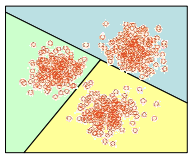
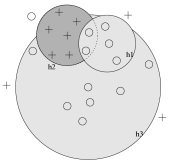
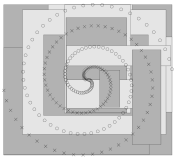
Clustering

Méthodes et algorithmes avancés
Mars - 2006


Clustering non supervisé


Sous contraintes


Un challenge : la spirale

Clustering (catégorisation)

- Objectifs du clustering
- Mesures de distances habituelles, similarités entre objets
- Les différents algorithmes
  - Clustering *dur* (K-means, ...)
  - Clustering *doux* (Fuzzy C-means, ...)
  - Clustering hiérarchique
- Exemples d'applications

Objectifs

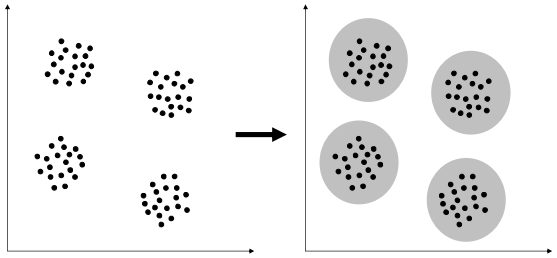
- Apprentissage non supervisé
- But : organisation d'objets en groupes (= classes)
- Nécessité d'une mesure de similarité

Clustering (catégorisation) ≠ Classification

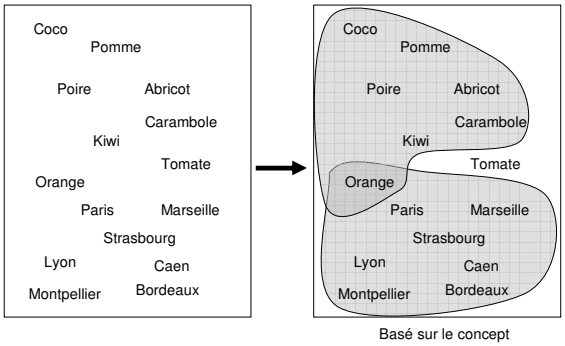
- Non supervisé
- Uniquement à partir d'une mesure de similarité

- Supervisé
- Classer *à partir de classes*

Exemple


Basé sur la distance

Exemple



Basé sur le concept

But de la catégorisation

- Grouper les données selon un critère d'homogénité
- Critère complexe à déteminer, peut dépendre :
  - des données
  - de l'utilisation que l'on souhaite faire des clusters
  - de la subjectivité de l'utilisateur

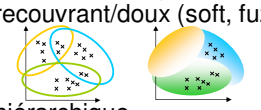
Un bon clustering ?

- Produire des catégories de grande qualité, càd :
  - la similarité intra-classe est grande
  - la similarité inter-classe est faible
- La qualité du résultat dépend :
  - de la mesure de similarité et de son implémentation
  - de la définition et la représentation d'un cluster
- La méthode peut être évaluée par sa capacité à découvrir des motifs cachés

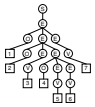
Plusieurs types

- Clustering exclusif, clustering dur (hard)

- Clustering recouvrant/doux (soft, fuzzy)

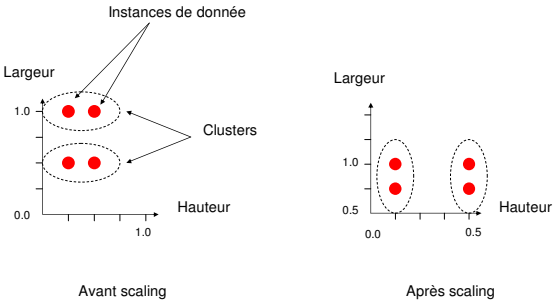


- Clustering hiérarchique



- Clustering probabilliste

Mesure de distance



Avant scaling

Après scaling

Mesures

Notations :

- { o_0, ..., o_N } : N échantillons de données
- K : nombre de classes
- g_k : centre de gravité de la classe C_k
- σ_k : variance de la classe C_k
- σ : variance du jeu complet

Mesure de Minkowski

$$d_p(o_i, o_j) = \left(\sum_{k=1}^{d} \left| o_{i,k} - o_{j,k} \right|^p\right)^{\frac{1}{p}}$$

p=1 Distance de Manhattan

p=2 Distance euclidienne

Utilisée pour : les données à grande dimensionnalité (d >> 3)

Mesure de distance

Critère	Formule	Description
Inertie intra-classe	$I_t = \sum_{k=1}^{K} \sum_{i \in C_k} \text{dist}(o_i, g_k)^2 = \sum_{k=1}^{K} \sigma_k$	Variance entre les échantillons et le centre de gravité de leurs classes
Compacité	$Cmp = \frac{1}{K} \cdot \sum_{k=1}^{K} \frac{\sigma_k}{\sigma}$	Degré de regroupement
Critère Xie-Beni	$XB = \frac{I_t}{N \cdot \min_{i,j \in K, i \neq j}(\text{dist}(g_j, g_i))}$	Mesure de la séparation des classes, indépendant de l'échelle des valeurs
Critère Wemmert-Gançarski	$WG = \frac{I_t}{N \cdot \min_{o_i \in C_k, k' \neq k}(\text{dist}(o_i, g_{k'}))}$	Séparation + compacité des objets des classes, indépendant de l'échelle

Hard-Clustering : K-Means

K-Means (MacQueen, 1967)

But : minimiser ***J***

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \underbrace{\left\| x_i^{(j)} - c_j \right\|^2}$$

mesure de distance choisie entre une donnée $x_i^{(j)}$ et le centre du cluster c_j

Indicateur de la distance des n points à partir des centres de leurs clusters respectifs

K-Means

L'algorithme :

K-MEANS (K)

- Placer aléatoirement K points (centroides) dans l'espace des objets à catégoriser. Ces points représentent les centres de chaque cluster

Faire

- Assigner à chaque objet le cluster dont le centroide lui est le plus proche
- Quand tous les objets ont été assignés, recalculer les positions des K centroides, en prenant le barycentre des clusters correspondants

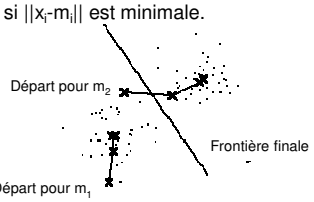
Tant que la position des centroides n'est pas stabilisée

K-Means

Justification :

- n exemples de données : [X_1, ..., X_n]
- k clusters, avec k<n
- m_i la moyenne des exemples du cluster i

X_i est dans C_i si $||x_i - m_i||$ est minimale.



Soft-Clustering : Fuzzy C-Means

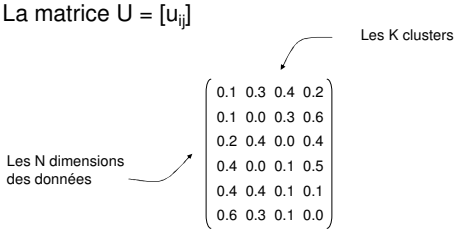
Fuzzy C-Means (FCM) [Dunn, 1973 ; Bezdek, 1981]

- Autorise une donnée à appartenir à deux clusters ou plus
- Utilisé très fréquemment en reconnaissance de motifs
- But : minimiser la fonction objectif J_m

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2, \quad 1 \le m < \infty$$

Soft-Clustering : Fuzzy C-Means

La matrice U = [u_{ij}]



Soft-Clustering : Fuzzy C-Means

FUZZY-C-MEANS (K, m)

- Initialiser aléatoirement une matrice U=[u_{ij}] $U^{(0)} = U$
- A l'étape k, faire
  - Calculer les centroides $C^{(k)}=[c_j]$ en utilisant $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

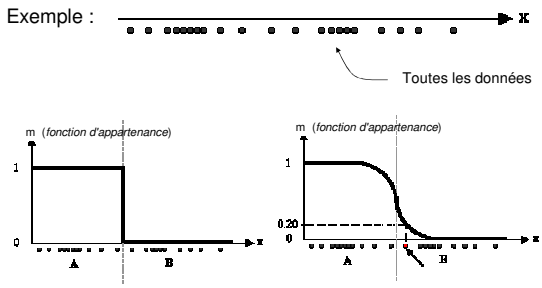
  - Mettre à jour $U^{(k)}$ qui devient $U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}}$$

Tant que $|| U^{(k+1)} - U^{(k)} || > \varepsilon$

Comparaison avec K-Means

Exemple :



K = 2

Catégorisation hiérarchique

Idée de base (S.C. Johnson,1967)

Soit :
- N exemples,
- une matrice de distance ou de similarité N*N

Processus :

- Commencer par assigner chaque donnée dans son propre cluster. On définit la distance entre les clusters identique à celle entre les données qu'ils contiennent.

Faire
- Trouver la paire de cluster la plus proche, et la fusionner dans un seul cluster.
- Calculer la distance *d* entre les clusters restants (le nouveau et tous les anciens)

Tant que tous les exemples ne sont pas dans un seul cluster de taille N

Sortie :

Catégorisation hiérarchique

Techniques pour calculer la distance *d(i,j)* :

single-linkage (ou minimum) :
d vaut la plus courte distance entre n'importe quel élément du cluster i et n'importe quel élément du cluster j.

complete-linkage (ou diamètre) :
d vaut la plus grande distance entre n'importe quel élément du cluster i et n'importe quel élément du cluster j.

average-linkage :
d vaut la distance moyenne entre les éléments des deux clusters.

Méthode UCLUS [D'Andrade 1978] :
utilisation du médian plutôt que de la moyenne.

[0.0 0.1 0.1 0.1 0.3 0.4 2.5]

median — moyenne = 0.5

Conclusion : la méthode UCLUS est plus robuste pour les jeux de données contenant des exemples extrêmes (*outliers*).

Catégorisation hiérarchique

Catégorisation **agglomérative** (comme celle de Johnson) :
fusionner les clusters itérativement

≠

Catégorisation **divisive** (beaucoup plus rare) :
partir d'un cluster, et le diviser

Catégorisation hiérarchique

Algorithme de Johnson

- Matrice de proximité D = [d(i,j)] pour deux objets i et j.
- d(i,i) = 0
- d(i,j) = d(j,i)
- D est de taille N*N
- Séquence de n clusters : (0), (1), ..., (n-1)
  i, numéro du cluster
  (i), le cluster lui-meme
- L(k) : niveau de la k-ième catégorisation (= représentation de la distance qui sépare ses constituants sous la forme d'un plateau dans l'arbre)
- Proximité entre les clusters (a) et (b) : d[(a),(b)]

Catégorisation hiérarchique

Algorithme de Johnson

HIERARCHICAL-CLUSTERING (D)

Commencer en plaçant chaque objet dans un cluster. L(c) = 0 pour tout cluster c. m = 0

Faire

- Trouver les paires de clusters (a) et (b) les moins dissimilaires. d[(a),(b)] = min d[(i)(j)] pour tout cluster i, j.

- Incrémenter le compteur m = m+1. Fusionner les clusters (a) et (b) en un seul cluster (m). Le niveau de la catégorisation est L(m) = d[(a),(b)].

- Mettre à jour la matrice de proximité D. Enlever les lignes et les colonnes qui correspondent aux clusters (a) et (b) et les remplacer par une ligne et une colonne qui correspondent au nouveau cluster (m). Calculer les nouvelles valeurs de D :

d[(k), (m)] = min { d[(k),(a)], d[(k),(b)] } pour tous les clusters (k) différents de (a) ou (b)

Tant que le nombre de clusters est plus grand que 1

Catégorisation hiérarchique

Problèmes avec la catégorisation agglomérative :

- Peu efficace avec un grand nombre de données. Complexité en temps : $O(n^2)$
- Ne remet pas en cause les regroupements de l'étape précédente

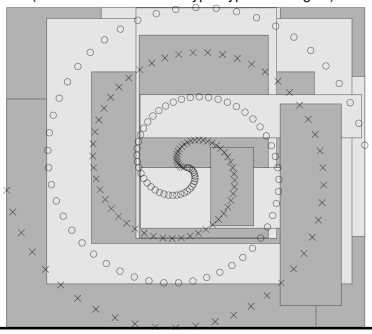
Applications possibles

- Marketing : trouver des groupes de consommateurs similaires (mêmes produits consommés, mêmes comportements) dans une grande base de données
- Biologie : classification d'animaux, de plantes, ...
- Librairie : classer des ouvrages
- Assurance, domaine bancaire : identifier des logements, des marchés, des cours boursiers, ... avec la meilleure plu-value ; identifier les fraudeurs
- Plannification urbaine : identifier des parcelles en fonction de leur valeurs, leur type et leur localisation géographique
- Etudes terrestres : identifier les zones dangeureuses (raz-de-marées, tsunami, tremblements de terre, ...)
- Web : classification automatique de documents par mots-clés (textes, images, sons, vidéos, ...)

Exemples

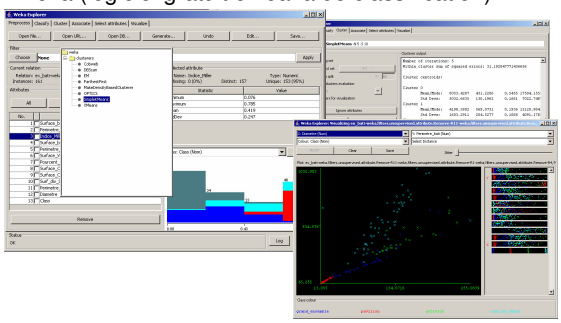
- Thèse de Jean-Pierre Novak (2000)

(réseaux neuronaux de type hyper-rectangles)



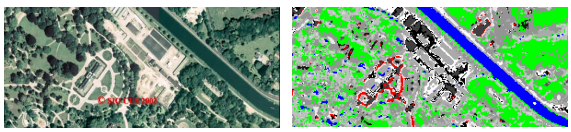
Exemples

- Weka (logiciel gratuit en Java de classification)

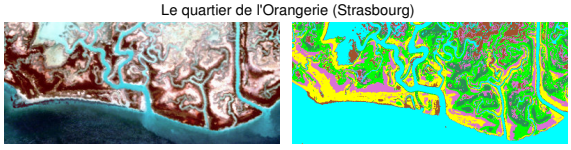


Exemples

- K-Means sur des images réelles



Le quartier de l'Orangerie (Strasbourg)



Les marais de San Felice (Venise, IT)