UNIVERSITÉ CÔTE D'AZUR

Introduction à l'Intelligence Artificielle (L2 Portail Sciences et Technologies)

Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
andrea.tettamanzi@univ-cotedazur.fr

univ-cotedazur.fr

Séance 7
Apprentissage non supervisé (clustering)

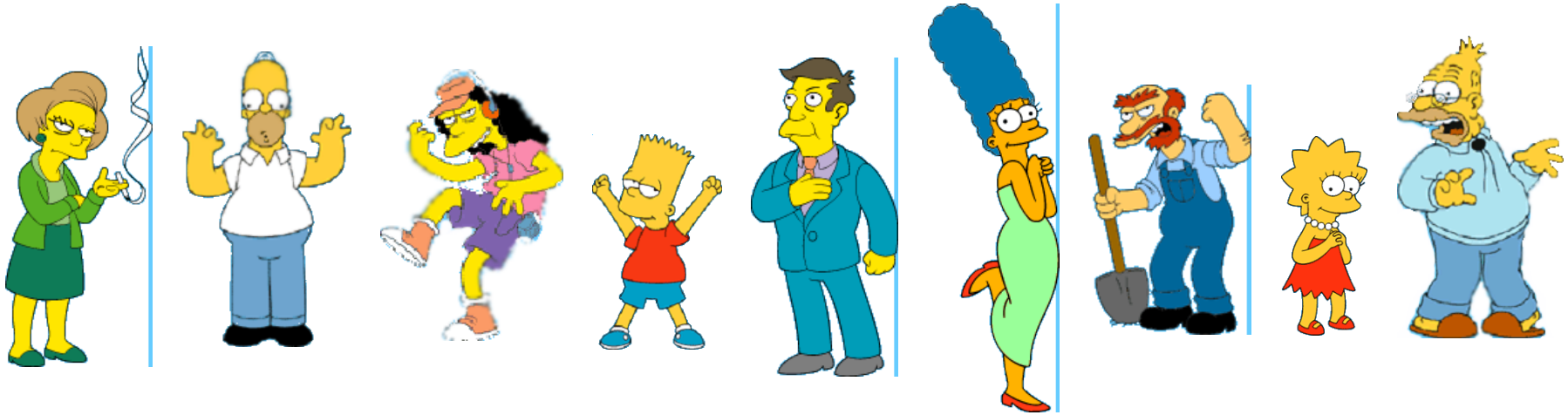
Plan pour cette séance

- Introduction et notions de base
- Survol des méthodes plus utilisées
  - Algorithmes de partitionnement
  - Méthodes hiérarchiques
  - Méthodes basés sur la densité
  - Méthodes basés sur les modèles

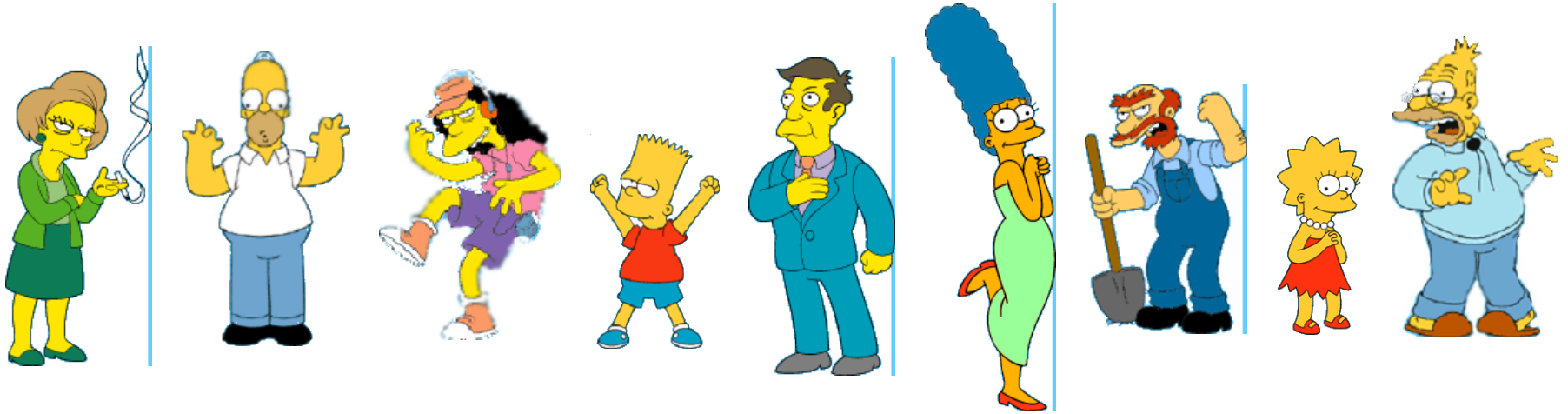
Clustering (Regroupement)

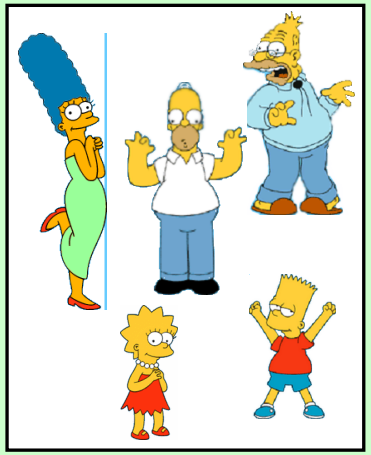
- Cluster: une collection d'objets (enregistrements du jeu de données)
  - Similaires entre eux au sein d'un même cluster
  - Dissimilaires des objets des autres clusters
  - Besoin d'une manière de calculer la similarité/distance entre objets
- Analyse de clusters
  - Trouver les similarités entre les données suivant leurs caractéristiques et regrouper les objets les plus similaires dans des clusters
- **Apprentissage non-supervisé** : pas de classes prédéfinies
- Applications typiques
  - Comme **outil en soi** pour mieux comprendre les données
  - Comme **étape de pré-traitement** pour d'autres algorithmes

Quel regroupement « naturel » pour ces objets?



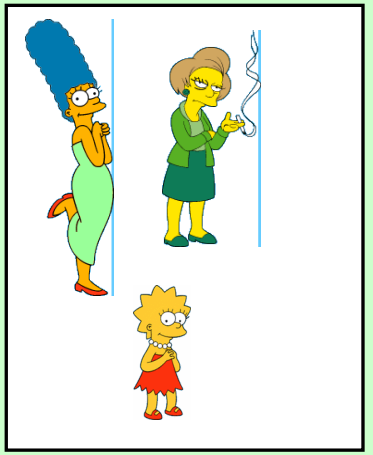
Le clustering est subjectif !





Simpson Family

School Employees



Females

Males

Qualité : qu’est-ce qui fait un bon regroupement ?

- Une méthode de clustering doit viser à produire des clusters de bonne qualité avec
  - Similarité intra-classe importante
  - Similarité inter-classe minimale
- La qualité d’un résultat de clustering dépend à la fois de la mesure de similarité et de la méthode qui l’utilise
- La qualité d’une méthode est aussi mesurée par sa capacité de découvrir des (ou tous) les patrons cachés dans les données

Qu’est-ce que la similarité ?

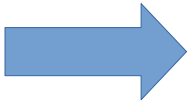
Mesure de qualité d’un regroupement

- **Métrique de (dis) similarité** : la similarité est exprimée en termes d’une fonction de distance, typiquement métrique : $d(i, j)$
- On définit une fonction de « qualité » séparée, qui mesure la « bonté » d’un cluster.
- Les définitions des **fonctions de distance** sont très différentes selon le type des variables.
- Les variables peuvent être pondérées suivant l’application et la sémantique des données.
- Il est difficile de définir « assez similaire » ou « assez bon »
  - La réponse est typiquement très subjective.

Structures de données

Matrice de données

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{im} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$



Matrice de dissimilarité (= distance)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & d(n,3) & \cdots & 0 \end{bmatrix}$$

Types de données

- Variables sur un intervalle
- Variables binaires
- Variables catégorielles
- Variables ordinales et proportionnelles
- Variables de type mixte

Variables sur un intervalle

- Normalisation des données
  - On calcule l'écart absolu moyen :

$$s_j = \frac{1}{n}\sum_{i=1}^{n}|x_{ij} - \mu_j| \qquad \text{où} \qquad \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

  - On calcule les mesures normalisées (*z-score*)

$$z_{ij} = \frac{x_{ij} - \mu_j}{s_j}$$

- L'écart absolu moyen est plus robuste que l'écart type

Similarité et dissimilarité

- Les distances sont utilisées normalement pour mesurer la similarité ou dissimilarité entre deux objets
- Parmi les plus utilisées : *distance de Minkowski* :

$$d(i,j) = \sqrt[q]{\sum_{k=1}^{m} |x_{ik} - x_{jk}|^q}$$

  - où $i = (x_{i1}, x_{i2}, \ldots, x_{im})$ et $j = (x_{j1}, x_{j2}, \ldots, x_{jm})$ sont deux objets m-dimensionnels (= enregistrements), et q est un entier positif
- Pour $q = 1$, d est la distance dite *de Manhattan*

$$d(i,j) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|$$

Similarité et dissimilarité

- *Pour q = 2*, *d* est la distance euclidienne

$$d(i,j) = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2}$$

- Propriétés des distances :
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
  - $d(i,j) \leq d(i,k) + d(k,j)$
- En outre, on peut utiliser une distance pondérée, une corrélation des moments produit de Pearson ou d'autres mesures de dissimilarité

Variables binaires

- Table de contingence pour données binaires (*m* variables/colonnes)

Obj i \ Obj j	1	0	sum
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	m

Mesure de distance pour les variables binaires symétriques: $d(i,j) = \frac{b+c}{a+b+c+d} = \frac{b+c}{m}$

- Mesure pour les variables binaires asymmétriques: $d(i,j) = \frac{b+c}{a+b+c}$
- Coéfficient de Jaccard (mesure de *similarité* pour les variables *asymmétriques*): $sim_{\text{Jaccard}}(i,j) = \frac{a}{a+b+c}$

Variables binaires

- Exemple

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

  - « gender » est une variable symmétrique
  - Les autres variables sont binaires asymmétriques
  - Soient les valeurs Y et P traitées comme 1, et N comme 0

$$
\begin{aligned}
d(\text{Jack}, \text{Mary}) &= \frac{0+1}{2+0+1} = 0.33 \\
d(\text{Jack}, \text{Jim}) &= \frac{1+1}{1+1+1} = 0.67 \\
d(\text{Jim}, \text{Mary}) &= \frac{1+2}{1+1+2} = 0.75
\end{aligned}
$$

Variables catégorielles

- Une généralisation des variables binaires, car elles peuvent prendre plus de 2 valeurs, p.ex., rouge, jaune, bleu, vert
- Méthode 1 : correspondance simple
  - *h*: nb des correspondances, *m*: nb total des variables

$$d(i,j) = \frac{m-h}{m}$$

- Méthode 2: utiliser un grand nombre de variables binaires
  - créer une nouvelle variable binaire pour chacune des *M* valeurs catégorielles

Variables ordinales

- Une variable ordinale peut être discrète ou continue
- L'ordre est significatif, p.ex., classement
- Peuvent être traitées comme des variables sur un intervalle
  - On remplace x_{ik} par son rang $r_{ik} \in \{1, \ldots, M_k\}$
  - On reporte le domaine de chaque variable sur [0, 1] en remplaçant l' *i*-ème valeur de la *k*-ème variable par

$$z_{ik} = \frac{r_{ik} - 1}{M_k - 1}$$

  - On calcule la dissimilarité en utilisant les méthodes pour les variables sur un intervalle

Variables proportionnelles

- Toute mesure positive sur une échelle nonlinéaire, plus ou moins exponentielle, comme Ae^{Bt} ou Ae^{-Bt}
- Méthodes :
  - On peut les traiter comme des variables sur un intervalle—*mauvaise idée !* (pourquoi ?—l'échelle sera biaisée)
  - Y appliquer une transformation logarithmique

$$y_{ik} = log(x_{ik})$$

  - Les traiter comme des données ordinale continue
  - Traiter leur rang comme un variable sur un intervalle

Variables de type mixte

- Un jeu de donnée peut contenir tous les six types de variables
- On peut utiliser une formule pondérée pour les combiner

$$d(i,j) = \frac{\sum_{k=1}^{m} \delta_{ij}^{(k)} d_{ij}^{(k)}}{\sum_{k=1}^{m} \delta_{ij}^{(k)}}$$

  - Si la variable *k* est binaire ou catégorielle :
    $d_{ij}^{(k)} = 0$ si $x_{ik} = x_{jk}$, $d_{ij}^{(k)} = 1$ sinon
  - Si la variable *k* est sur intervalle : distance normalisée
  - Si la variable *k* est ordinale ou proportionnelle
    - On calcule les rangs r_{ik} et
    - On traite z_{ik} comme intervalle

$$z_{ik} = \frac{r_{ik} - 1}{M_k - 1}$$

Objets vectoriels

- Exemples : mots clés dans des documents, gènes dans un micro-array, etc.
- Applications : recherche d'information, taxinomie biologique, etc.
- Distance cosine

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \cdot \mathbf{y}}{|\mathbf{x}| \cdot |\mathbf{y}|}$$

- Variante : coefficient de Tanimoto

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \cdot \mathbf{y}}{\mathbf{x}^T \cdot \mathbf{x} + \mathbf{y}^T \cdot \mathbf{y} - \mathbf{x}^T \cdot \mathbf{y}}$$

Méthodes de Clustering

- Partitionnement (construction itérative de partitions)
  - K-Means, k-Medoids, etc.
- Hiérarchiques (construction d'un dendrogramme d'instances)
  - Diana, Agnes, BIRCH, ROCK, CAMELEON
- Basées sur la densité (basées sur connectivité et fonction de densité)
  - DBSCAN, OPTICS, DenClue
- Basées sur un grille
  - STING, WaveCluster, CLIQUE
- Basées sur un modèle
  - expectation maximization
  - Self-organizing maps
- Basées sur les patron fréquents

Alternatives pour le calcul de la distance entre clusters

- Single linkage : la plus petite distance entre un élément d'un cluster et celui d'un autre, $d(K_p, K_q) = \min d(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(q)})$
- Complete linkage: la plus grande distance entre un élément d'un cluster et celui d'un autre, $d(K_p, K_q) = \max d(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(q)})$
- Average linkage: distance moyenne entre un élément d'un cluster et celui d'un autre, $d(K_p, K_q) = \text{avg } d(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(q)})$
- Centroïde: distance entre les centroïdes de deux clusters, $d(K_p, K_q) = d(C_p, C_q)$
- Médoïde : distance entre les médoïdes, $d(K_p, K_q) = d(M_p, M_q)$
  - Médoïde : un objet sélectionné, situé vers le centre du cluster

Centroïde, rayon et diamètre d'un cluster (données numériques)

- Centroïde : point du milieu d'un cluster $C_p = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{x}_i^{(p)}$
- Rayon : distance moyenne du centroïde des éléments du cluster

$$R_p = \sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} \left(\mathbf{x}_i^{(p)} - C_p \right)^2}$$

- Diamètre : distance moyenne entre chaque paire d'éléments du cluster

$$D_p = \sqrt{\frac{1}{N_p(N_p - 1)} \sum_{i=1}^{N_p} \sum_{j \neq i} \left(\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p)} \right)^2}$$

Algorithmes de partitionnement : notions de base

- Méthode de partitionnement : on construit une partition du jeu de données *D* de *n* objets en un ensemble de *k* clusters, en minimisant

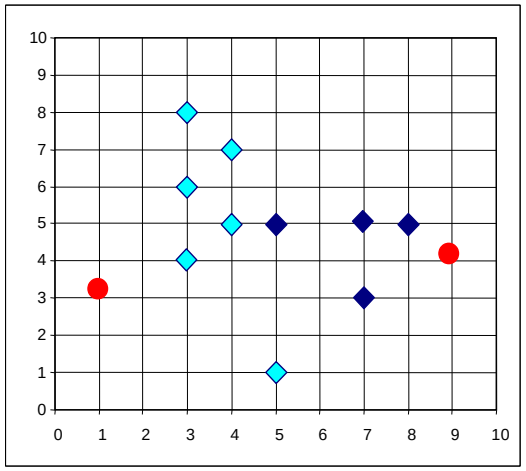
$$\sum_{p=1}^{k} \sum_{i=1}^{N_p} \left(\mathbf{x}_i^{(p)} - C_m \right)^2$$

- Étant donné *k*, trouver la partition de *k clusters* qui optimise le critère de partitionnement choisi
  - Optimum global : il faut énumérer toutes les partitions possibles !
  - Méthode heuristique : algorthmes *k-means* et *k-medoids*
    - *k-means* (MacQueen ’67) : chaque cluster est représenté par son centroïde
    - *k-medoids* ou PAM (Partition around medoids) (Kaufman & Rousseeuw ’87) : chaque cluster est représenté par un de ses éléments

K-Means

- Étant donné *k*, l'algorithmne *k-means* se déroule en quatre étapes:
  1) On partitionne les objets en *k* sousensembles non vides
  2) On prend comme points initiaux les centroïdes des clusters de la partition courante
  3) On affecte chaque objet au cluster associé au centroïde le plus proche
  4) On revient à l'étape 2 et on s'arrête dès que les clusters ne changent plus

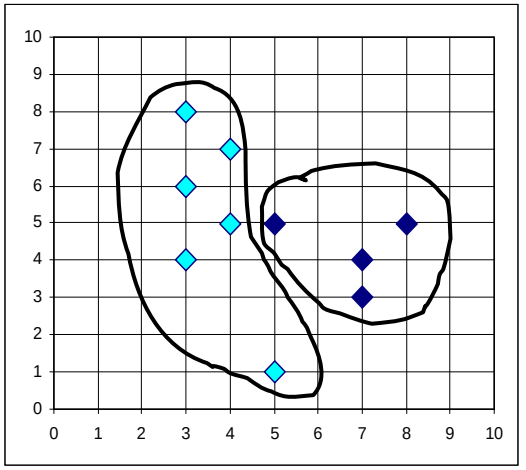
Exemple de K-Means



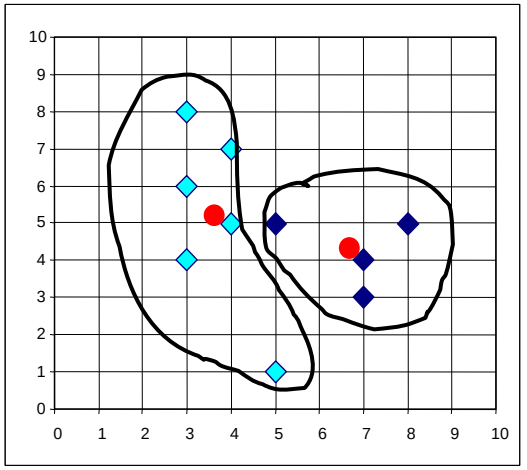
k=2

On choisit arbitrairement k objets comme centroïdes initiaux

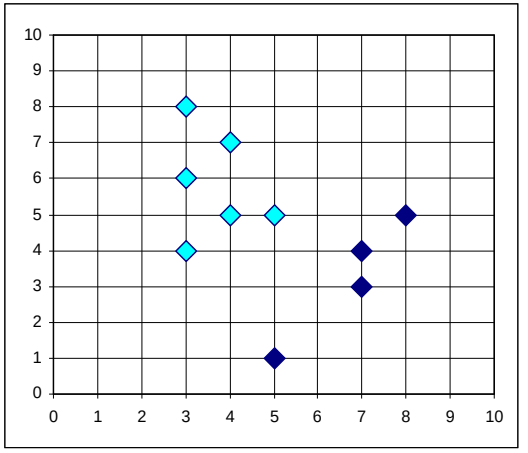
On affecte chaque objet au plus proche centre



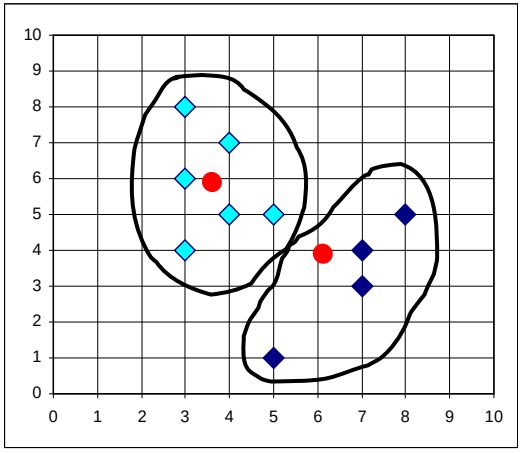
Mise à jour des centroïdes



réaffectation



Mise à jour des centroïdes



réaffectation

Quelques commentaires sur *K-Means*

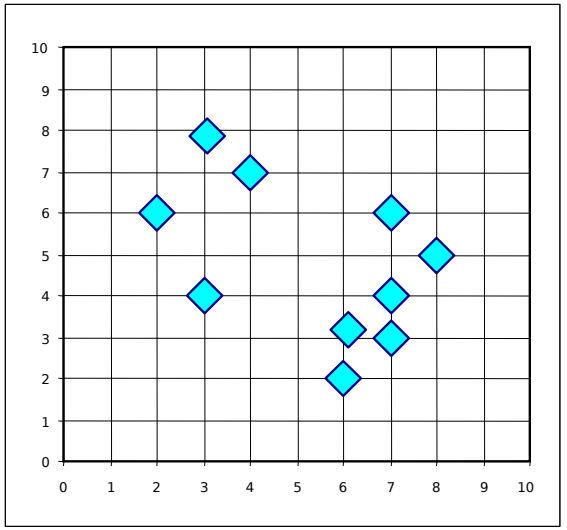
- Avantages: *Relativement efficace : O*(*tkn*), où *n* = nb. objects, *k* = nb. de clusters, et *t* = nb. itérations. Normalement, *k*, *t* << *n*.
    - Pour comparaison : PAM: $O(k(n – k)^2)$, CLARA: $O(ks^2 + k(n – k))$
- Remarque: On termine souvent dans un *optimum local*. Si on veut l'*optimum global,* il faut avoir recours à des méthodes d'optimization comme : *recuit simulé* ou *algorithmes évolutionnaires*
- Inconvénients
    - Applicable seulement quand la *moyenne* est définie, donc quid des données catégorielles ?
    - Il faut spécifier *k,* le *nombre* de clusters, à l'avance
    - Ne sait pas gérer des données bruitées et des *outliers*
    - Pas adapté pour découvrir des clusters avec des *formes non-convexes*

K-Médoïdes

- On trouve des objets *representatifs*, appelés médoïdes, dans les clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - On commence par un ensemble initial de médoïdes et on replace itérativement un des médoïdes par un des non-médoïdes s'il améliore la distance totale du regroupement résultant
  - *PAM* est efficace pour des jeux de données petits, mais il ne passe pas à l'échelle pour des jeux de données grands
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): échantillonnage aléatoire
- Focus + structure de données spatiale (Ester et al., 1995)

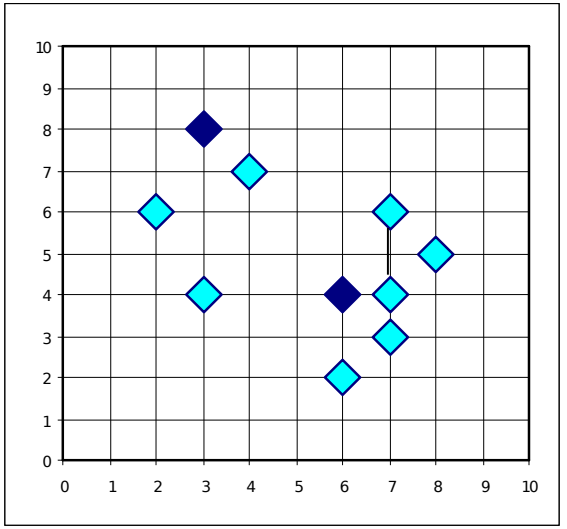
Fonctionnement d'un algorithme de *k*-Médoïdes typique (PAM)

Total Cost = 20

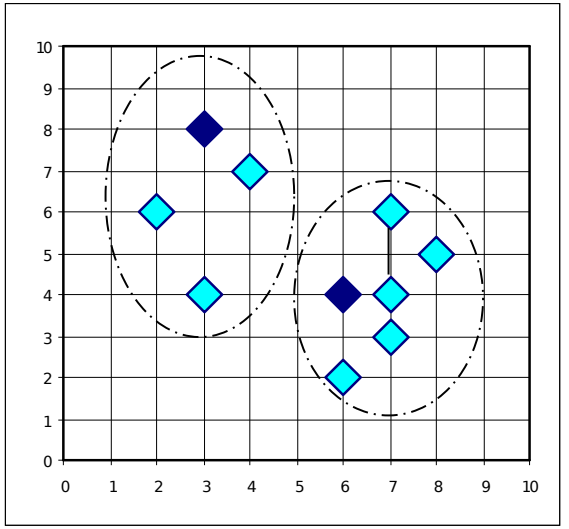


K=2

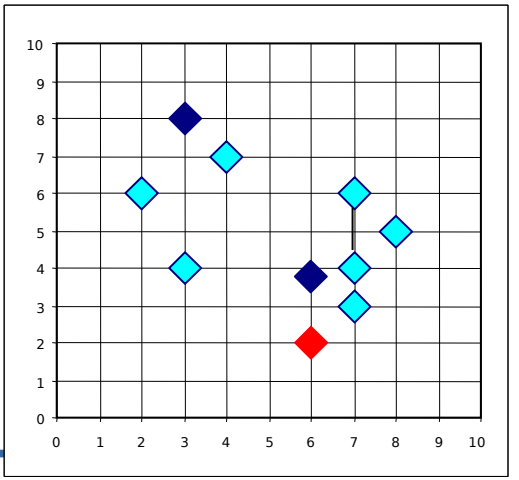
Choisir k objets comme médoïdes initiaux



Affecter chacun des autres objets au plus proche médoïde

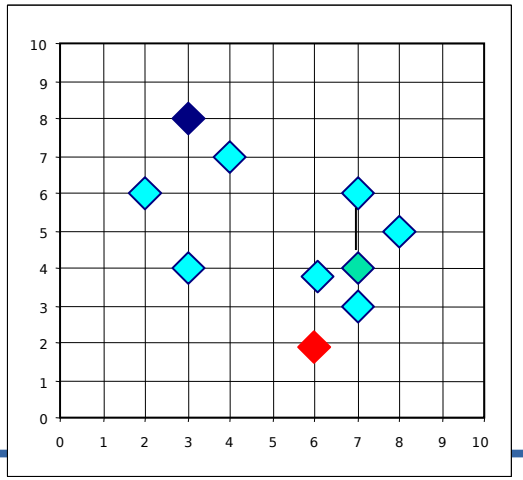


Choisir aléatoirement un objet non-médoïde O_{random}



Calculer le coût total de l'échange

Coût Total = 26



Tant qu'il y a changement

Échanger O et O_{random}

si la qualité s'améliore.

Ensembles Flous

- Un ensemble « classique » est complètement spécifié par une fonction caractéristique $\chi : U \rightarrow \{0, 1\}$, telle que, pour tout $x \in U$,
  - $\chi(x) = 1$, si et seulement si x appartient à l'ensemble
  - $\chi(x) = 0$, autrement.
- Pour définir un ensemble « flou », on remplace χ par une fonction d'appartenance $\mu : U \rightarrow [0, 1]$, telle que, pour tout $x \in U$,
  - $0 \leq \mu(x) \leq 1$ est le degré auquel x appartient à l'ensemble
- Puisque la fonction μ spécifie complètement l'ensemble, on peut dire que μ « est » l'ensemble
- Un ensemble classique est un cas particulier d'ensemble flou !
- L'univers U est le référentiel de l'ensemble μ

Fuzzy C-Means

- Une extension floue de l'algorithme *k*-means (avec clusters flous)
- Un objet peut appartenir à plus d'un cluster, à un certain degré

$$0 \leq \mu_k(\mathbf{x}_i) \leq 1$$

$$\sum_{k=1}^{c} \mu_k(\mathbf{x}_i) = 1$$

$$0 < \sum_{i=1}^{N} \mu_k(\mathbf{x}_i) < n$$

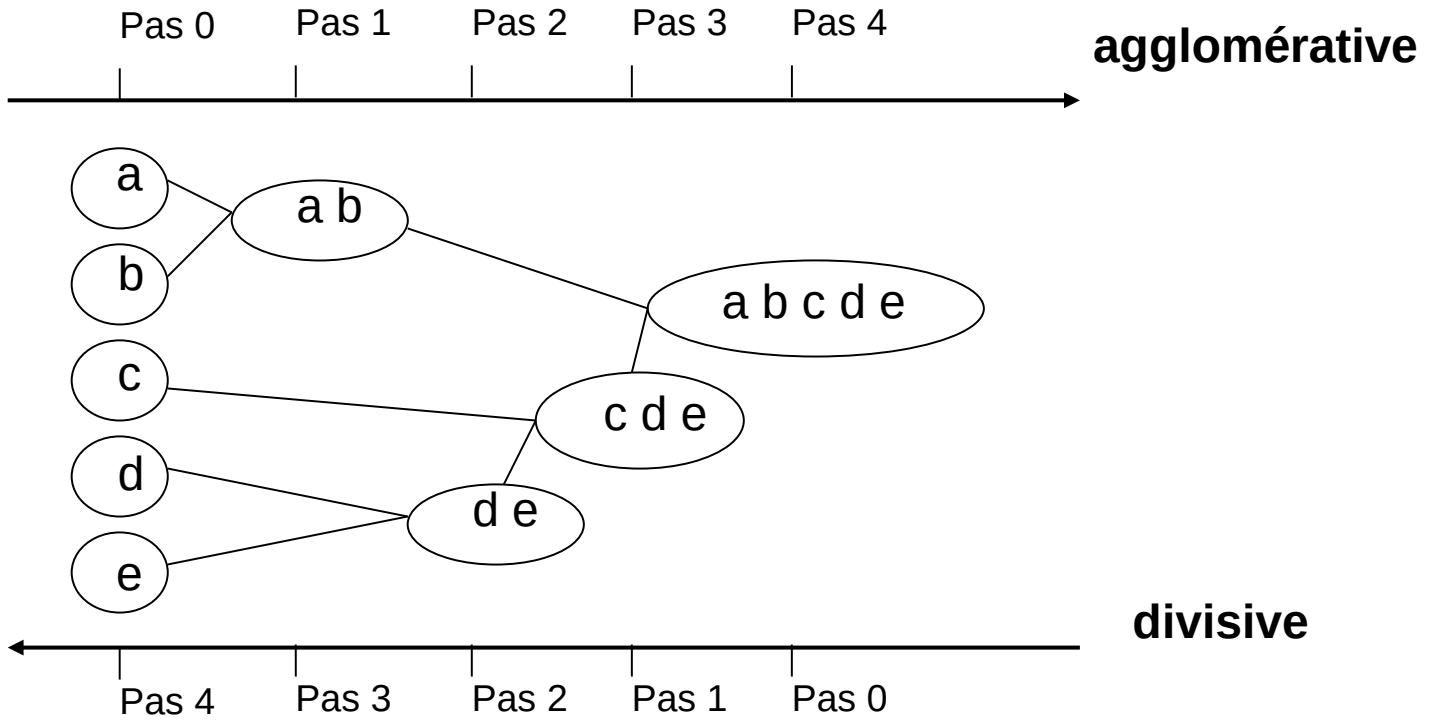
Fonction objectif :

$$\min \sum_{k=1}^{c} \sum_{i=1}^{N} \mu_k(\mathbf{x}_i) d(\mathbf{x}_i, \mathbf{v}_k)$$

Prototype du cluster *k*

Méthodes hiérarchiques

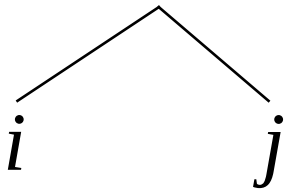
- **Entrée:** matrice des distances **Sortie:** *dendrogramme*
- Cette méthode ne demande pas le nombre des clusters *k* en entrée, mais il a besoin d'une condition de terminaison

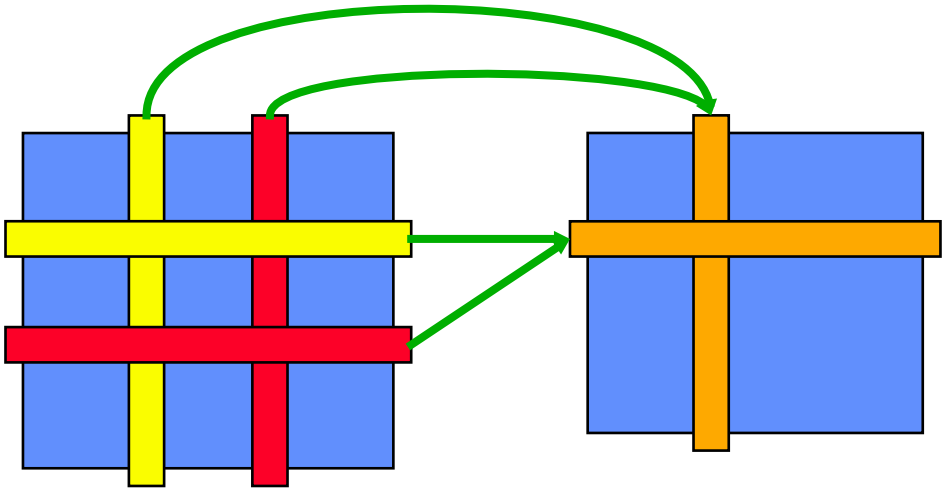


Algorithmes de *Linkage*

1 $$(i, j) = \arg\min_{i,j} d_{ij}$$

2 $\{i, j\}$





3 $$d_{\{i,j\},k} = f(d_{ik}, d_{jk})$$

Fonction de combinaison (min, avg, or max)

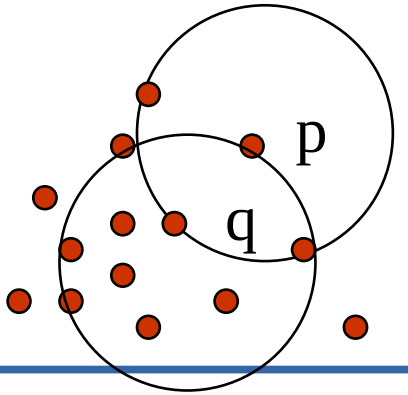
Méthodes basées sur la densité

- Le regroupement se fait sur la base de la densité (critère local), par exemple des points connectés par la densité
- Caractéristiques principales :
  - Peuvent découvrir des clusters de forme arbitraire
  - Gèrent le bruit
  - Une seule passe
  - Besoin de paramètres de densité
- Exemples de méthodes basées sur la densité :
  - DBSCAN, OPTICS, DENCLUE, CLIQUE

Clustering basé sur la densité : notions de base

- Deux paramètres*:*
  - ε : rayon maximum d'un voisinage
  - *MinPts* : nombre minimum de points en un ε-visinage d'un point
- $N_\varepsilon(p)$: $\{q$ dans $D \mid d(p,q) \leq \varepsilon\}$
- Joignabilité directe par densité: un point *p* est directement joignable par densité d'un point *q* étant donnés ε, *MinPts* si
  - *p* appartient à $N_\varepsilon(q)$
  - Condition de noyau :

$$|N_\varepsilon(q)| \geq MinPts$$

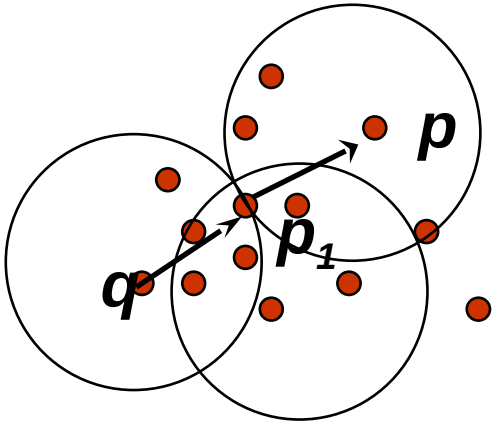


MinPts = 5

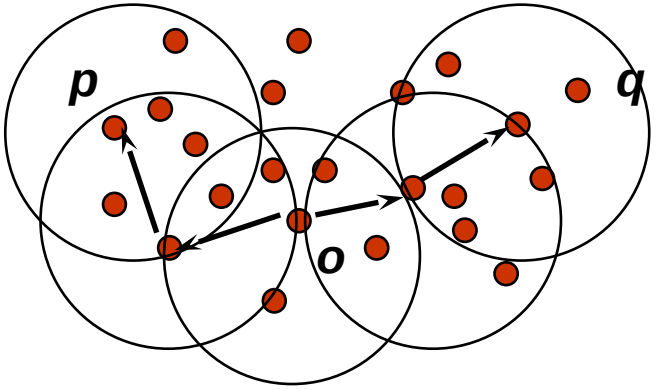
ε = 1 cm

Joignabilité par densité et connexion par densité

- Joignable par densité:
  - Un point *p* est joignable par densité à partir d'un point *q* étant données ε, *MinPts* s'il existe une chaîne de points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ telle que p_{i+1} est directement joignable par densité à partir de p_i



- Connexion par densité
  - Un point *p* est connecté par densité à un point *q* étant donnés ε, *MinPts* s'il existe un point *o* tel que *p* et *q* sont joignables par densité à partor de *o* étant donnés ε et *MinPts*



EM — Expectation Maximization

- Un algorithme de raffinement itératif basé sur un modèle probabiliste
- Extension de *k*-means
  - Un cluster est une distribution de probabilités sur les variables des objets
  - L’appartenance d’un objet à un cluster est une probabilité
  - Les nouvelles moyennes sont calculées sur la base de ces probabilités
- Idée générale
  - On commence avec une estimation initiale du vecteur des paramètres
  - On recalcule les probabilités des objets sur la base de la mixture de distributions produite par le vecteur des paramètres
  - On utilise ces probabilités recalculées pour mettre à jour les paramètres
  - Les objets appartiennent au même cluster s’ils y sont placés par leur probabilités
- Cet algorithme converge rapidement, mais pas toujours à l’optimum global

L’algorithme EM (Expectation Maximization)

- Au début, on affecte aléatoirement *c* centroïdes de clusters
- On raffine itérativement les clusters en deux étapes :
  - Étape de « Expectation »: on affecte chaque objet X_i au cluster C_j avec probabilité

$$P(X_i \in C_j) = p(C_j \mid X_i) = \frac{p(C_j)p(X_i \mid C_j)}{p(X_i)}$$

$$p(X_i \mid C_j) = \phi(X_i; \mu_j, \sigma_j)$$

  - Étape de « Maximisation »:
    - Estimation des paramètres du modèle (ici, gaussien)

$$\mu_k = \frac{\sum_{i=1}^{N} X_i P(X_i \in C_k)}{\sum_{j=1}^{N} P(X_i \in C_j)} \quad \sigma_k = \frac{\sum_{i=1}^{N} (X_i - \mu_k)^2 P(X_i \in C_k)}{\sum_{j=1}^{N} P(X_i \in C_j)}$$