant les données : $O(n \log n)$

Modèles gaussiens

Gaussienne à une dimension

- Distribution gaussienne spécifiée par 2 paramètres

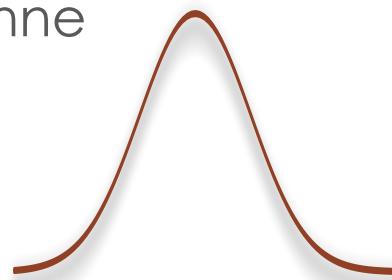
- la moyenne

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

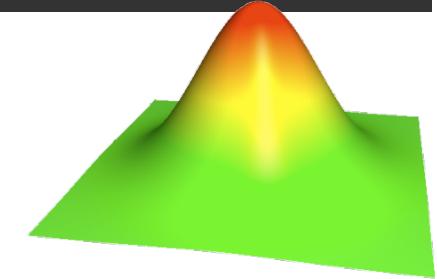
- σ : son écart type, σ^2 la variance

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N x_i^2 - \mu^2$$

- On parlera de gaussienne pour désigner une distribution de loi normale gaussienne



Gaussienne à n dimensions



- Distribution spécifiée par 2 paramètres
 - Moyenne, un vecteur à D dimensions

$$\mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_i \\ \vdots \\ \mu_d \end{bmatrix}$$

avec

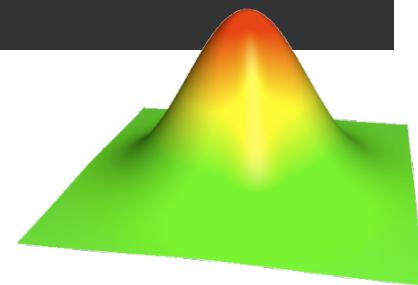
$$\mu_i = \frac{1}{N} \sum_{k=1}^N x_i^k$$

- Matrice de covariance

$$\Sigma = \begin{bmatrix} \sigma_{1,1} & \dots & \dots & \dots & \sigma_{1,d} \\ \vdots & \ddots & & & \vdots \\ \vdots & & \sigma_{i,j} & \sigma_{i,i} & \vdots \\ \vdots & & & \ddots & \vdots \\ \sigma_{d,1} & \dots & \dots & \dots & \sigma_{d,d} \end{bmatrix}$$

avec $\sigma_{i,j} = \frac{1}{N} \sum_{k=1}^N x_i^k x_j^k - \mu_i \mu_j$

Gaussienne à N dimensions



- ❑ Cas plein : Σ une matrice à $N \times N$ dimensions
 - ❑ Exprime la corrélation entre les dimensions
 - ❑ Matrice symétrique, positive
 - ❑ Cas diagonal : Σ une matrice à $N \times N$ dimensions avec des 0 sauf sur la diagonale
-
- ❑ Exemple
 - ❑ 13 attributs = dimensions 13

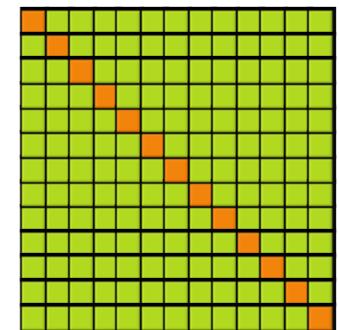
La moyenne :



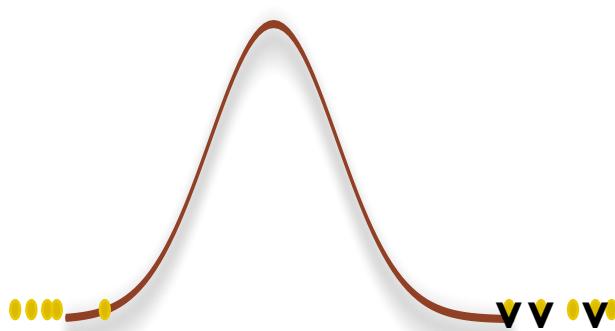
Vecteur à 13 dimensions

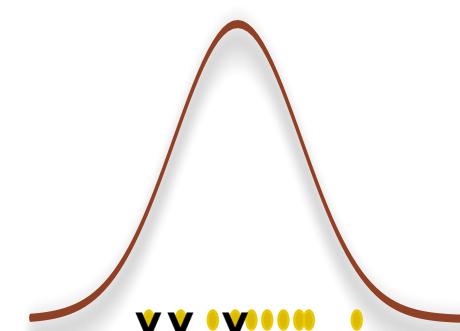
La covariance :

Une matrice à
13 x 13 dimensions



Gaussienne et vraisemblance

- Un échantillon d'observations issues d'une distribution
 - Peu vraisemblable que la distribution soit à l'origine de l'échantillon
 - Les observations sont dans une région où la densité de probabilité est faible
 - → petite valeur
- 
- Vraisemblable que l'échantillon soit issu de la distribution
 - Les observations sont dans une région où la densité de probabilité est forte
 - → grande valeur



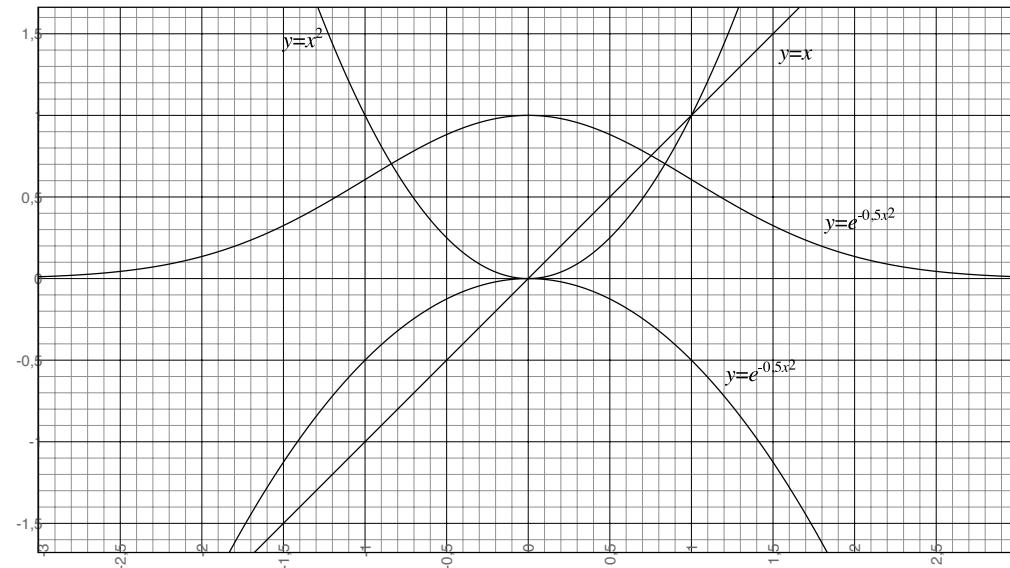
Gaussienne et vraisemblance

- Pour une gaussienne

$$L(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} e^{\frac{-1}{2}(x-\mu)^t \Sigma^{-1} (x-\mu)}$$

- Remarques $L(x|\mu, \Sigma) > 0$

- Plus $L(o|\mu, \Sigma)$ est grand plus il est vraisemblable que y soit issu de la gaussienne



Mixture de gaussiennes

- Une combinaison linéaire de M gaussiennes

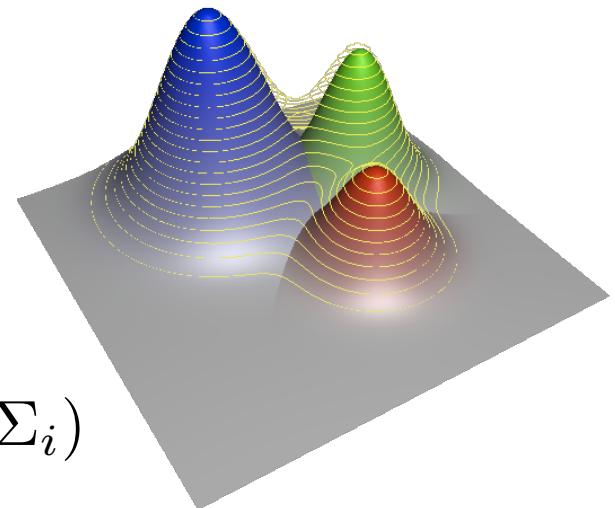
$$X = \{w_i, \mu_i, \Sigma_i\}, \forall i \in 1 \dots M$$

- w_i : le poids de la $i^{\text{ème}}$ composante

$$\sum_{i=1}^M w_i = 1$$

- La log vraisemblance

$$\log L(o|X) = \sum_{i=1}^M w_i \log L(o|\mu_i, \Sigma_i)$$



- Remarques

- La gaussienne peut être multidimensionnelle ou non
- Les matrices de covariance peuvent être diagonales ou pleines
- On parle de GMM : Gaussien Mixture Modèle

Classification par modèle

- On va remplacer
 - le centre de gravité par une gaussienne de paramètres μ et Σ
 - La distance euclidienne par la vaissamblance entre la gaussienne et chaque individu

Extension K-mean

- Étapes 1 :
 - Affecter chaque individu à la classe avec la plus grande vraisemblance
- Étapes 2 :
 - Mettre à jour les gaussiennes de chaque classe
 - En fonction des paramètres affectés, calculer la moyenne et la covariance
 - Le poids sera de : nombre de paramètres affectés / K
- Test d'arrêt
 - Reprendre à l'étape 1, tant que la vraisemblance entre deux itérations augmente de plus de ϵ
 - Sinon, on stop

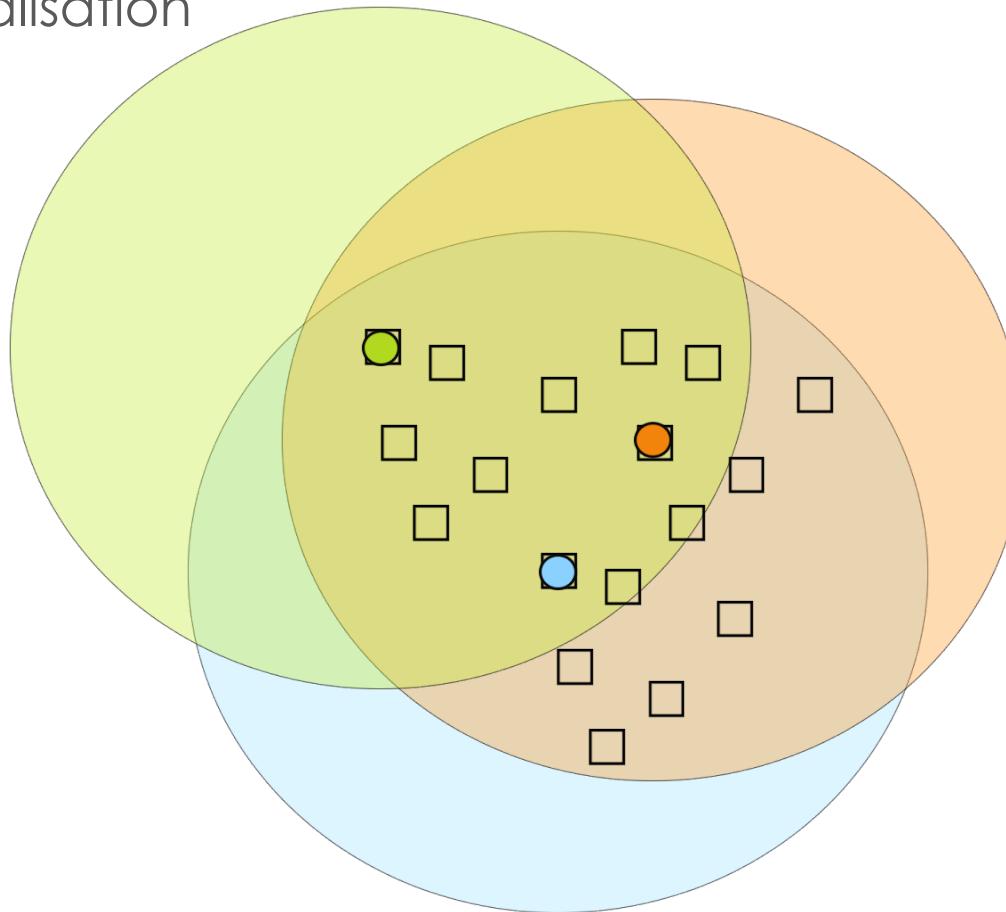
Exemple

■ Initialisation

$$\mathcal{W} = 1/3$$

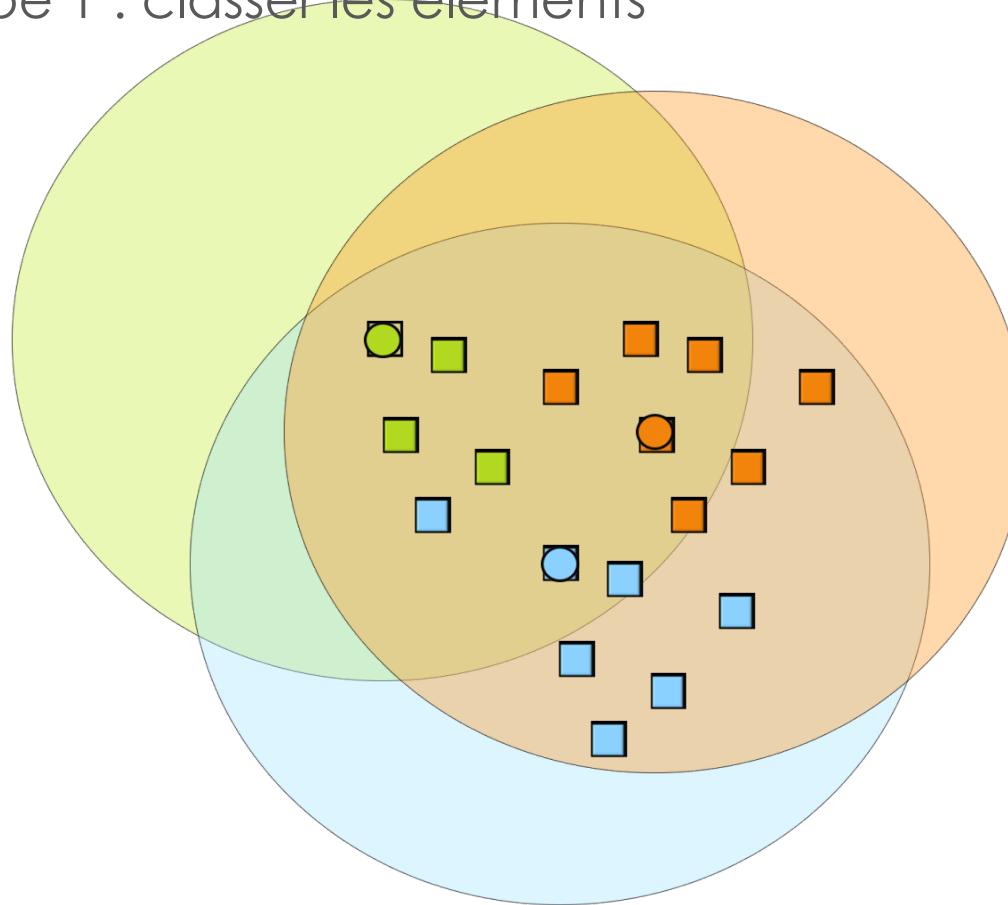
$$\mathcal{W} = 1/3$$

$$\mathcal{W} = 1/3$$



Exemple

■ Etape 1 : classer les éléments



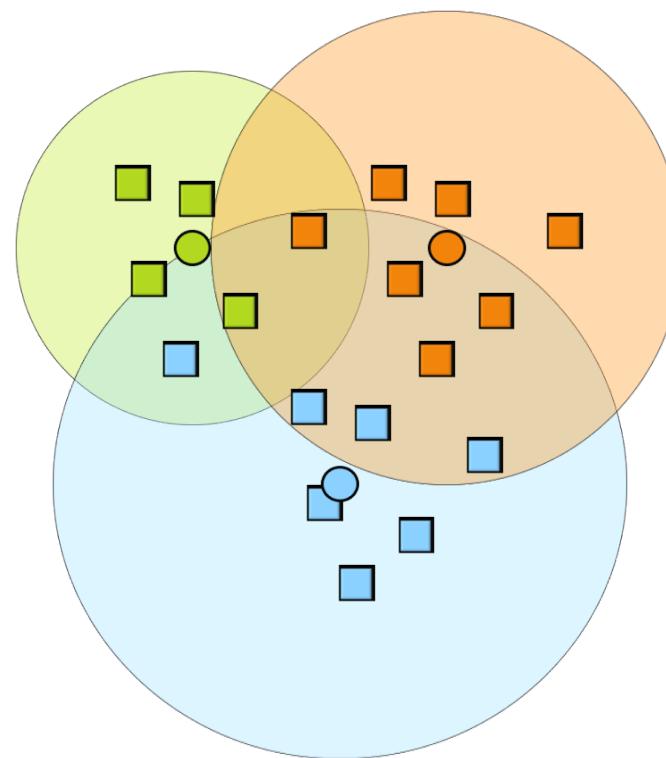
Exemple

- Etape 2 : mettre à jour les modèles

$$\mathcal{W} = 4/18$$

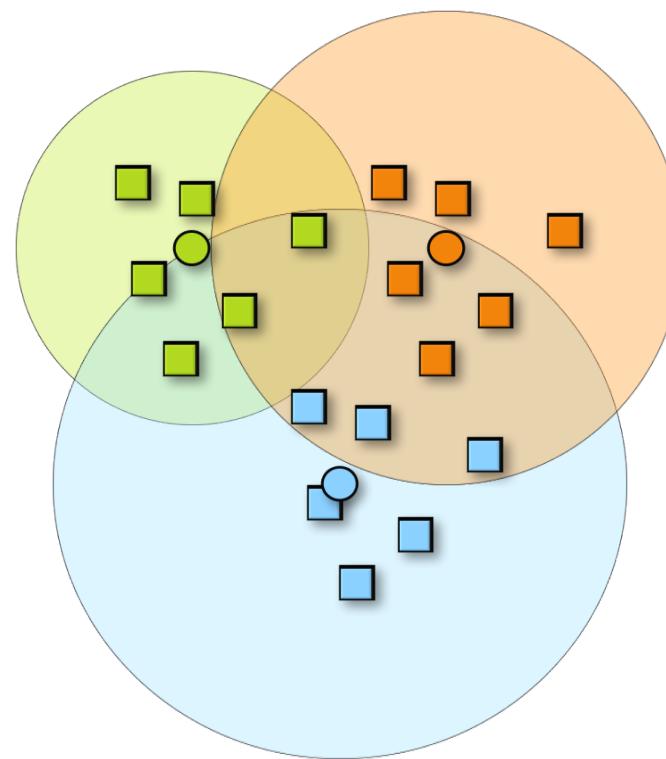
$$\mathcal{W} = 7/18$$

$$\mathcal{W} = 7/18$$



Exemple

■ Etape 1



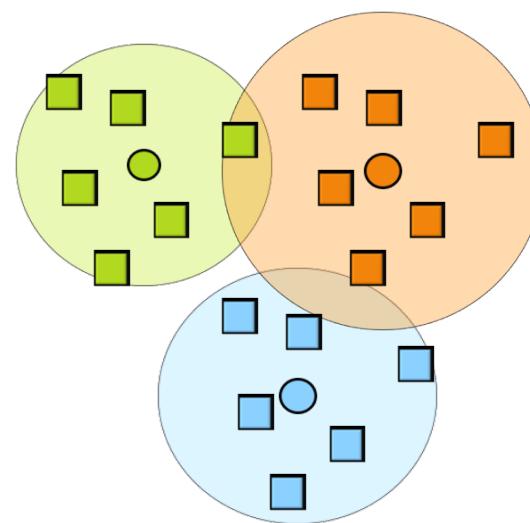
Exemple

■ Etape 2

$$\mathcal{W} = 6/18$$

$$\mathcal{W} = 6/18$$

$$\mathcal{W} = 6/18$$



GMM : EM

- ❑ EM : Expectation – Maximization
- ❑ But : trouver un maximum de vraisemblance d'un GMM
 - ❑ Minimum local
- ❑ Algorithme itératif, alterne entre 2 étapes
 - ❑ E : calcul l'espérance de la vraisemblance en tenant compte des dernières variables observées
 - ❑ M : estimation du maximum de vraisemblance des paramètres en maximisant la vraisemblance trouvée à l'étape E

- ❑ Converge vers un optimum local
- ❑ EM est un algorithme qui estime ces paramètres du GMM
- ❑ EM n'est donc pas spécifiquement un algorithme de classification, mais il peut être utilisé pour faire de la classification
- ❑ C'est une méthode de gradient : EM maximise la vraisemblance que les données résultent d'une certaine mixture

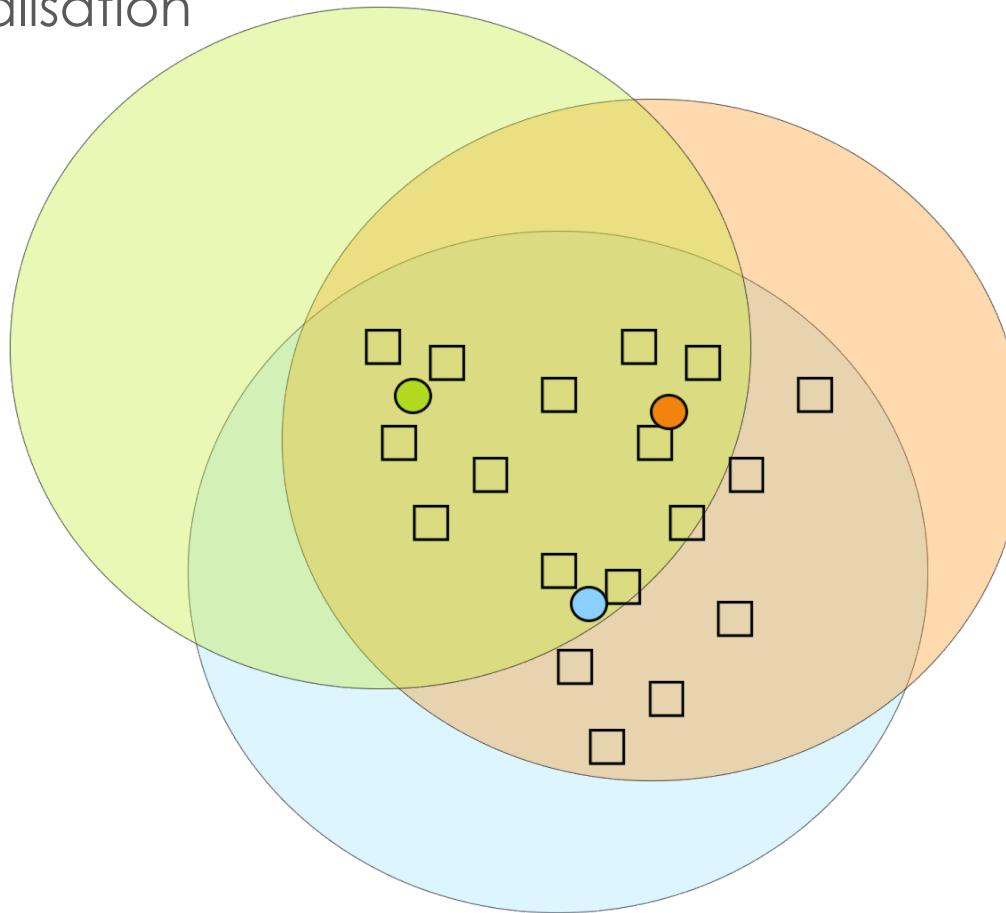
Exemple

■ Initialisation

$$\mathcal{W} = 1/3$$

$$\mathcal{W} = 1/3$$

$$\mathcal{W} = 1/3$$

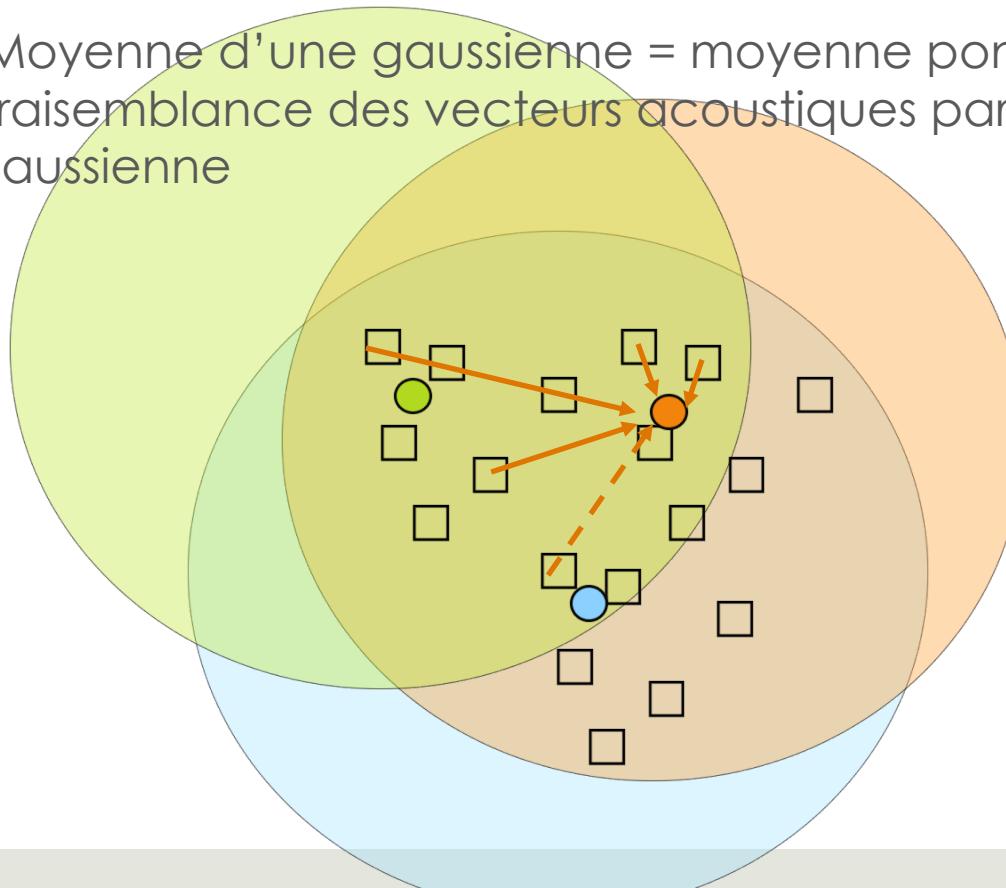


Exemple

■ Étape E

- Chaque vecteur acoustique contribue au calcul de chaque gaussienne
- Moyenne d'une gaussienne = moyenne pondérée par la vraisemblance des vecteurs acoustiques par rapport à la gaussienne

$$W = 1/3$$
$$W = 1/3$$
$$W = 1/3$$



EM v1

- Soit le modèle multigaussien (GMM) : $\mathcal{M} = \{w_g, \mu_g, \Sigma_g\}$,
 - l'indice de la gaussienne : $g \in [1..G]$
 - w_g , μ_g , Σ_g , sont le poids, la moyenne et la covariance de la gaussienne g
 - La somme des poids = 1
- La probabilité qu'un individu x_t soit émis par \mathcal{M} = à la somme pondérée des vraisemblances

$$p(x_t | \mathcal{M}) = \sum_{g=0}^G w_g p_g(x_t | \mu_g, \Sigma_g)$$

avec

$$p_g(x_t | \mu_g, \Sigma_g) = 2\pi^{-D/2} |\Sigma|^{-1/2} \exp\left(\frac{1}{2}(x_t - \mu_g)' \Sigma_g^{-1} (x_t - \mu_g)\right)$$

EM v1

- La probabilité d'un ensemble d'individus $x = \{x_1, \dots, x_t, \dots, x_T\}$

$$l(x|\mathcal{M}) = \sum_{t=0}^T \log p(x_t|\mathcal{M})$$

- Un GMM est estimé à partir d'un ensemble d'individus en optimisant le critère du maximum de vraisemblance
- L'algorithme est itératif
- A chaque itération i , on construit un nouveau modèle $\mathcal{M}^{(i)}$ en garantissant que

$$l(x|\mathcal{M}^{(i)}) > l(x|\mathcal{M}^{(i-1)})$$

EM v1

- Algorithme en deux étapes

- E : calcul des statistiques

- Contribution d'une gaussienne $\gamma_g^{(i)}$

$$\gamma_g^{(i)}(x_t) = \frac{w_g^{(i)} p_g(x_t | \mu_g^{(i)}, \Sigma_g^{(i)})}{\sum_{j=1}^G w_j^{(i)} p_j(x_t | \mu_j^{(i)}, \Sigma_j^{(i)})}$$

$$N_g^{(i)}(x) = \sum_{t=1}^T \gamma_g^{(i)}(x_t)$$

$$F_g^{(i)}(x) = \sum_{t=1}^T \gamma_g^{(i)}(x_t) x_t$$

$$S_g^{(i)}(x) = \sum_{t=1}^T \gamma_g^{(i)}(x_t) x_t x_t'$$

EM v1

- Etape de Maximisation : mise à jour du modèle à partir des statistiques

$$\begin{aligned}w_g^{(i+1)} &= \frac{1}{G} N_g^{(i)}(x) \\ \mu_g^{(i+1)} &= \frac{1}{N_g^{(i)}(x)} F_g^{(i)}(x) \\ \Sigma_g^{(i+1)} &= \frac{1}{N_g^{(i)}(x)} S_g^{(i)}(x) - \mu_g^{(i)} \mu_g^{(i)'}\end{aligned}$$

- Critère d'arrêt :
 - Nombre d'itérations
 - Gain en vraisemblance entre deux itérations inférieur à un seuil

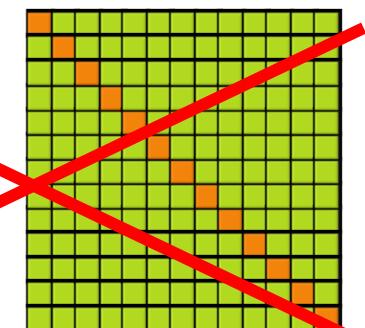
$$l(x|\mathcal{M}^{(i+1)}) - l(x|\mathcal{M}^{(i)}) < \delta$$

EM v2 : Etape E

```
for (int f = 0; f < P; f++) {  
    for(int g = 0; g < K; g++) {  
        L = vraisemblance (x[f], gmm[g]);  
        sw[g] += L;  
        sgw += L;  
        for (int i = 0; i < N; i++) {  
            sm[g][i] += L * x[f][i];  
            sc[g][i] += L * x[f][i] * x[f][i];  
        }  
    }  
}
```

La moyenne : 
Vecteur à n dimensions

La covariance :
Une matrice à
N x N dimensions



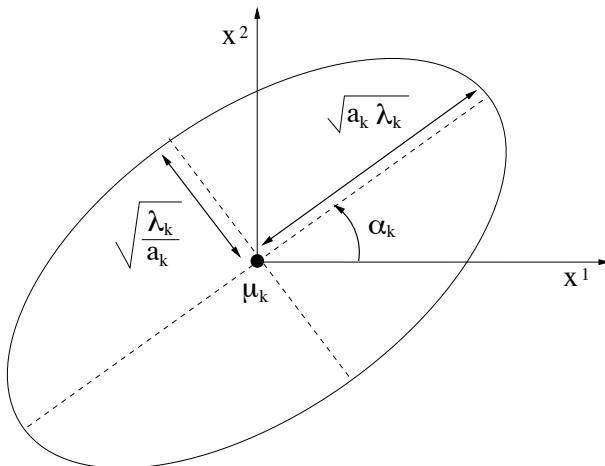
Ici la variance : 
Vecteur à N dimensions

EM v2 : Étape M

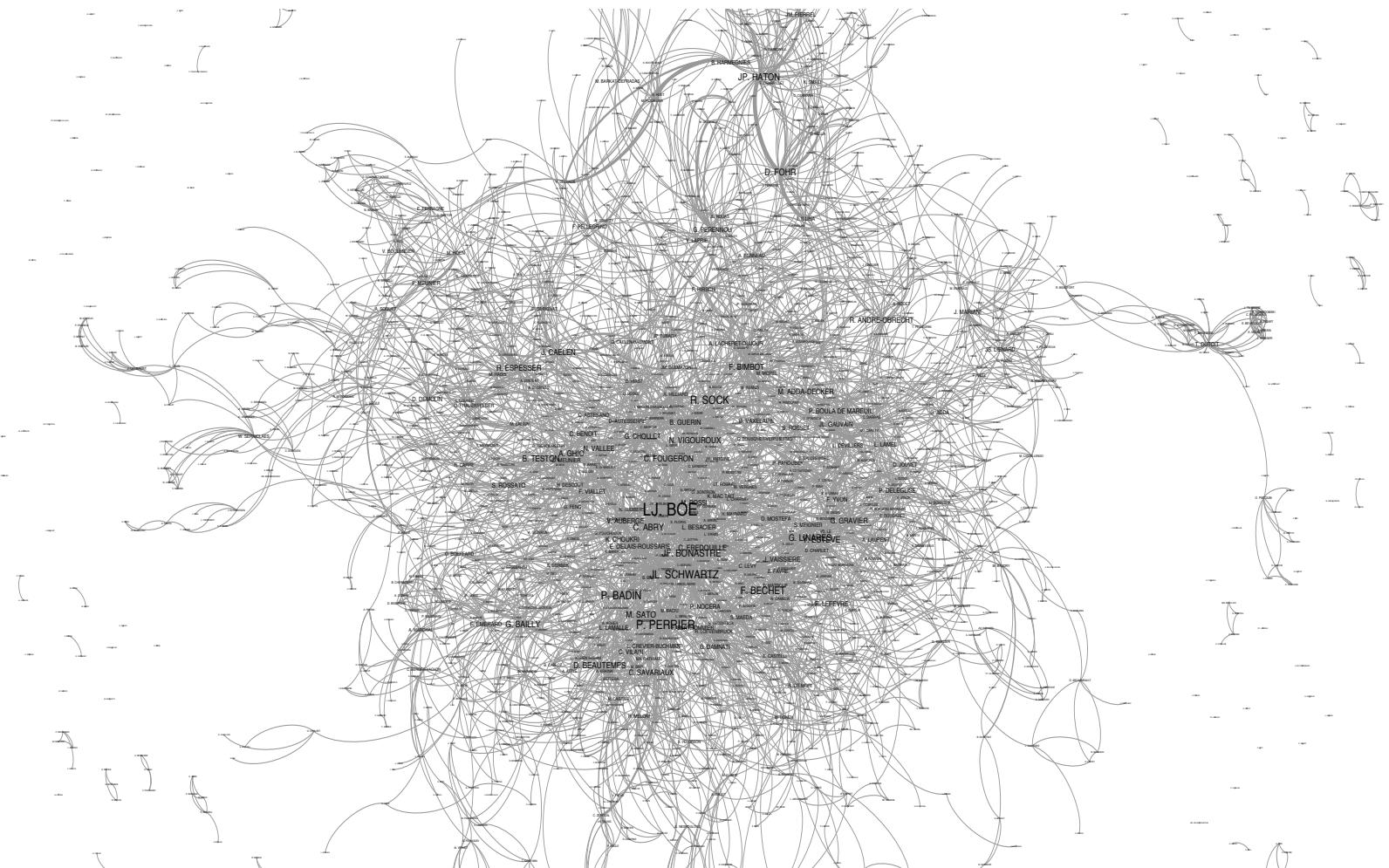
```
for(int g = 0; g < K; g++) {  
    w[g] = sw[g] / sgw;  
    for (int i = 0; i < N; i++) {  
        m[g][i] = sm[g][i] / sw[g];  
        c[g][i] = sc[g][i] / sw[g] - m[g][i]*m[g][i];  
    }  
}
```

Différentes covariances

$$\Sigma_k = \underbrace{\lambda_k}_{\text{volume}} \cdot \underbrace{\mathbf{D}_k}_{\text{orientation}} \cdot \underbrace{\mathbf{A}_k}_{\text{forme}} \cdot \mathbf{D}'_k$$



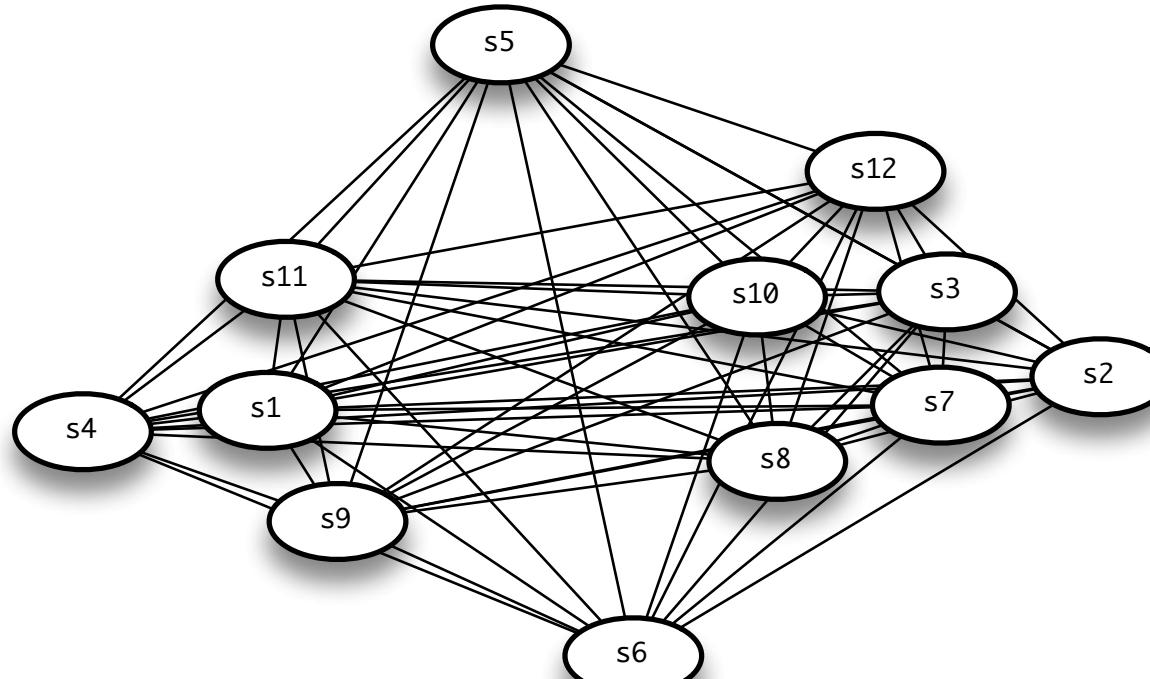
$\pi_{(k)} \lambda I$	$\pi_{(k)} \lambda_k I$
$\pi_{(k)} \lambda B$	$\pi_{(k)} \lambda_k B$
$\pi_{(k)} \lambda B_k$	$\pi_{(k)} \lambda_k B_k$
$\pi_{(k)} \lambda C$	$\pi_{(k)} \lambda_k C$
$\pi_{(k)} \lambda C_k$	$\pi_{(k)} \lambda_k C_k$
$\pi_{(k)} \lambda D A_k D$	$\pi_{(k)} \lambda_k D A_k D$
$\pi_{(k)} \lambda D_k A D_k$	$\pi_{(k)} \lambda_k D_k A D_k$



Classification et graphe

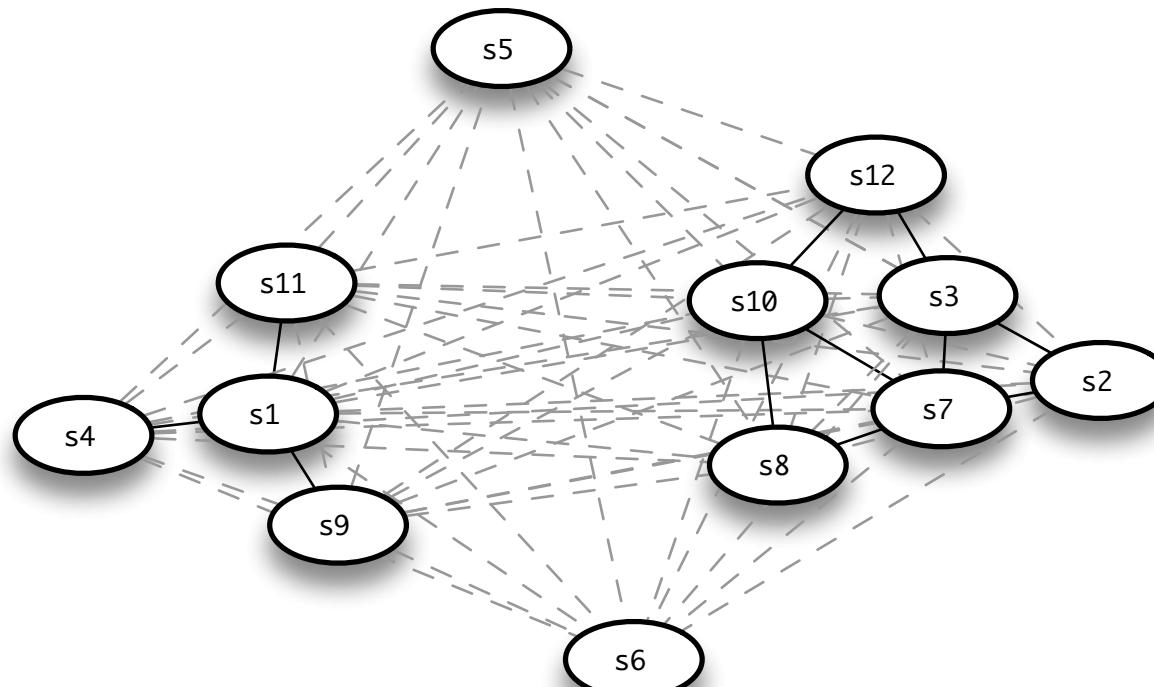
Matrice de distance et graphe

- Chaque noeud = un individu
- chaque lien = la distance entre deux individus



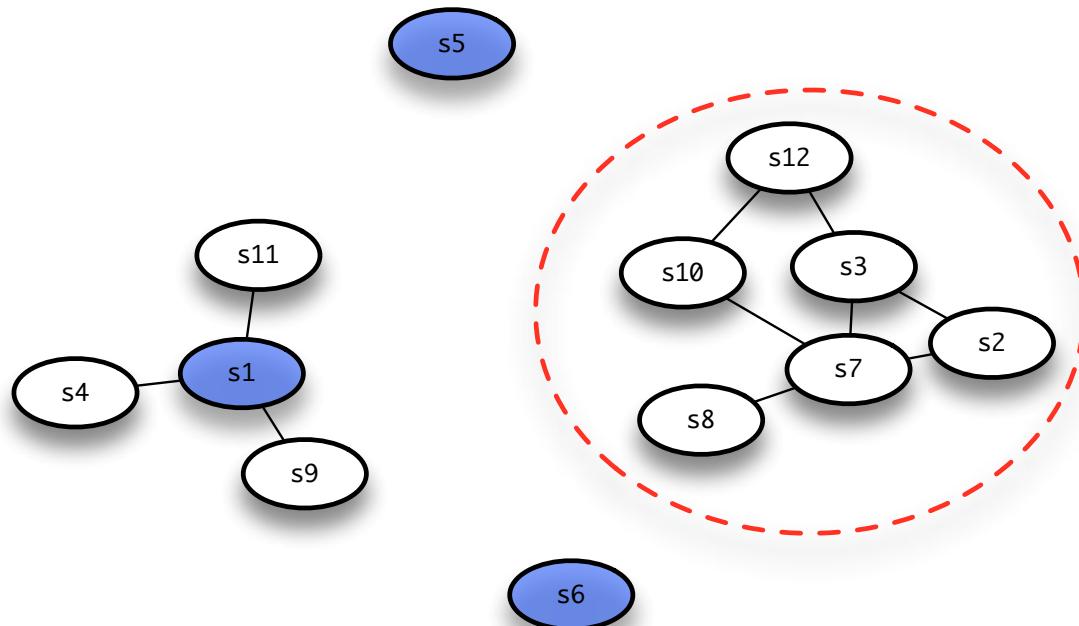
Graphe et seuil

- On grade les liens dont la distance est inférieure à un seuil



Graphe et composantes connexes

- On cherche les composantes connexes = sous graphe
- On cherche les graphes en étoile



Graphe, arbre et classification hiérachique

- Arbre : un graphe connexe et sans cycle

Classification exhaustive

Formulation ILP

- On a une matrice de distance entre nos individus
- On cherche k individu qui seront les centres des classes de manière à ce que les individus de la classe soit à une distance inférieure à un seuil

$$\text{Minimiser : } \sum_{k \in C} x_{k,k} + \frac{1}{D+1} \sum_{j \in C} \sum_{k \in K_j} d(k,j) x_{k,j}$$

sous les contraintes :

$$x_{k,j} \in \{0, 1\}, \forall k \in K_j, \forall j \in C$$

$$\sum_{k \in K_j} x_{k,j} = 1, \forall j \in C$$

$$x_{k,j} - x_{k,k} \leq 0, \forall k \in K_j, \forall j \in C$$

Exemple

Minimize :

$$X_{1,1} + X_{2,2} + X_{3,3} + X_{4,4} + X_{5,5} + X_{6,6} + X_{7,7} + X_{8,8} + 0.49 X_{1,6} + 0.49 X_{6,1}$$

Subject To

#Contrainte 1

$$X_{1,1} + X_{1,6} = 1$$

$$X_{2,2} = 1$$

$$X_{3,3} = 1$$

$$X_{4,4} = 1$$

$$X_{5,5} = 1$$

$$X_{6,6} + X_{6,1} = 1$$

$$X_{7,7} = 1$$

$$X_{8,8} = 1$$

#Contrainte 2

$$X_{1,6} - X_{6,6} \leq 0$$

$$X_{6,1} - X_{1,1} \leq 0$$

No.	Column name	Activity	Lower bound	Upper bound
1	X1,1 *	1	0	1
2	X2,2 *	1	0	1
3	X3,3 *	1	0	1
4	X4,4 *	1	0	1
5	X5,5 *	1	0	1
6	X6,6 *	0	0	1
7	X7,7 *	1	0	1
8	X8,8 *	1	0	1
9	X1,6 *	0	0	1
10	X6,1 *	1	0	1

Inertie et variance

Inertie avec distance euclidienne

$$G = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\mathcal{I} = \sum_{i=1}^N d(x_i, G)^2$$

$$d(x_i, G) = \sqrt{\sum_{k=1}^P (x_i^k - G^k)^2}$$

$$\mathcal{I} = \sum_{i=1}^N \sum_{k=1}^P (x_i^k - G^k)^2$$

Inertie et variance

$$\mathcal{I} = \sum_{i=1}^N \sum_{k=1}^P (x_i^k - G^k)^2$$

$$G = \mu = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\mathcal{I} = \sum_{i=1}^N \sum_{k=1}^P (x_i^k - \mu^k)^2$$

$$\mathcal{I} = \sum_{i=1}^N (x_i - \mu)^2 \quad Var = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

Distance de Mahalanobis

■ Modèle et distance de Mahalanobis

$$G = \mu = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\mathcal{I} = \sum_{i=1}^N d(x_i, \nu)^2$$

$$d_m(x_i, G) = \sqrt{(x_i - \mu)^t \Sigma (x_i - \mu)}$$

$$\mathcal{I} = \sum_{i=1}^N (x_i - \mu)^t \Sigma (x_i - \mu)$$

$$\mathcal{I} = \sum_{i=1}^N (x_i - \mu)^t \Sigma (x_i - \mu)$$

$$\mathcal{I} = \sum_{i=1}^N (x_i - \mu)^2 \quad Var = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

$$\mathcal{I} = \sum_{i=1}^N (x_i - \mu)^t (x_i - \mu)$$

$$\mathcal{I} = \sum_{i=1}^N (x_i - \mu)^t I(x_i - \mu)$$