Introduction à l'Apprentissage Automatique

- Chapitre 1 : Introduction aux méthodes de décision
- Chapitre 2 : Classification non supervisée automatique
  - Classification non hiérarchique : Algorithme des k-moyennes
  - Classification hiérarchique : Arbre de classification
- Chapitre 3 : Approche discriminante linéaire
  - Problème à deux classes
  - Hyperplan séparateur
  - Neurone formel
- Chapitre 4 : Approche statistique
  - Maximum de vraisemblance
  - Cas gaussien
  - Règle du plus proche voisin

Chapitre 2
Classification non supervisée automatique

$$Y = \{y_1, y_2, \ldots, y_N\}, y_i \in R^d$$

~ trouver une partition du nuage telle qu'au sein de chaque partie les individus se ressemblent et puissent être assimilés à une classe

~ trouver des prototypes ou références représentatives de l'ensemble

une partie = une classe = une référence = un prototype

I Algorithme de quantification vectorielle ou algorithme des « K means » ou Algorithme des Nuées Dynamiques

Hypothèses de base :

- Nombre de classes « connu » et égal à K
- Existence d'une distance euclidienne (notion de centre de gravité), noté d

Vocabulaire de base :

- Dictionnaire (codebook) : ensemble de points références de R^d,
- Référence = prototype = codeword (élément de R^d)

Formulation du problème : Recherche du ***meilleur*** dictionnaire pour représenter le nuage de points initial Y

Meilleur dictionnaire

ou meilleure partition ?

$$\frac{1}{K!}\sum_{i=1}^{K}\left(C_i^K\right)(-1)^{K-i}i^N$$

→Nombre de partitions de K parties pour un nuage de N individus

→ Recherche sous optimale à partir d'un critère

Algorithme des K-means

Critère de recherche de D = {d$_1$, …, d$_K$}

$$Crit(D) = \sum_{n=1}^{N} d(y_n, d_{\hat{n}})^2$$

$$\hat{D} = \arg\min_{D} Crit(D)$$

$$\hat{n} = \arg\min_{1 \le k \le K} d(y_n, d_k)$$

Exemple 1 : Soit le nuage de points de R^2 formé des 4 points

$$x_1 = \begin{pmatrix} 4 \\ 5 \end{pmatrix}, x_2 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}, x_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, x_4 = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$$

1) $Y_1 = \{x_1, x_2\}, \quad Y_2 = \{x_3, x_4\}$
2) $Y_1 = \{x_1, x_4\}, \quad Y_2 = \{x_3, x_2\}$
3) $Y_1 = \{x_1, x_2, x_3\}, \quad Y_2 = \{x_4\}$

Quelle est la meilleure partition au sens de ce critère?

Relation entre une partition de Y et un dictionnaire

1. Soit D = $\{d_1, \ldots, d_K\}$ un dictionnaire de K éléments, connu

→ calcul de la partition associée $Y_D = \coprod_{k=1,K} Y^D{}_k$

$$Y^D{}_k = \{y_n \in Y \,/\, d(y_n, d_k) \prec d(y_n, d_j), j \neq k\}$$

$Y_k{}^D$ est une classe associée à la référence ou prototype d_k

2. A partir d'une partition connue,

$$Y = \coprod_{k=1,K} Y_k$$

$$Crit(D^{Y^D}) \leq Crit(D)$$

→ calcul d'un dictionnaire associé

$D^Y = \{d_1{}^Y, \ldots, d_K{}^Y\}$

où $d_k{}^Y$ est le centre de gravité de la partie Y_k

Algorithme de quantification vectorielle ou K-means:

Soit D_0, un dictionnaire initial de K éléments, $t = 0$.
Soit D_{t-1}, le dictionnaire obtenu avant la $t^{ième}$ itération

1. Définition de la partition associée au dictionnaire D_{t-1},

$$Y = \coprod_{k=1,K} Y_k$$

2. Calcul des centres de gravité de chacune des parties de cette partition pour former un nouveau dictionnaire $D_t = \{d^t_1, \ldots, d^t_K\}$

3. Calcul de la valeur du *Crit(D_t)*.

4. Utiliser le test d'arrêt : si $(Crit(D_{t-1}) - Crit(D_t)) / Crit(D_{t-1}) \leq \lambda$

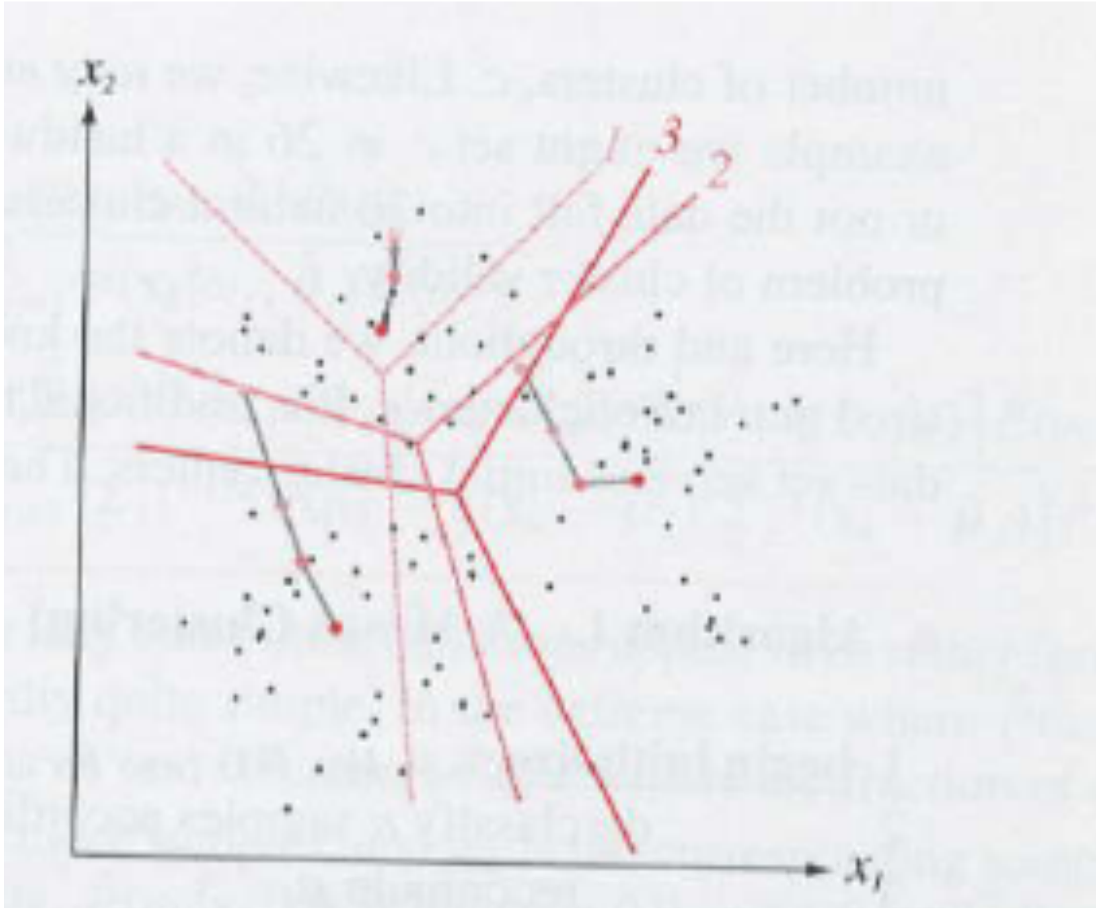
le dictionnaire optimal est trouvé, $D_{final} = D_t$.
sinon $t = t+1$, et retour à l'étape 1.

Exemple de simulation (3 itérations) dans R^2:

• K=3 ,

• Evolution des 3 partitions

• Evolution des 3 centres de gravité
  (du plus clair au plus foncé)



Exemple 2 : Soit le nuage de points de R^2 formé des 4 points

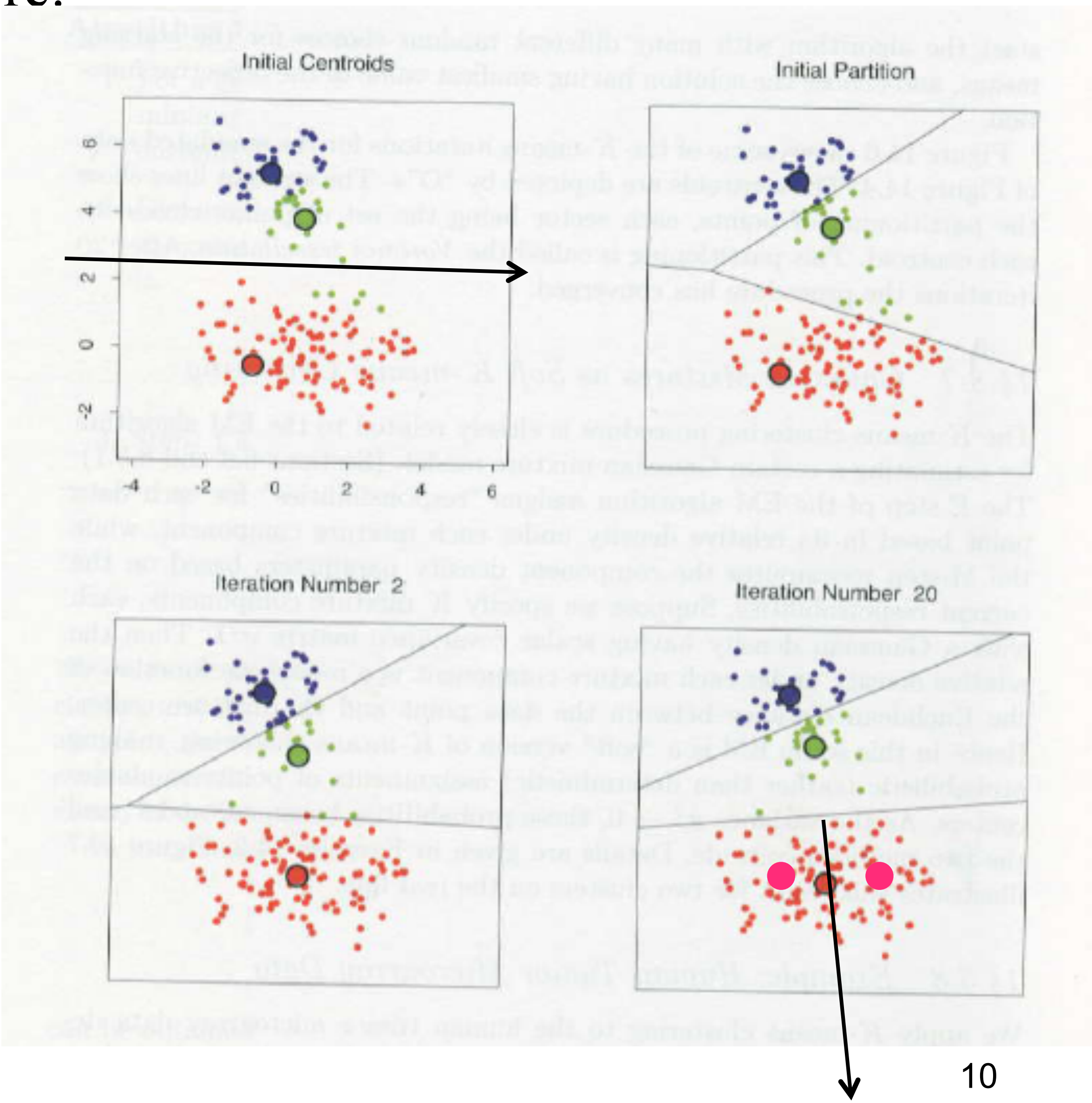
$$x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, x_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, x_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, x_4 = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}$$

Quelle est la meilleure partition obtenue par l'algorithme des K-means en prenant comme dictionnaire initial, les dictionnaires associés aux partitions définies ci-dessous ?

1) $Y_1 = \{x_1, x_2\}, \quad Y_2 = \{x_3, x_4\}$
2) $Y_1 = \{x_1, x_4\}, \quad Y_2 = \{x_3, x_2\}$
3) $Y_1 = \{x_1, x_2, x_3\}, \quad Y_2 = \{x_4\}$

Problèmes de mise en œuvre:

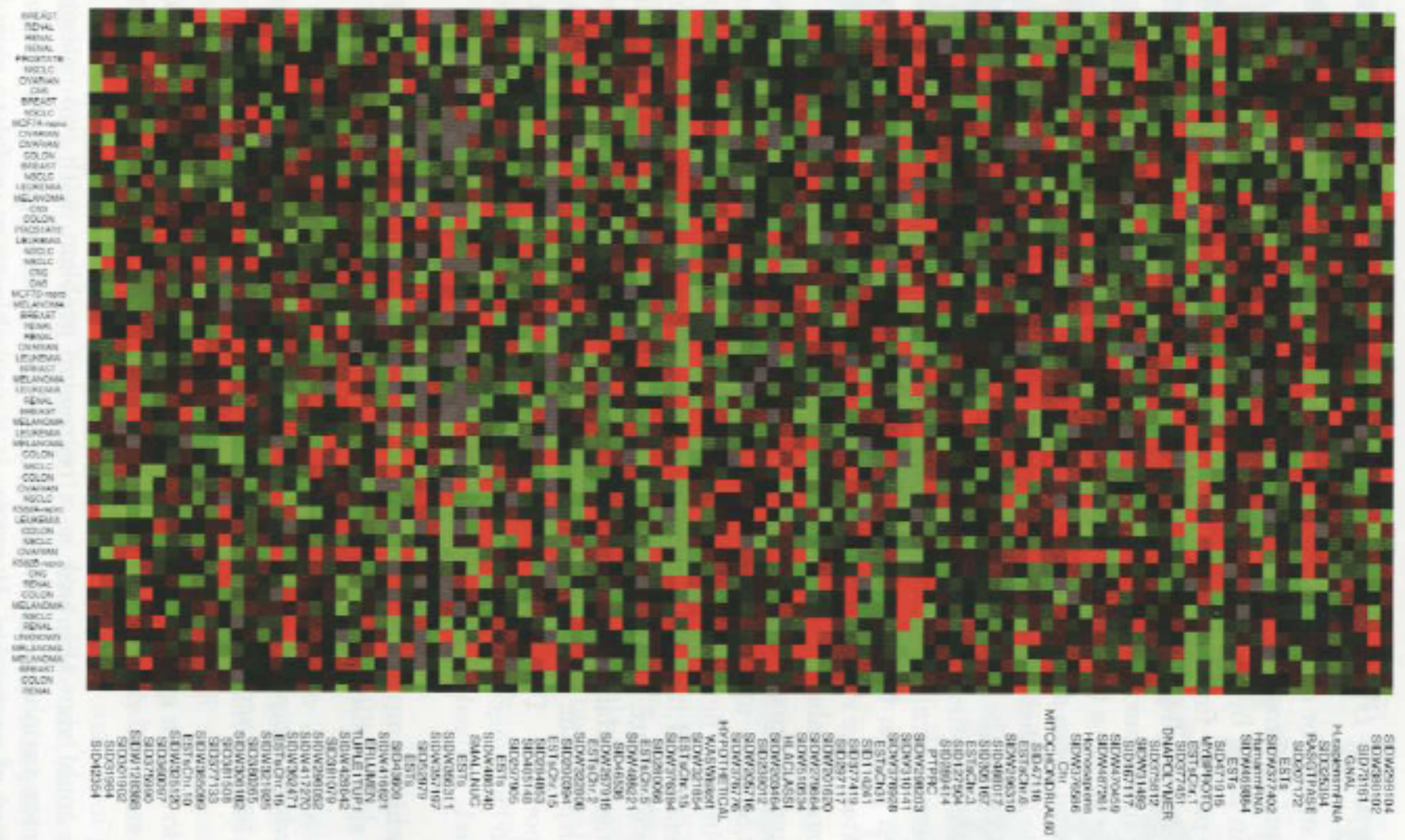
- Que choisir pour K ?
- Quel seuil λ ?
- Nombre d'itérations?
- Exemple K= 3



Pourquoi pas K=2 ?

$\boxed{K = 4\ ?}$

Exemple en Bioinformatique



• • •

Matrice des individus

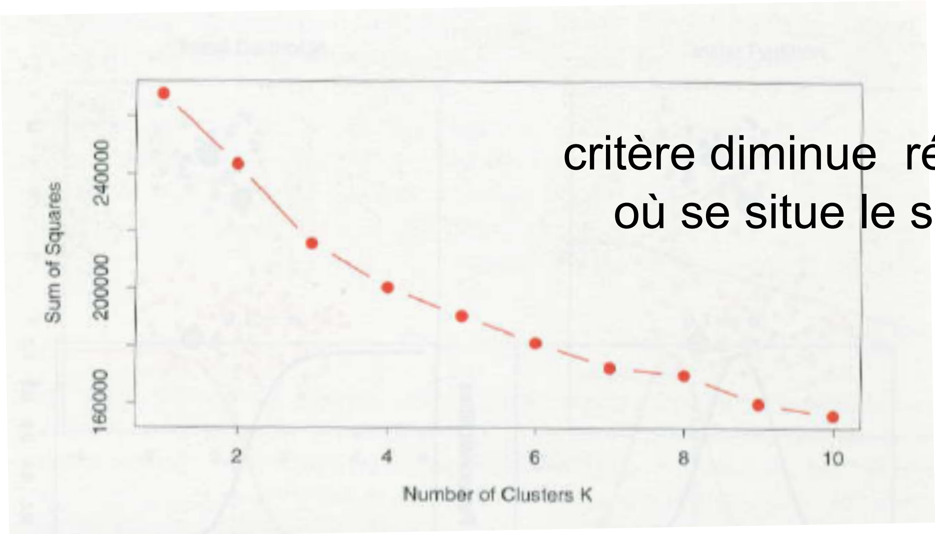
•64 lignes = 64 individus du nuage de points dans R^{6830}
•une ligne = une analyse génétique d'un individu atteint d'une tumeur (nom de la tumeur à gauche)
•une colonne = le résultat de l'analyse d'un des 6830 gènes

OBJECTIF : trouver dans cet espace les tumeurs qui donnent les mêmes résultats d'analyse génétique

TABLE 14.2. *Human tumor data: number of cancer cases of each type, in each of the three clusters from K-means clustering.*

Cluster	Breast	CNS	Colon	K562	Leukemia	MCF7
1	3	5	0	0	0	0
2	2	0	0	2	6	2
3	2	0	7	0	0	0
Cluster	**Melanoma**	**NSCLC**	**Ovarian**	**Prostate**	**Renal**	**Unknown**
1	1	7	6	2	9	1
2	7	2	0	0	0	0
3	0	0	0	0	0	0

Mais :



critère diminue régulièrement =>
où se situe le seuil optimal??

II Classification automatique hiérarchique – Arbres de décision

Hypothèse de base : Nombre de classes « inconnu »

Idée :

- construire un partitionnement hiérarchique
  ou une suite de partitions « emboitées »

Partition de K éléments

$$Y = \coprod_{k=1}^{K} Y_k$$

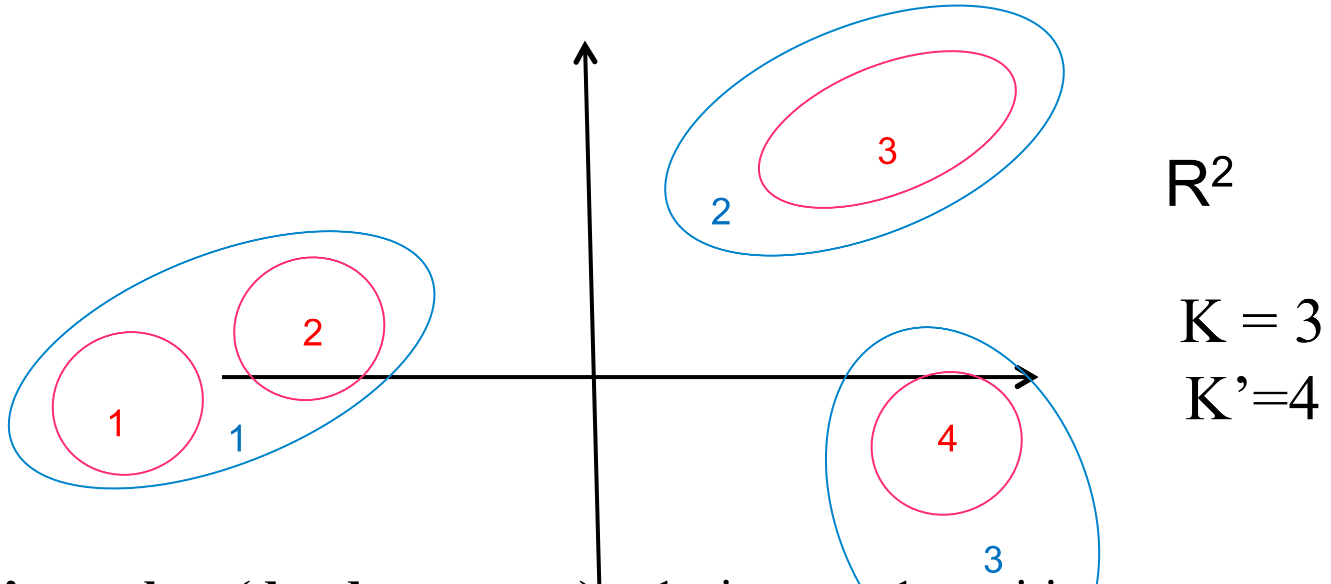
Partitions de K' éléments, K'>K

$$Y = \coprod_{k'=1}^{K'} Y'_{k'}$$

$$\forall k', \exists k \ tel\ que\ Y'_{k'} \subseteq Y_k$$

- déterminer le nombre de classes a posteriori

Illustration



Construction d'un arbre (dendogramme) : 1 niveau = 1 partition

Niveau 1 Y

Niveau 3 1 2 3

Niveau 4 1 2 3 4

Niveau N $\{y_1\}$, $\{y_2\}$, $\{y_k\}$, $\{y_N\}$

→Algorithme de construction d'arbres, ascendant ou descendant

→Mesure pour décider de grouper deux parties ou scinder une partie en 2

II.1 Mesures de dissimilarité entre points

Mesures de similarité entre individus de R^d

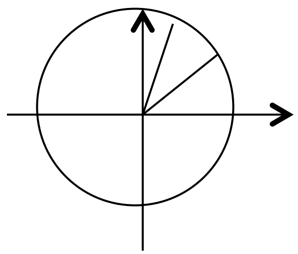
$\sigma : R^d \times R^d \rightarrow R^+$

$\sigma(x_1, x_2) >= 0$

σ est symétrique

$\sigma(x_1, x_2) < \sigma(x_1, x_1)$, $x_1 \neq x_2$

Exemple : σ : $C(0,1)xC(0,1) \rightarrow R^+$
(points sur le cercle unité)
$(x_1, x_2) \rightarrow \cos(x_1, x_2)$



Plus $\sigma(x_1, x_2)$ augmente, plus x_1 et x_2 se ressemblent

Mesure de dissimilarité associée à une mesure de similarité (bornée)

Soit σ une mesure de similarité bornée par $\sigma_{\max} = \max_{x_1, x_2} \sigma(x_1, x_2)$

$$\boxed{dis(x_1,x_2) = \sigma_{\max} - \sigma(x_1, x_2)}$$

Attention : l'inégalité triangulaire n'est pas assurée

Distances :

✓ distance euclidienne (invariante par translation et rotation)

✓ distance quadratique : $d^2(x_1, x_2) = (x_1 - x_2)^t M(x_1 - x_2)$

avec M matrice d x d définie positive symétrique

✓ distance de Mahalanobis : distance quadratique avec M = Σ^{-1} ,

Σ la matrice de covariance d'un nuage de points

✓ distance de Minkowsky : $d(x_1, x_2) = (\sum_{k=1}^{d} \left| x_1^k - x_2^k \right|^q)^{1/q}, \quad q \geq 1$

avec x_i^k la $k^{ième}$ coordonnée de x_i.

II.2 Mesures de dissimilarité entre nuages de points

Soit $(A_j)_{j=1,J}$ un ensemble de J nuages de points de R^n ,
soit N_i le cardinal de A_i et soit d une distance sur R^n.

- distance min : $d(A_i, A_j) = \min_{x \in A_i, y \in A_j} d(x, y)$

- distance max : $d(A_i, A_j) = \max_{x \in A_i, y \in A_j} d(x, y)$

- distance moyenne : $d(A_i, A_j) = \frac{1}{N_i N_j} \sum_{x \in A_i, y \in A_j} d(x, y)$

- distance entre les moyennes : $d(A_i, A_j) = d(g_i, g_j)$

avec g_i le centre de gravité
(ou moyenne) du nuage A_i.

Inertie d'un nuage A de points autour d'un point a (N_A est le cardinal de A):

$$I_a(A) = \frac{1}{N_A} \sum_{x \in A} d^2(x, a)$$

Si a est le centre de gravité g de A, $I_a(A) = I(A)$ est l'inertie du nuage A.

Théorème de Huyghens : Soient $(A_j)_{j=1,J}$ un ensemble de J nuages de points de R^n, N_j le cardinal de A_j, g_j le centre de gravité de A_j, les nuages sont disjoints deux à deux. Alors :

$$I_a(\bigcup_j A_j) = \sum_j I(A_j) + \sum_j N_j d^2(g_j, a)$$

Interprétation : Si a est le centre de gravité de l'union des A_j, cette formule s'interprète comme :

Inertie totale = inerties intra-classes + inerties interclasses

Cas particulier : J=2 et g centre de gravité de $A_1 \cup A_2$

$$I(A_1 \cup A_2) = I(A_1) + I(A_2) + N_1 d^2(g_1, g) + N_2 d^2(g_2, g)$$

On montre que : $N_1 d^2(g_1, g) + N_2 d^2(g_2, g) = \dfrac{N_1 N_2}{N_1 + N_2} d^2(g_1, g_2)$

→Définition de la pseudo distance (moment intra nuage) :

$$\boxed{psd(A_i, A_j) = \frac{N_i N_j}{N_i + N_j} d^2(g_i, g_j)}$$

Interprétation :

Inertie totale = inerties intra-classes + inerties interclasses

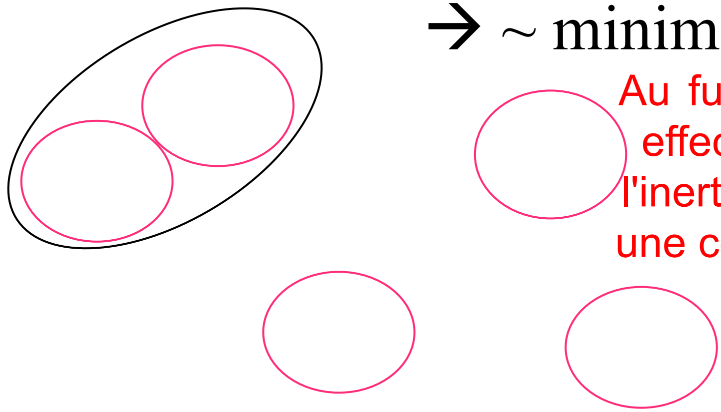
$$I(A_1 \cup A_2) = I(A_1) + I(A_2) + \frac{N_1 N_2}{N_1 + N_2} d^2(g_2, g_1)$$

Ensemble de parties $A_1, \ldots, A_J$:

- réunion des parties ayant, après union, l'inertie interne minimale

→minimisation de la pseudo distance

→ ~ minimisation de l'interclasse



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

Interprétation :

Inertie totale = inerties intra-classes + inerties interclasses

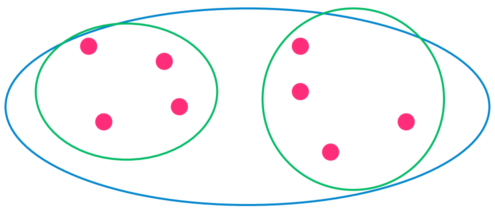
$$I(A_1 \cup A_2) = I(A_1) + I(A_2) + \frac{N_1 N_2}{N_1 + N_2} d^2(g_2, g_1)$$

Scinder une partie en deux parties A_1, A_2 :

• minimiser l'inertie interne de chacune des parties

→ maximiser la pseudo distance

→ ~ maximiser l'interclasse



Attention : tous les couples de parties doivent être testés!!!

II.3 Construction hiérarchique d'un ensemble de classes sur Y

$$Y = \left\{ y_1, y_2, \ldots, y_N \right\}, y_i \in R^n$$

1. Hiérarchie ascendante
   - méthode des distances → choix d'une distance
   - méthode des moments d'ordre 2 (psd)

2. Hiérarchie descendante
   - méthode des moments d'ordre 2 (psd)

Algorithme ascendant = regroupement de parties

Processus itératif :

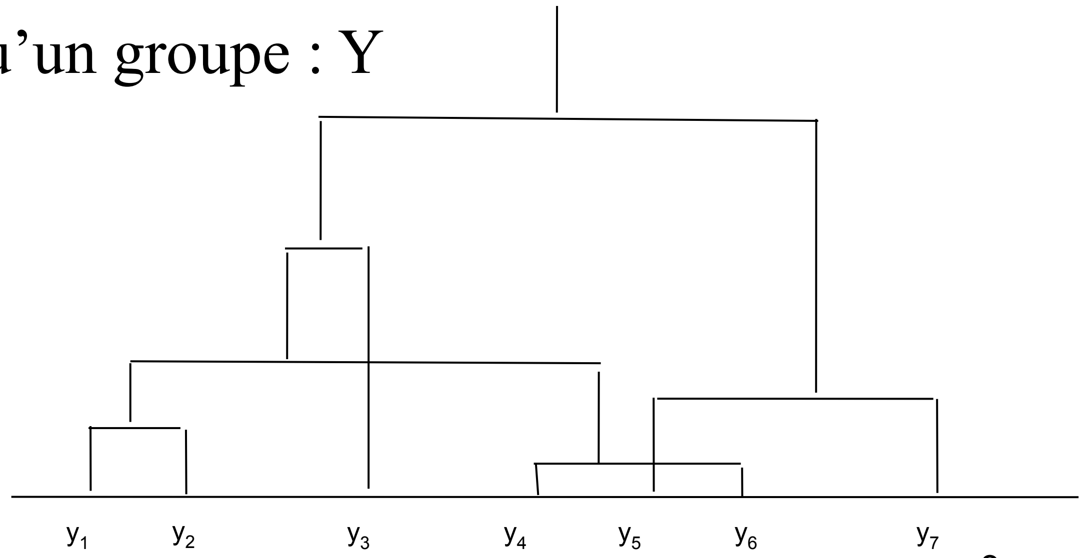
- T=0, chaque groupe est constitué d'un seul point de Y
  → N groupes.

- A chaque itération, regroupement des deux groupes les plus proches au sens de la dissimilarité d.

- Arrêt lorsqu'il n'y a plus qu'un groupe : Y

Construction simultanée du dendogramme

Choix du niveau pour définir K



Algorithme descendant = division de parties

Processus itératif :

- T=0, un groupe est constitué de Y

- A chaque itération
  - choix d'un groupe à scinder
  - scinde le groupe en 2

2 questions très difficiles

- Arrêt lorsqu'il y a N groupes

Algorithme descendant = division de parties

A chaque itération

- choix d'un groupe à scinder → Critère de choix :
  - inertie la plus forte
  - cardinal le plus élevé
- scinde le groupe en 2

Méthode descendante des moments 2 :

Pour scinder un groupe A_i en deux sous groupes :

prendre chaque paire d'éléments i_1 et i_2 de A_i

- Affecter les éléments de A_i à deux A_{i1} et A_{i2} par la règle des plus proches voisins
- Calculer la pseudo distance des moments pour ces deux groupes

$$psd(A_{i1}, A_{i2}) = \frac{N_{i1} N_{i2}}{N_{i1} + N_{i2}} d^2(g_{i1}, g_{i2})$$

choisir la paire (i_1, i_2) qui maximise cette pseudo distance.