

Clustering

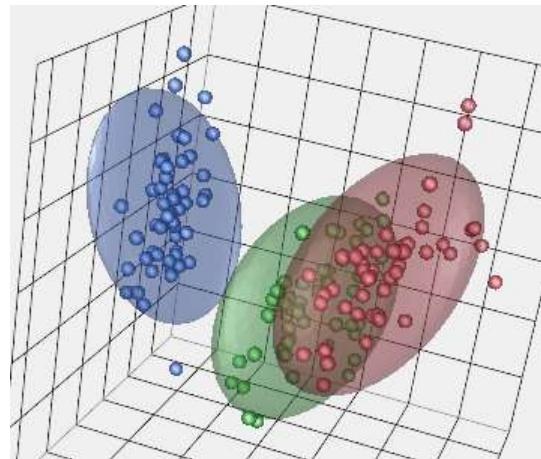
Le clustering est la technique non supervisée la plus répandue en datamining.

Elle permet de distinguer des groupes homogènes (classes, segments, clusters) au sein d'un grand volume de données.

- De part leur constitution, ces groupes peuvent apporter une information pertinente sur les données, notamment s'ils sont représentés graphiquement à l'aide d'une ACP.
- Ils peuvent aussi servir à découper une étude en sous-parties, chacune pouvant bénéficier de traitements particuliers.

Méthodes de clustering :

- Généralités
- K-means
- Classification hiérarchique ascendante



Généralités

L'objectif des méthodes est

- à la fois, de regrouper les observations ayant des caractéristiques similaires au sein d'une même classe,
 - distance entre observations
- à la fois de construire des classes les plus dissemblables possibles.
 - distance entre classes

Recherche exhaustive impossible

Notons que le nombres de partitions distinctes de n objets est

$$\frac{1}{e} \sum_{k \geq 1} \frac{k^n}{k!}$$

Par exemple pour 30 objets, on a plus 10^{23} partitions possibles.

⇒ *Algorithme de recherche performant*

Les métriques sur les observations : variables quantitatives

Pour trouver des similarités entre les observations il faut définir une métrique sur les observations :

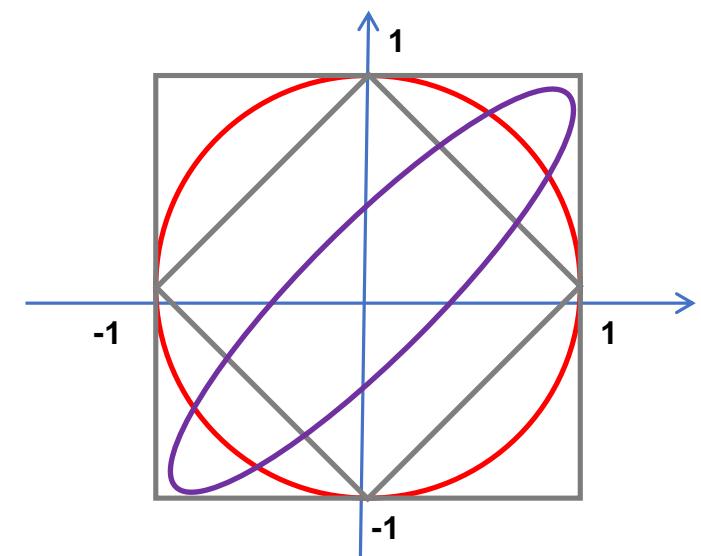
Distance euclidienne : $d_2(x, y) = \left(\sum_{i=1}^d (x_i - y_i)^2 \right)^{1/2}$

Distance Manhattan : $d_1(x, y) = \sum_{i=1}^d |x_i - y_i|$

Atténue l'impact des individus hors norme car pas d'écart au carré

Distance infinie : $d_\infty(x, y) = \max \{|x_1 - y_1|, \dots, |x_d - y_d|\}$

Distance de Mahalanobis : $d(x, y) = \left((x - y)^T \Sigma^{-1} (x - y) \right)^{1/2}$
 Σ = matrice carrée définie positive (permet d'introduire une corrélation entre les variables)



Les métriques sur les classes

Pour construire des classes dissemblables il faut définir une métrique sur les classes, c'est-à-dire une distance entre deux ensembles de points

- *Distance minimale* entre deux observations des deux classes :

$$d_{\min}(C_1, C_2) = \min_{x \in C_1, y \in C_2} d(x, y)$$

déetecte les formes allongées voire sinuuses, sensible à l'effet de la chaîne (2 points éloignés sont considérés comme appartenant à la même classe car reliés par une série de points proches les uns des autres)

- *Distance maximale* entre deux observations des deux classes :

$$d_{\max}(C_1, C_2) = \max_{x \in C_1, y \in C_2} d(x, y)$$

très sensible aux observations atypiques

- *Distance moyenne* entre deux observations des deux classes :

$$d_{\text{moy}}(C_1, C_2) = \text{moyenne}_{x \in C_1, y \in C_2} d(x, y)$$

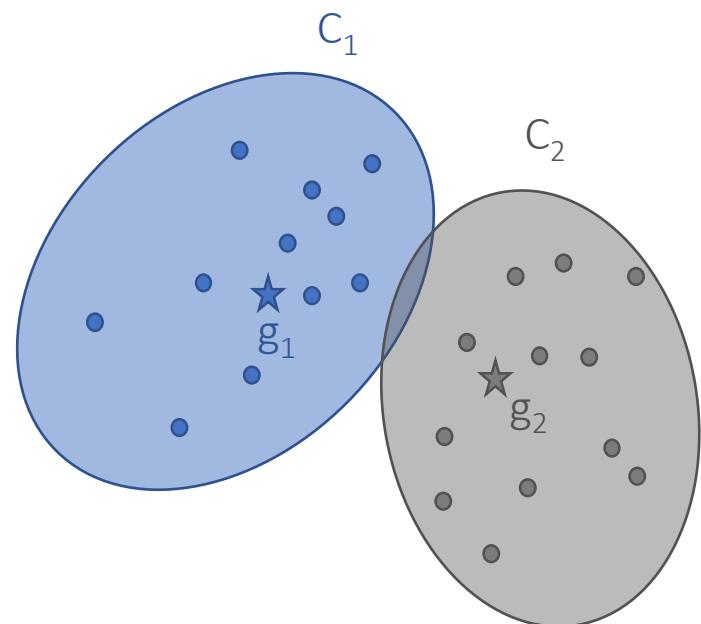
moins sensible au bruit, tend à produire des classes de même variance.

- *Distance de Ward* :

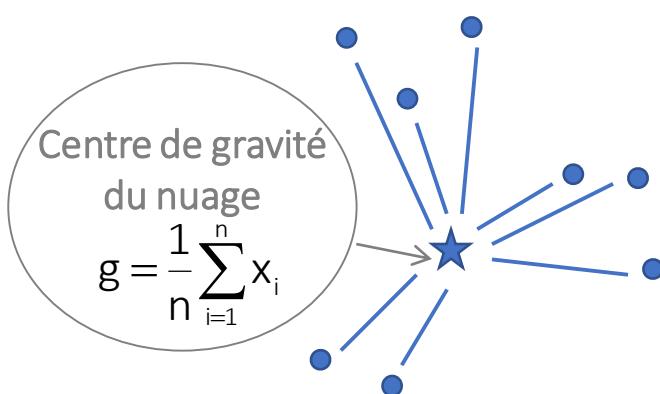
$$d_{\text{Ward}}(C_1, C_2) = \frac{n_1 \times n_2}{n_1 + n_2} d(g_1, g_2)$$

où n_i et g_i sont l'effectif et le centre de gravité de la classe C_i

la plus utilisée, permet de fusionner les deux classes faisant le moins baisser l'inertie inter-classes, tend à produire des classes sphériques de même effectif .



Inertie inter et intra classes



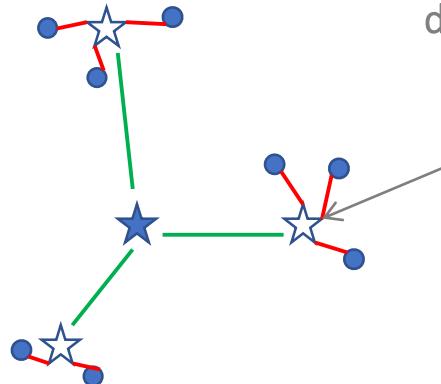
Inertie Totale

d=distance euclidienne

$$\frac{1}{n} \sum_{i=1}^n d^2(x_i, g)$$

$\underbrace{\quad}_{I_{tot}}$

Ça ne vous rappelle rien?



Centre de gravité d'une classe C_k

$$g_k = \frac{1}{n_k} \sum_{x_i \in C_k} x_i$$

Inertie intra classes + Inertie inter classes

$$\underbrace{\frac{1}{n} \sum_{k=1}^p \sum_{i=1}^{n_k} d^2(x_i - g_k)}_{I_{intra}} + \underbrace{\frac{1}{n} \sum_{k=1}^p n_k d^2(g_k, g)}_{I_{inter}}$$

- Chercher la partition qui minimise l'inertie intra classes (*homogénéité des observations dans les classes*)
- Chercher la partition qui maximise l'inertie inter classes (*dissimilarité des classes entre elles*)

Le coefficient

$$R^2 = \frac{I_{inter}}{I_{tot}}$$

est le pourcentage d'inertie du nuage expliquée par les classes. L'objectif est d'obtenir un R^2 proche de 1 avec un minimum de classes (si nb classes = n alors $R^2=1$)

Il peut servir pour

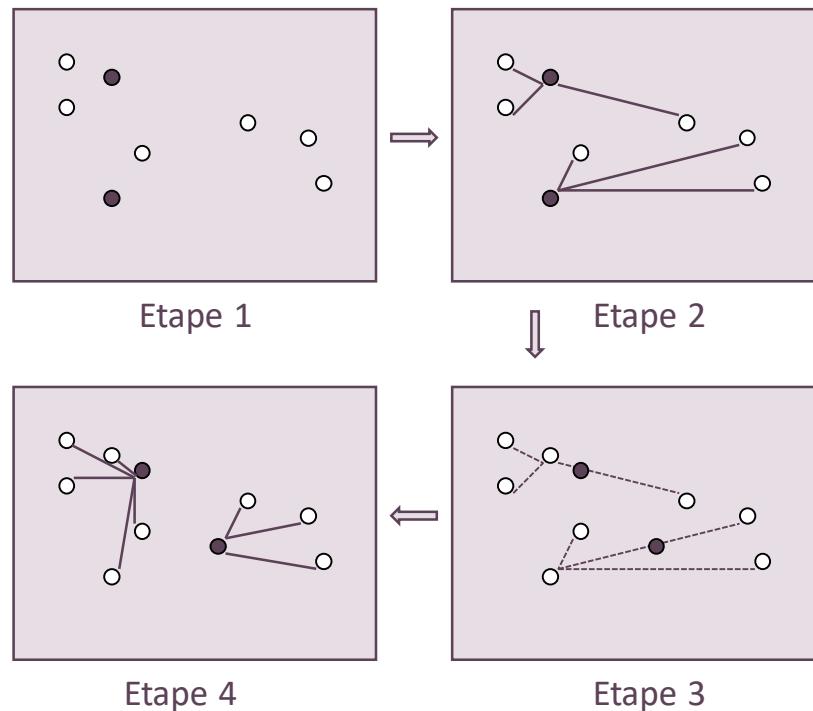
- Comparer deux partitionnements ayant le même nombre de classes
- Sélectionner le nombre de classes (courbe R^2 vs nb classes. on choisit le dernier saut important du R^2)

Algorithme des k-means (1/2)

Soit C le nombre de classes souhaitées.

Algorithme

- Etape 1 : Choisir C individus au hasard comme centres initiaux des classes
- Etape 2 : On calcule les distances entre chaque individu et chaque centre de classe, et on affecte l'individu à la classe la plus proche
- Etape 3 : On remplace les centres des classes par les C barycentres des classes définies à l'étape 2
- Etape 4 : On itère à partir de l'étape 2 jusqu'à convergence



Basé sur la
distance entre
individus

Algorithme des k-means (2/2)

Caractéristiques

- Dépend de l'initialisation des centres \Rightarrow répéter plusieurs fois l'algorithme
- Nombre C de classes fixé à l'avance \Rightarrow tester plusieurs valeurs de C
- Un individu atypique est détecté car il forme une classe à lui tout seul (en général)
- Complexité linéaire \Rightarrow adapté à de grands volumes de données
(attention toutefois car il faut tester plusieurs nombres de classes et répéter l'algorithme plusieurs fois pour chaque classe)

Classification hiérarchique ascendante (1/2)

Algorithme

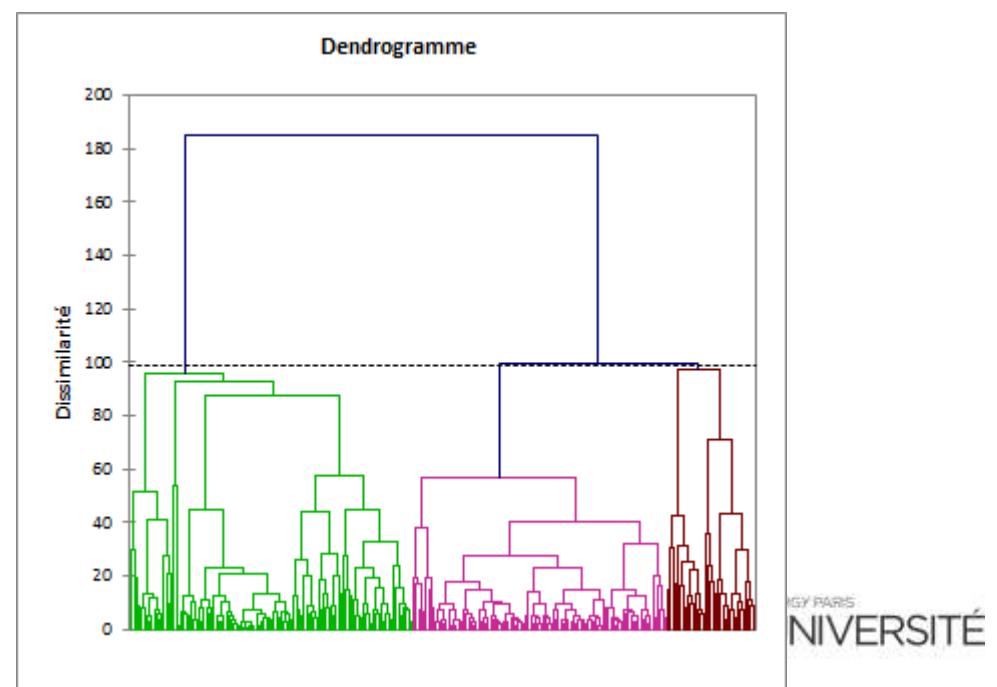
- Etape 1 : Chaque individu forme une classe (n classes)
- Etape 2 : On calcule les distances entre les classes et on regroupe les deux classes les plus proches (C classes \rightarrow C-1 classes)
- Etape 3 : On itère à partir de l'étape 2 jusqu'à n'avoir qu'une seule classe
- Etape 4 : Choix du partitionnement à partir de dendrogramme

Basé sur la distance entre classes

Le **dendrogramme** représente la suite de partitions obtenues au cours de l'algorithme. L'axe des ordonnées représente une mesure de dissimilarité/inertie inter-classes (R^2 partiel,...).

On coupe le dendrogramme où la hauteur des branches est élevée. Cela permet d'obtenir simultanément :

- Le nombre de classes
- La constitution des classes



Classification hiérarchique ascendante (2/2)

Caractéristiques

- Regroupe des individus ou des variables dès qu'il y a une notion de distance
- Pas de dépendance à l'initialisation
- Nombre de classes non fixé à l'avance
- formes diverses des groupes grâce au choix de la distance
- A chaque étape le partitionnement dépend de celui obtenu avant \Rightarrow Optimum local
- Complexité exponentielle de l'algorithme

Conclusion

Méthodes non hiérarchiques

- ✓ Il faut avoir une idée a priori du nombre de classes
- ✓ L'initialisation de l'algorithme peut avoir un impact sur la partition finale
- ✓ L'algorithme converge assez vite (complexité linéaire)

Algorithme hiérarchique

- ✓ La complexité de l'algorithme est exponentielle
- ✓ On n'a pas besoin de connaître à l'avance le nombre de classes

Quand cela est possible confirmer les résultats par plusieurs méthodes

Avez-vous des questions?

Documents ayant servi à la rédaction des slides et TD :

- *DataMining et Statistiques décisionnelles*, Stéphane Tufféry, Ed. Technip
- <https://penseeartificielle.fr/choisir-distance-machine-learning/>
- http://eric.univ-lyon2.fr/~ricco/cours/slides/classif_centres_mobiles.pdf
- <https://scikit-learn.org/stable/index.html>