Classification supervisée et clustering

Mihaela JUGANARU-MATHIEU
mathieu@emse.fr

École Nationale Supérieure des Mines de St Etienne

2019-2020

MINES Saint-Étienne

Bibliographie :

- Oded Maimon · Lior Rokach (Editors) - *Data Mining and Knowledge Discovery Handbook*, Second Edition, Springer 2010
- Jiawei Han · Micheline Kamber - *Data Mining : Concepts and Techniques*, Morgan Kaufmann Publication, 2006

MINES Saint-Étienne

Plan

MINES Saint-Étienne

Objectif

Départ : un ensemble de N observations réelles :

$$\mathcal{D} = \{x_i \in \mathbb{R}^d | i = 1, N\}$$

Une observation (tuple) est composée de d attributs.

Objectif du **clustering** (data segmentation ou **classification non-supervisée**) : trouver un ensemble de catégories ou **clusters** de $\mathcal{D}$: $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_k$ auxquelles les données de $\mathcal{D}$ appartiennent.

Les données d'un cluster doivent être les plus semblables / **similaires** possible.
Les clusters se doivent être homogènes.

MINES Saint-Étienne

Objectif

Formellement : $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k \subset \mathcal{D}$ avec $k \ll N$ et

$$\mathcal{C}_1 \cup \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_k = \mathcal{D}$$

Clustering dur (hard clustering) :

- $\mathcal{C}_i \cap \mathcal{C}_j = \Phi$, pour tout i, j

"Une observation appartient a un seul cluster."

Clustering flou (soft clustering) :

- $x_i \in \mathcal{C}_j$ avec une probabilité p_{ij} et :

$$\sum_{j=1}^{k} p_{ij} = 1, \text{ pour tout } i = 1, N$$

"Une observation appartient à plusieurs clusters."

MINES Saint-Étienne

Classification versus clustering

Classe $\equiv$ cluster

Dans la tâche DM (Data Mining) de classification les classes sont connues à l'avance, mais pas pour le clustering.

Classification : but prédictif, apprentissage supervisé.
Clustering : but descriptif, apprentissage non-supervisé.

MINES Saint-Étienne

Caractéristiques possibles d'un procédé de clustering

- supporte le passage à l'échelle (attention : le clustering appliqué à un échantillon de $\mathcal{D}$ peut fournir des résultats biaisés)
- supporte des attributs hétérogènes (rarement une observation a des attributs homogènes dpvd signification et type)
- découvrir des clusters avec des frontières arbitraires
- avoir un minimum d'information (paramètres) : nombre de clusters
- capacité à détecter le bruit
- travaille de manière incrémentale
- supporte un grand nombre de dimension (d - nombre d'attributs de l'espace d'observations)

MINES Saint-Étienne

Fonction distance

A partir de la matrice qui caractérise $\mathcal{D}$ $(N \times d)$ avec la représentation on peut calculer la matrice de distance entre les observations :

$$\begin{pmatrix} 0 & & & & \\ dist(2,1) & 0 & & & \\ dist(3,1) & dist(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ dist(N,1) & dist(N,2) & \cdots & \cdots & 0 \end{pmatrix}$$

$dist(i,j) = distance(x_i, x_j)$, distance ou dissimilarité.

MINES Saint-Étienne

Fonction distance

Une fonction *dist* distance métrique doit satisfaire :

- $dist(x, x) = 0$ pour tout x
- $dist(x, y) = dist(y, x)$, pour tout x, y (symétrie)

- $dist(x, y) \leq dist(x, z) + dist(z, y)$ (inégalité triangulaire)
- si $dist(x, y) = 0$, alors $x = y$ (définitivité)

Une fonction de **dissimilarité** est une fonction distance métrique qui satisfait aussi :

- $dist(x, y) \leq max(dist(x, z), dist(z, y))$ (inégalité ultramétrique)
- si $dist(x, y) = 0$, alors $dist(z, x) = dist(z, y)$ (régularité)

Plus la dissimilarité est grande, plus les points sont éloignés.
La **similarité** est l'inverse de la dissimilarité.

MINES Saint-Étienne

Distances entre observations numériques

Soient $x, y \in \mathbb{R}^d$, $x = (x_1, x_2, \ldots x_d)$, $y = (y_1, y_2, \ldots y_d)$.

- distance de Minkowski :

$$dist(x, y) = \left(\sum_{j=1}^{d} |x_j - y_j|^g \right)^{1/g}$$

- distance pondérée par $w_j, j = 1, d$:

$$dist(x, y) = \left(\sum_{j=1}^{d} w_j \times |x_j - y_j|^g \right)^{1/g}$$

MINES Saint-Étienne

Distances

La distance de Minkowski devient :

- pour g = 1 - distance de Manhattan :

$$dist(x, y) = \sum_{j=1}^{d} |x_j - y_j|$$

- pour g = 2 - distance euclidienne :

$$dist(x, y) = \sqrt{\sum_{j=1}^{d} |x_j - y_j|^2}$$

- pour $g \to \infty$ distance de Tchebychev :

$$dist(x, y) = \max_j |x_j - y_j|$$

MINES Saint-Étienne

Distance entre observation discrète (binaire ou non)

- distance de Hamming :

$$dist(x, y) = card(\{j | x_j \neq y_j\})$$

- distance de Levenshtein ou distance d'édition : le nombre minimum de pas pour transformer x en y

MINES Saint-Étienne

Distances entre observations binaires

Une observation binaire est formée de **variables binaires** (0 ou 1).

Une variable binaire est symétrique, si les deux valeurs ont une signification équivalente *(exemple fille/garçon dans des observations médicales)*.
Il convient souvent de prendre en compte les variables asymétriques.

MINES Saint-Étienne

Distances entre observations binaires

Une mesure de dissimilarité :

$$dist(x, y) = \frac{r + s}{q + t + r + s}$$

avec r et s le nombre de positions avec des valeurs différentes (0-1 et 1-0) et t et q le nombre de position avec la même valeur 0 ou 1
Si on ignore t (variable asymétrique), on peut prendre la distance de Jaccard :

$$dist(x, y) = \frac{r + s}{q + r + s}$$

MINES Saint-Étienne

Distances entre attributs nominaux

- matching simple :

$$dist(x, y) = \frac{d - m}{d}$$

m le nombre d'attributs à valeurs égale

- transformation en variables binaires en associant pour chaque attribut nominal avec v valeurs v attributs binaires

MINES Saint-Étienne

Distances entre attributs mixtes

On calcule une pondération des distances entre les attributs (ou groupe d'attributs) de même type :

$$dist(x,y) = \frac{\sum_{j=1}^{d} \delta^{(j)} d^{(j)}(x,y)}{\sum_{j=1}^{d} \delta^{(j)}}$$

avec $\delta^{(j)}$ un coefficient de pondération de l'attribut j.

MINES Saint-Étienne

Indices de similarité

- cosinus :

$$cos(x, y) = \frac{x \times y}{|x||y|}$$

- indice de Tanimoto (pour attributs binaires)

$$T(x, y) = \frac{x \times y}{|x|^2 + |y|^2 - x \times y}$$

$x \times y$ est le produit scalaire

- coefficient de Dice :

$$s(x, y) = \frac{2x \times y}{|x|^2 + |y|^2}$$

MINES Saint-Étienne

Texte

Si les observations à examiner sont de type texte :

- prétraitement
- calcul des fréquences de mots dans chaque observation (chaque document) et dans la totalité des documents (corpus)
- sur la base des fréquences calculées construire un tableau de valeurs pour chaque document
- prendre des fonctions distance dédiées comme le cosinus, le produit scalaire ou Jaccard

MINES Saint-Étienne

Texte (2)

Prétraitement :

- séparer en unités (mots)
- corriger les fautes d'orthographe ou remplacer les sigles
- filtrer : éliminer les mots vides (très fréquents et / ou non significatifs) ou ne garder que les mots pré-listés
- lemmatisation : mettre les mots à leur forme invariable
- autres traitements quant à leur classe grammaticale, un dictionnaire, etc.

Considérer l'ensemble des mots présents dans le corpus (termes) : $t_1, t_2, \ldots, t_m$

MINES Saint-Étienne

Texte (3)

Calculer les fréquences des termes pour chaque document et dans le corpus :

- fréquence d'un terme dans un document d :
  $tf(t,d)$ le nombre d'apparitions de t dans d

  Le plus souvent on normalise cette valeur par le max :

$$\frac{tf(t,d)}{\max\limits_{d'\text{-document}} df(t,d')}$$

- fréquence d'un terme dans le corpus :

$$cf(t) = \sum_{d \text{ document}} tf(t,d)$$

- fréquence des documents par rapport à un terme :

$$df(t) = card(\{d\text{-document} | d \text{ contient } t\})$$

MINES Saint-Étienne

Texte (4)

Pour représenter un document, il faut considérer pour chaque document soit le tableau

$$tf(d) = (tf(t_1, d), tf(t_2, d) \dots tf(t_m, d)))$$

soit le tableau

$$tf.idf(d) = tf(d) \times \log \frac{D}{df(t)}$$

(produit scalaire)
D est le nombre total des documents.

MINES Saint-Étienne

Texte (5)

Remarque 1 : la matrice tf est (très) creuse.
Remarque 2 : la matrice tf n'est jamais une matrice carrée et $m = K \times n^{\alpha}$, loi de Heaps sur la taille du vocabulaire (K et α constantes)
Remarque 3 : les fréquences des mots ne sont pas comparables : la fréquence d'un mot dans un corpus est inversement proportionnelle à son rang des fréquences, loi de Zipf (après avoir analysé "Ulysse" de James Joyce)
Remarque 4 : assez peu de lois de probabilité modélisent le nombre d'occurrences d'un terme dans un document (loi de Poisson, distribution de Dirichlet)

MINES Saint-Étienne

Un clustering, deux clusterings ...

Si on fournit des clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ qui forment une partition de $\mathcal{D}$, on se pose la question de caractériser la solution autrement et aussi de donner des critères de choix entre deux solutions de clustering.

MINES Saint-Étienne

Métriques de qualité du cluster

On peut calculer facilement pour un cluster $\mathcal{C}_k$:

- son centre :

$$\mu_k = \frac{\sum_{x_i \in \mathcal{C}_k} x_i}{N_k}$$

avec $N_k = card(\mathcal{C}_k)$

- son diamètre :

$$diam(\mathcal{C}_k) = \max_{x,y \in \mathcal{C}_k} dist(x, y)$$

- inertie :

$$J_k = \sum_{x \in \mathcal{C}_k} dist(x, \mu_k)^2$$

MINES Saint-Étienne

Métriques de qualité du inter-cluster

- distance inter-cluster :

$$dist(\mathcal{C}, \mathcal{C}') = dist(\mu(\mathcal{C}, \mu(\mathcal{C}'))$$

- distance entre les éléments :

$$dist(\mathcal{C}, \mathcal{C}') = \min_{x \in \mathcal{C}, y \in \mathcal{C}'} dist(x, y)$$

- intertie inter-cluster

$$JB = \sum_k N_k dist(\mu, \mu_k)^2$$

avec μ le centre de gravité de $\mathcal{D}$

- indice de Dunn, Manotobis ...

MINES Saint-Étienne

Classes d'algorithmes de clustering

- algorithmes de partitionnement
- algorithmes hiérarchiques (ascendants ou descendants)
- méthodes basées sur les techniques de grille
- méthodes basées sur une connaissance du modèle

MINES Saint-Étienne

Clustering par partitionnement

Si on cherche un clustering fort avec K clusters, cela signifie trouver une partition de $\mathcal{D}$. Il est impossible de construire toutes les partitions et de choisir ensuite la meilleure.

Problème NP-difficile - espace exponentiel.

Idée : générer un partitionnement (pas trop mauvais) et l'affiner jusqu'à obtenir une certaine stabilité.
(algorithme connu aussi comme "partitionnement par centres mobiles")

MINES Saint-Étienne

Algorithme de K-moyennes

Algorithme de K-moyennes

Idée : Hugo Steinhaus 1957, algorithme : James MacQueen 1967

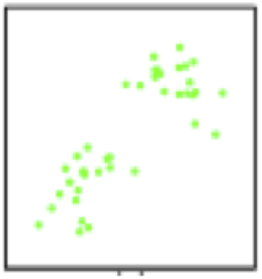
L'algorithme de K-moyennes (K-*means*) : on cherche un clustering fort avec K clusters avec des centres de clusters mouvants. On s'arrête quand le partitionnement devient stable.

- pas 0 : choix aléatoire de K centres parmi les N éléments de $\mathcal{D}$
- pas 1 : si on connaît le μ_k, $k = 1, K$, on affecte un élément x_i au cluster $(C)_k$ le plus proche
- pas 2 : on calcule pour chaque $(C)_k$ son nouveau centre μ_k
- on alterne pas 1 et pas 2 jusqu'à la convergence

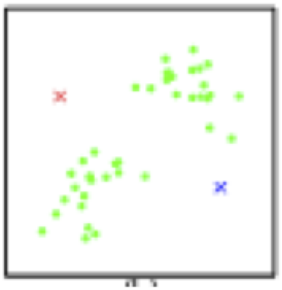
MINES Saint-Étienne

Exemple K=2

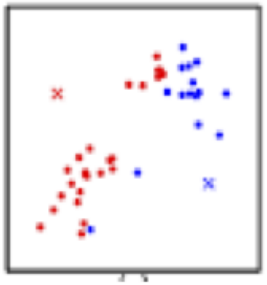
Données



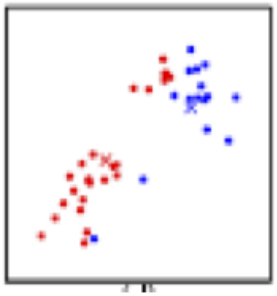
Initialisation



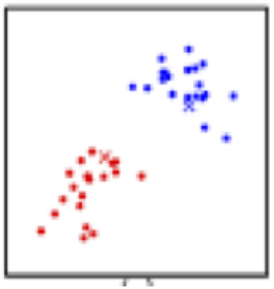
Itération 1



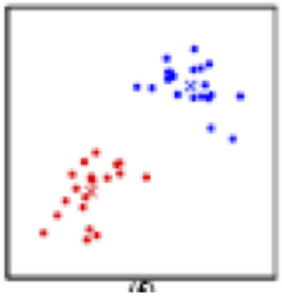
Itération 2



Itération 3



Itération 4



MINES Saint-Étienne

Algorithme K-moyennes

Avantages :

- efficace : complexité en $\mathcal{O}(KNt)$ – K : nb clusters, N : nb objets, t : nb itérations – en général $K \ll t \ll N$
- interprétation aisée des résultats
- le centroid (centre de gravité) caractérise le cluster

Désavantage :

- nécessité de fixer K
- sensible au bruit et données hors-rang (outliers) $\rightarrow$ besoin éventuel de normalisation et de centrage
- clusters convexes uniquement
- valeurs numériques

MINES Saint-Étienne

Fixer k le nombre de clusters :

- k connu (par les experts et donnée dans la définition du problème)
- méthode du coude : on représente sur un graphique les inerties intra-cluster (mieux - la moyenne des inerties pour 10 à 20 exécutions) et on trouve un point de fléchissement (W ou $\log W$)

MINES Saint-Étienne

Fixer k le nombre de clusters (suite) :

- méthode de Tibshiranni (2001) : usage d'une courbe des statistiques de Gap

$$G(k) = \log W_k - E(k)$$

  - W_k la variance attendue pour un partage en k clusters d'une distribution uniforme
  - $s_k = E(k)$ la variance observé pour l'inertie inter-cluster

  La valeur optimale pour k :

$$k^* = min\{k \| G(k) \geq G(k+1) - s'_k\}$$

avec $s'_k = s_k\sqrt{1 + 1/\alpha}$ une correction de s_k pour les α exécutions

MINES Saint-Étienne

Version quant au choix des centres initiaux de cluster :

- choix aléatoire de k éléments parmi les k observations (version de Llyod)
- choix aléatoire d'un cluster pour chaque élément et calcul des moyennes
- choix aléatoire des points quelconque dans l'espace de valeurs possibles
- version $k++$ (2007 , D. Arthur et S. Vassilvitskii) : choix d'un premier centre, pour tout observation on calcule $D(x)$ la distance au plus proche centre, on choisi un nouveau centre avec une probabilité proportionnelle à $D(x)$

MINES Saint-Étienne

Algorithme K-médoïdes

PAM - Partitioning Around Medoids

Proposé par Kaufmann and Rousseeuw (1987)

- permet de traiter des valeurs non-numériques
- un cluster est représenté par un objet lui appartenant : un médoïde - qui a la propriété d'être à distance minimale des autres objets
- principe : remplacement itératif des médoïdes par un objet non-médoïde

MINES Saint-Étienne

Algorithme K-médoïdes

- pas 0 : choix aléatoire de K objets de $\mathcal{D}$ comme médoïdes initiaux
- pas 1 : assigner chaque objet restant de $\mathcal{D}$ au médoïde le plus proche
- pas 2 : choix aléatoire d'un objet $x \in \mathcal{D}$ qui n'est pas médoïde
- pas 3 : pour chaque cluster calculer le coût de remplacement du médoide de $\mathcal{C}_k$ par x (k varie entre 1 et K)
- pas 4 : si le coût est meilleur : remplacer ce médoïde par x
- on reprend pas 1 jusqu'au pas 4 jusqu'à la convergence

MINES Saint-Étienne

Algorithme K-médoïdes

Avantages :

- moins sensible au bruit (usage de la médiane au lieu de la moyenne)
- cluster représenté par un objet

Désavantage :

- temps plus long (pas 2 et 3 en plus)

MINES Saint-Étienne

Méthodes hiérarchique de clustering

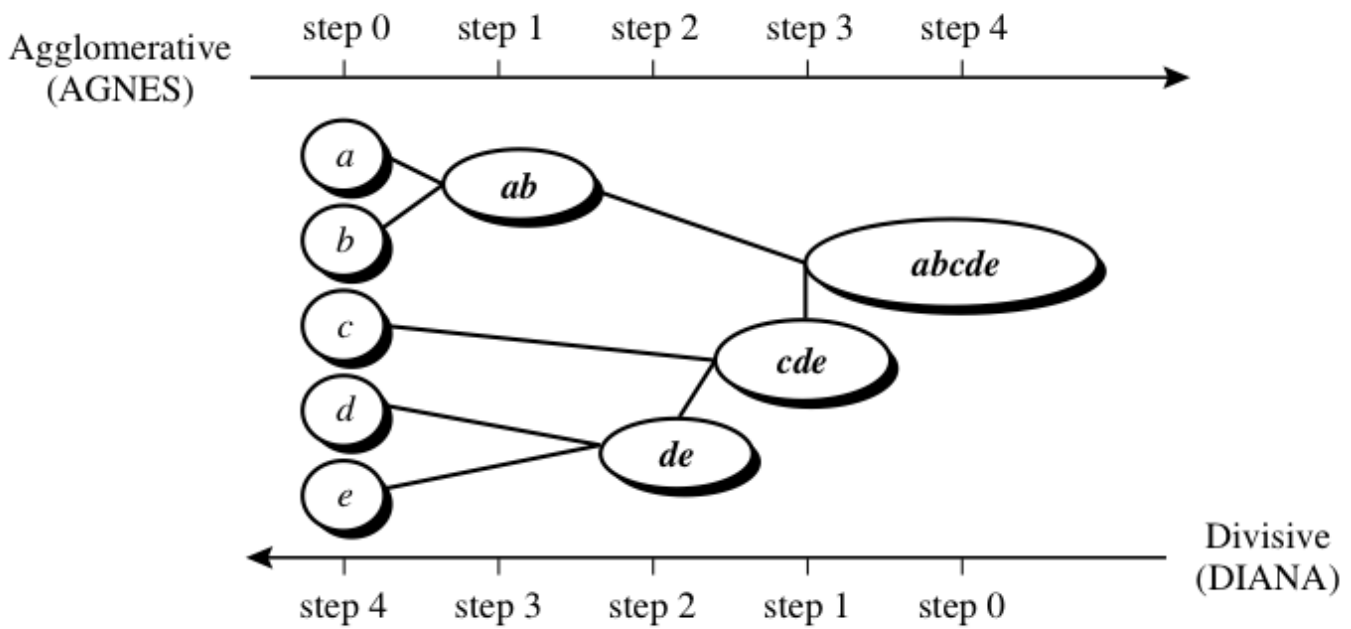
Deux classes de clustering hiérarchique :

- agglomerative (bottom - up)
  - chaque objet forme un cluster
  - on fusionne les clusters en clusters plus grands
  - jusqu'à obtenir un seul
- divisive (top-down)
  - on commence par considérer un unique cluster qui contient tous les objets
  - on divise en clusters plus petits
  - instable, difficilement applicable

MINES Saint-Étienne

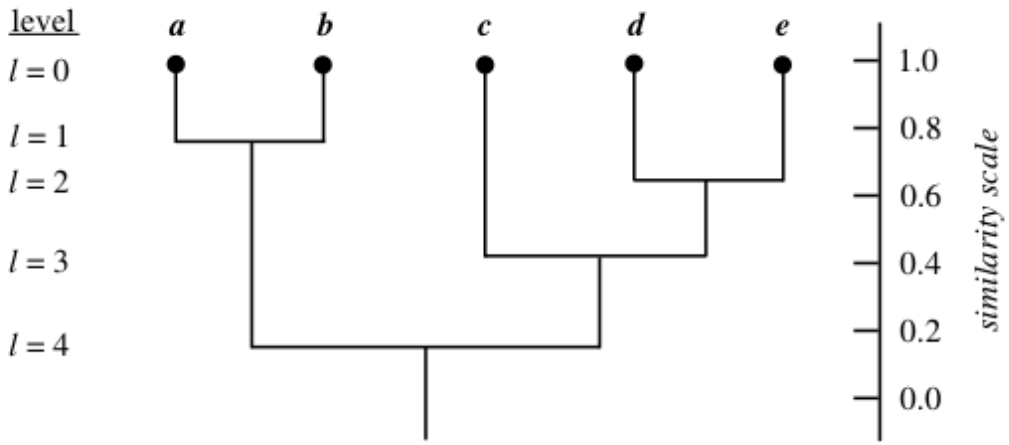
Méthodes hiérarchique de clustering

AGNES (AGglomerative NESting)
DIANA (DIvisive ANAlysis)



Méthodes hiérarchique de clustering

On représente le résultat par un dendogramme :



Pour un clustering avec K clusters on coupe à une certaine hauteur.

Algorithme hiérarchique ascendant

- initialisation $C_i \leftarrow x_i,\ i = 1, N$
  N clusters initiaux, formé chacun d'un seul élément
- répéter $N - 1$ fois
- $\rightarrow$ parmi toutes les paires de clusters, choisir celle qui satisfait mieux un critère

MINES Saint-Étienne

Méthodes hiérarchique de clustering

La fusion/division de cluster se décide selon des **stratégies** :

- distance minimale : saut minimal (single linkage)

$$d_{min}(C_j, C_k) = \min_{x \in C_j, x' \in C_k} dist(x, x')$$

- distance maximale : saut maximal (complete linkage)

$$d_{max}(C_j, C_k) = \max_{x \in C_j, x' \in C_k} dist(x, x')$$

- distance entre les barycentres :

$$d_G(C_j, C_k) = dist(centre(C_j), centre(C_k))$$

- distance moyenne :

$$d_{avg}(C_j, C_k) = \frac{1}{card(C_j)card(C_k)} \sum_{x \in C_j, x' \in C_k} dist(x, x')$$

MINES Saint-Étienne

Méthode Ward pour clustering hiérarchique

Famille d'algorithmes de Lance–Williams :
Soient C_i, C_j et C_k clusters qui existent et on envisage de fusionner $C_i \cup C_j$.
On note $dist_{ij}$, $dist_{jk}$ et $dist_{ik}$ les distances entre les clusters et $dist_{(ij)k}$ une distance qui permet de voir le bénéfice de former le cluster $C_i \cup C_j$.

$$dist_{(ij)k} = \alpha_i dist_{ik} + \alpha_j dist_{jk} + \beta dist_{ij} + \gamma |dist_{ik} - dist_{jk}|$$

avec α_i, α_j, β, γ constantes ou dépendantes de la taille des clusters.

MINES Saint-Étienne

Méthode Ward pour clustering hiérarchique (2)

- la **méthode de Ward** est un cas particulier de Lance-Williams :

  $$\alpha_l = \frac{n_l + n_k}{n_i + n_j + n_k},\ \beta = -\frac{n_k}{n_i + n_j + n_k} \text{ et } \gamma = 0$$

  Pour une paire candidate on calcule toutes les distances par rapport aux autres clusters :

  $$d_{(ij)k} = \frac{n_i + n_k}{n_i + n_j + n_k} d_{ik} + \frac{n_j + n_k}{n_i + n_j + n_k} d_{jk} - \frac{n_k}{n_i + n_j + n_k} d_{ij}$$

  méthode Ward : une méthode des plus utilisées, car elle vise réduire l'inertie inter-cluster.

- On se résume parfois à minimiser uniquement un facteur calculé sur la base de clusters C_i et C_j et de leurs barycentres μ_i, μ_j :

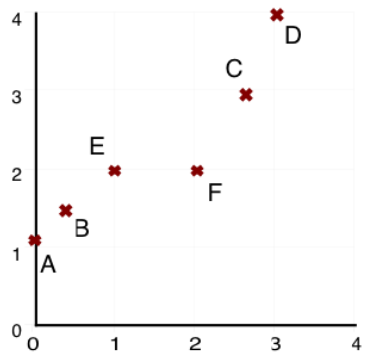
  $$\delta(C_i, C_j) = \frac{n_i + n_j}{n_i n_j} d^2(\mu_i, \mu_j)$$

MINES Saint-Étienne

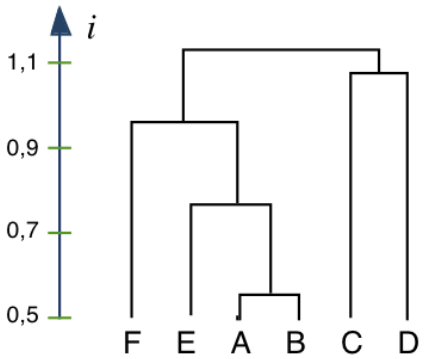
Méthodes hiérarchique de clustering

Selon l'ultramétrique utilisée le résultat est différent :

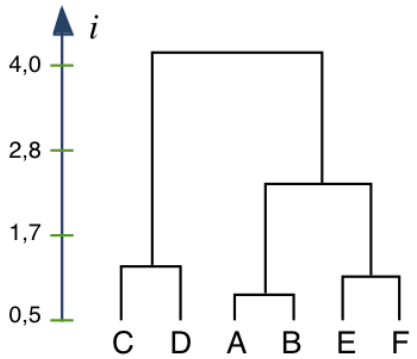
Données (métrique : dist. Eucl.)



Saut minimal



Saut maximal



MINES Saint-Étienne

Méthodes hiérarchique de clustering

Avantage :

- pas besoin de fixer K
- peut s'appliquer (certaines ultramétrique) à des données non-numériques

Désavantages :

- mauvaise complexité : au moins $O(N^2)$
- aucune remise en cause d'une solution partielle

Il y a beaucoup d'algorithmes de clustering hiérarchique : BIRCH (1996), CURE (1998), CHAMELEON (1999)

MINES Saint-Étienne

DBSCAN

DBSCAN - Density-Based Spatial Clustering of Applications with Noise

Algorithme proposé en 1996 par par Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu.

On se base sur une notion de densité de cluster exprimée par deux valeurs : *MinPt* le nombre minimum de points dans un cluster et ϵ un rayon à l'intérieur les points sont considérés comme "proches" et faisant partie d'une même cluster.

MINES Saint-Étienne

Principe : on visite tous les points de l'ensemble des données initial $\mathcal{D}$.

Si le point a peu de voisins (moins que *MinPt*) dans un rayon de ϵ, on considère ce point comme étant du bruit et on ne construit pas de cluster.

Si le voisinage est assez important, on explore récursivement les points de ce voisinage en augmentant le voisinage à explorer avec les voisins des voisins.

MINES Saint-Étienne

DBSCAN($\mathcal{D}$, *MinPts*, ϵ)
Pour chaque point p pas encore visité :

- $Voisinage \leftarrow Voisin(p, \epsilon)$ (y compris p)
- si $card(Voisinage) < MinPts$ alors p est bruit, ignorer
- sinon C nouveau numéro de cluster et $Expand(\mathcal{D}, MinPts, \epsilon, p, C, voisinage)$

MINES Saint-Étienne

Expand($\mathcal{D}$, *MinPts*, ϵ, p, C, *Voisinage*) est une fonction qui parcourt au plus loin possible l'ensemble de données ($\mathcal{D}$) à partir de p et *Voisinage* :

- ajouter p au cluster C
- pour chaque point $p' \in$ *Voisinage*
  - si p' n'est pas encore visité
    - marquer p' comme visité
    - $BVoisinage \leftarrow Voisin(p', \epsilon)$
    - si $card(Bvoisinage) >= MinPts$, alors $Voisinage \leftarrow Voisinage \cup BVoisinage$
  - si p' n'est pas encore dans un cluster, ajouter p' à C
  - éliminer p' de *Voisinage*

MINES Saint-Étienne

Avantages :

- capable de détecter des clusters de toute forme, y compris concaves
- capable d'éliminer les outliers
- pas besoin de savoir le nombre de clusters

Désavantages :

- temps de calcul important et espace mémoire non-négligeable
- on doit avoir des informations sur la densité.

MINES Saint-Étienne

Méthodes mixtes

Applicable pour des gros volumes de données.

Trois étapes :

- appliquer l'algorithme de K-moyennes pour K suffisamment grand
- clustering ascendant
- coupure à un certain niveau (consolidation selon un indice de qualité - inerties, indice de Dunn, etc)

MINES Saint-Étienne

Autres méthodes

- algorithme bio-inspiré (algorithmes génétiques)
- cartes de Kohonen (Self Organizing Map ou SOM)

MINES Saint-Étienne

Plan

MINES Saint-Étienne

Classification

La **classification** (appelée aussi **classification supervisée**) consiste à découvrir (apprendre) un modèle sur la base d'un ensemble d'exemples déjà classés, le modèle appris servira à classer des nouvelles observations.

On dispose donc d'un ensemble initial $\mathcal{D}$ (sous ensemble fini de $\mathcal{X}$ univers possibles d'observations) et pour chaque $x \in \mathcal{D}$ on connaît sa classe d'appartenance y, $y \in \mathcal{Y}$ où $\mathcal{Y}$ est un ensemble fini de classes possibles.

MINES Saint-Étienne

La classification comporte généralement deux phases :

- l'**apprentissage** du modèle qui consiste à déterminer :
  - soit une caractérisation de chaque classe $y_i \in \mathcal{Y}$
  - soit une fonction :

$$h : \mathcal{X} \rightarrow \mathcal{Y} \ , h(x) = y, \forall x \text{ (idéalement)}$$

  La fonction h (le modèle) s'appelle **classifieur**.
- une phase de **prédiction** = traitement des nouvelles observations = détecter la classe d'un nouvel objet :

$$\text{pour n'importe quel } x \in \mathcal{X}, \text{ trouver } , \hat{y} \in \mathcal{Y} \text{ tel que } h(x) = \hat{y}$$

Propriétés de $h(x)$:

- $\forall i (x_i, y_i) \in \mathcal{D}$, on veut que $h(x_i)$ prédise la bonne valeur de y_i
- h doit pouvoir prédire les bonnes sorties pour des exemples futurs $x_j \notin \mathcal{D}$

MINES Saint-Étienne

Qualité des méthodes de classification / prédiction

- exactitude : capacité de étiqueter correctement des nouveaux objets
- vitesse de traitement (coûts)
- robustesse : faire des prédictions correctes sur des données fausses ou bruitées
- passage à l'échelle
- interprétabilité : capacité de rendre le résultat de la phase d'apprentissage dans une forme intelligible

MINES Saint-Étienne

Qualité et exactitude d'une classification

Erreurs de classement :

- prédire un exemple dans C alors qu'il appartient à $non(C)$
- prédire un exemple dans $not(C)$, alors qu'il appartient à C

Calcul du taux d'erreur :

- Vrai Positif (True Positive) : exemple prédit dans C et appartient à C
- Vrai Négatif (True Negative) : exemple prédit dans $not(C)$ et appartient à $not(C)$
- Faux Positif (False Positive) : exemple prédit dans C et appartient à $non(C)$
- Faux Négatif (False Negative) : exemple prédit dans $not(C)$ et appartient à C

MINES Saint-Étienne

Qualité et exactitude d'une classification

La matrice de confusion est une matrice $K \times K$ (K nb de classes).
Pour $K = 2$:

		Predicted class	
		C_1	C_2
Actual class	C_1	true positives	false negatives
	C_2	false positives	true negatives

En R on utilise peut utiliser la fonction `table`

MINES Saint-Étienne

Qualité et exactitude d'une classification

Sur la base de la matrice de confusion on peut calculer diverses mesures :

- sensibilité : $sens = \frac{t\_pos}{pos}$
- spécificité : $spec = \frac{t\_neg}{neg}$
- précision : $prec = \frac{t\_pos}{t\_pos+f\_pos}$
- taux d'erreur : $\tau = \frac{f\_pos+f\_neg}{pos+neg}$
- exactitude : $exact = \frac{t\_pos+t\_neg}{pos+neg}$

MINES Saint-Étienne

Qualité et exactitude d'une prédiction

Si les valeurs prédites y_i sont numériques ($\mathbb{R}^d$) et comparables, on peut calculer diverses erreurs pour une prédiction y'_i par rapport à la valeur correcte y_i :

- erreur absolue : $|y_i - y'_i|$
- erreur carré : $(y_i - y'_i)^2$

Sur la base de ces erreurs on peut calculer l'erreur de test ou l'erreur généralisée.

MINES Saint-Étienne

Qualité et exactitude d'une prédiction

On calcule donc les erreurs moyennes :

$$\frac{\sum_{i=1}^{K} |y_i - y'_i|}{K} \text{ ou } \frac{\sum_{i=1}^{K} (y_i - y'_i)^2}{K}$$

ou les erreurs relatives par rapport ou centre $\bar{y}$ (valeur moyenne) de l'ensemble d'apprentissage :

$$\frac{\sum_{i=1}^{K} |y_i - y'_i|}{\sum_{i=1}^{K} |y_i - \bar{y}|} \text{ ou } \frac{\sum_{i=1}^{K} (y_i - y'_i)^2}{\sum_{i=1}^{K} (y_i - \bar{y})^2}$$

MINES Saint-Étienne

Méthode k-plus proches voisins

Depuis 1950 : *k-nearest neighbor (kNN)*

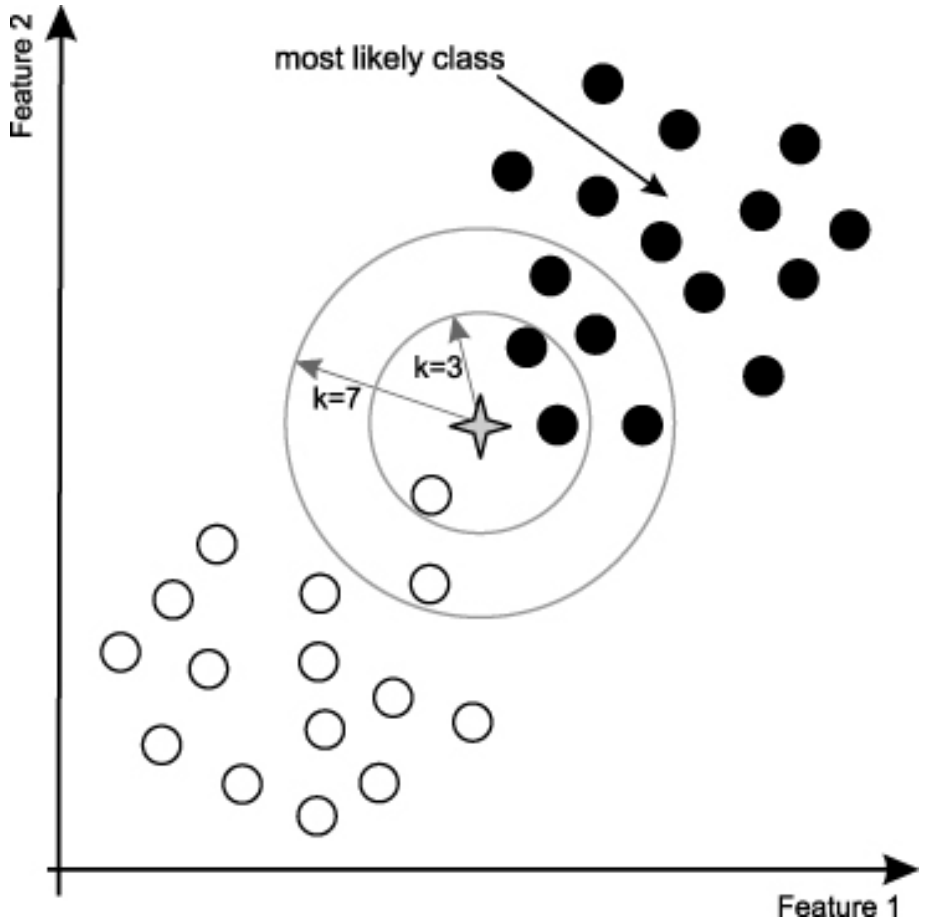
La phase d'apprentissage n'est pas explicitement utilisée (sauf si normalisation).
Pour un élément inconnu x

- on calcule les distances par rapport à tous les éléments de l'ensemble d'apprentissage x_i, $i = 1, N'$
- on le trie ces valeurs
- on choisit la classe majoritaire parmi les k premiers.

Donc parmi les k plus proches voisins on choisit la classe la plus représentée.

MINES Saint-Étienne

k est un paramètre qu'on fixe selon le problème à traiter (taille du problème et signification)



Méthode k-plus proches voisins

On prend, généralement, la distance euclidienne ou Manhatann et k relativement petit.
La méthode s'implémente facilement dans un langage de programmation.