

# Data Mining

## 3. Clustering

Nicolas Pasquier

<http://www.i3s.unice.fr/~pasquier>

# ***Classification***

- Classification : regrouper les instances en groupes ayant une ou des propriétés communes
- Les groupes sont les « classes » distinguées
- Classification non-supervisée
  - Clustering (anglo-saxons) ou *cluster analysis*
  - Les classes ne sont pas connues à l'avance
  - Apprentissage non-supervisée
- Classification supervisé
  - Classification (anglo-saxons) ou classement
  - Les classes sont connues à l'avance
  - Apprentissage supervisée

# Classement

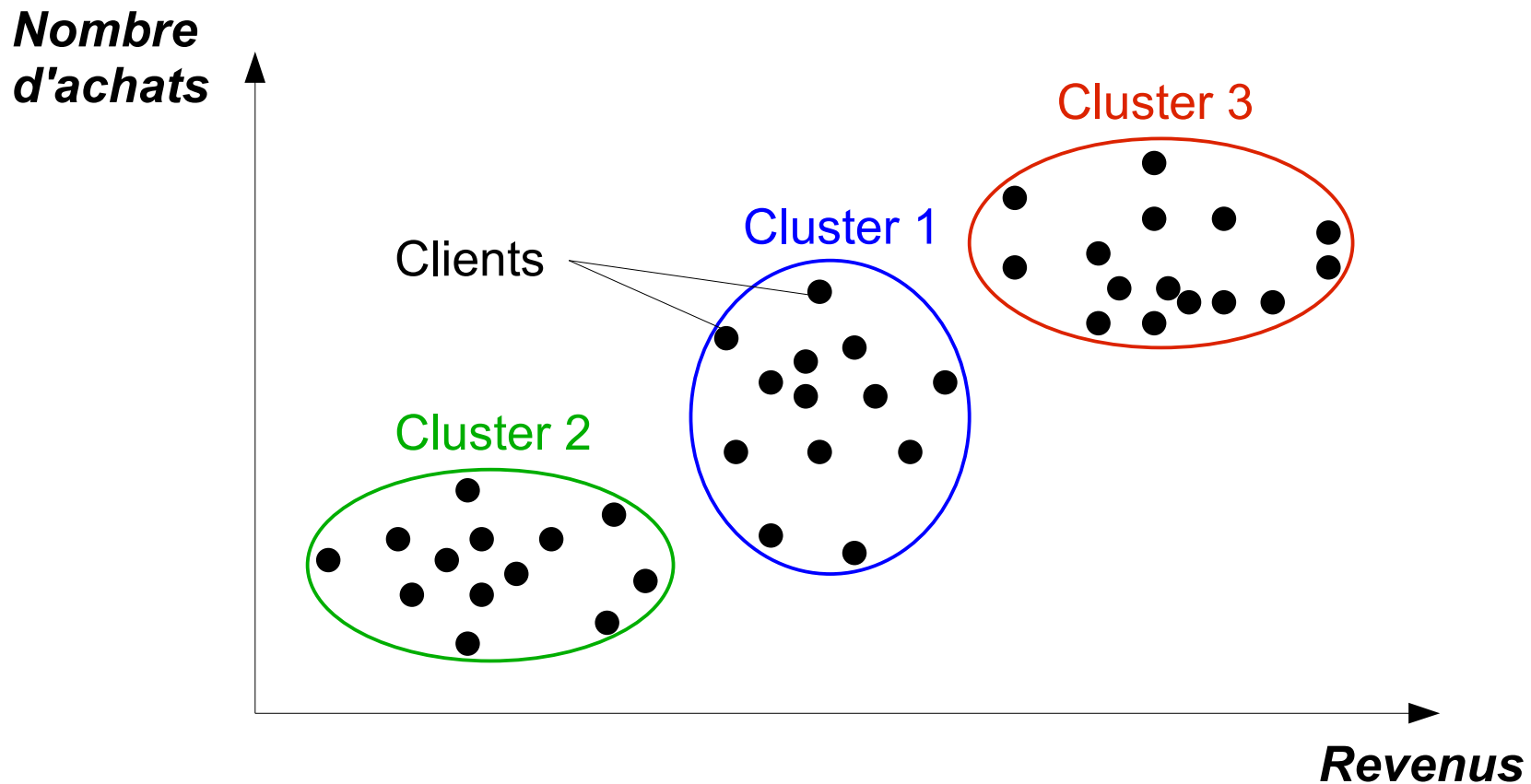
- Classification supervisée (classement)
  - Tâche de prédiction
  - Prédit des variables catégorielles
  - 1. Construire un modèle de classement (classifieur) des données en se basant sur un ensemble appelé ensemble d'apprentissage (EA) (training set)
  - 2. Utiliser le modèle pour classer de nouvelles données
- Exemple
  - On dispose de données sur des patients atteint d'un cancer et d'autres sains
  - On déduit de leurs caractéristiques un modèle
  - Ce modèle est utilisé pour déterminer le risque pour d'autres patients de développer un cancer

# *Clustering*

- Clustering (classification non-supervisée)
  - Tâche de description
  - Recherche des groupes (clusters) dans un ensemble de données
    - avec la plus grande similarité possible intra-groupe
    - et la plus grande dissimilarité possible inter-groupes
- Exemple
  - On dispose de données sur des clients (age, nombre d'enfants, revenus, nombre d'achats, etc.)
  - On regroupe en clusters les clients ayant des caractéristiques communes
  - Pour chaque cluster, on définit une offre commerciale adressée aux clients de ce clusters

# Clustering : exemple

- Représentation bi-dimensionnelle des données



# *Mesures de similarité*

- Objets « suffisamment similaires » regroupés en clusters
  - Définition du seuil de similarité difficile
- Évaluation des clusters
  - Distance entre objets à l'intérieur du cluster
  - Distance avec les objets des autres clusters
- Les données bruitées et les déviations (outliers, objets hors-normes) nuisent à la qualité du clustering

# Mesures de similarité

- Similarité : comparaison des valeurs des variables (attributs)
  - Fonctions de distance :  $d(o_1, o_2)$ 
    - Propriétés en général vérifiées :
      - $d(i, j) \geq 0$
      - $d(i, i) = 0$
      - $d(i, j) = d(j, i)$
      - $d(i, j) \leq d(i, k) + d(k, j)$
  - Fonctions différentes selon le type de variables
    - Numériques linéaires, binaires, nominales, ordinales, par ratios
  - Pondération des variables selon l'application

# Distances de Minkowski

- Mesure classique pour les variables numériques linéaires
  - Objets  $i$  et  $j$  décrits par  $n$  variables  $V_1$  à  $V_n$

$$d(i, j) = \sqrt[q]{|V_1(i) - V_1(j)|^q + |V_2(i) - V_2(j)|^q + \dots + |V_n(i) - V_n(j)|^q}$$

- $q = 1$  : distance de Manhattan
- $q = 2$  : distance Euclidienne
- Pondération de la mesure
  - Donner à chaque variable un poids  $w_i$  selon son importance dans l'application

$$d(i, j) = \sqrt[q]{w_1|V_1(i) - V_1(j)|^q + w_2|V_2(i) - V_2(j)|^q + \dots + w_n|V_n(i) - V_n(j)|^q}$$

# Normaliser les données

- Égaliser le poids des variables pour assurer l'indépendance par rapport aux unités de mesures (ex : Km/h et Miles/h)

- Calculer la déviation absolue moyenne :

$$dev(V_1) = \frac{1}{n} (|V_1(1) - m(V_1)| + |V_1(2) - m(V_1)| + \dots + |V_1(n) - m(V_1)|)$$

- avec :

$$m(V_1) = \frac{1}{n} (V_1(1) + V_1(2) + \dots + V_1(n))$$

- Puis calculer le z-score :

$$z_{V_1}(i) = \frac{V_1(i) - m(V_1)}{dev(V_1)}$$

- Déviation absolue moyenne plutôt que déviation standard

# Variables binaires

- Calculer une matrice de dissimilitude pour les deux objets :

		Objet $j$		
		1	0	Somme
Objet $i$	1	$a$	$b$	$a + b$
	0	$c$	$d$	$c + d$
	Somme	$a + c$	$b + d$	$p$

- Coefficient simple de dissimilitude :
  - Invariante, variables binaires symétriques

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

- Coefficient de Jaccard :
  - Non-invariante, variables binaires asymétriques

$$d(i, j) = \frac{b + c}{a + b + c}$$

# Variables binaires : exemple

- Données :

Nom	Genre	Fièvre	Toux	Test1	Test2	Test3	Test4
Jean	M	Oui	N	P	N	N	N
Marie	F	Oui	N	P	N	P	N
Eric	M	Oui	P	N	N	N	N

- Genre est un attribut symétrique non significatif pour cette application
- Les autres sont des attributs asymétriques
- Les valeurs Oui et P sont transformées en 1
- Les valeurs Non et N en 0

Nom	Fièvre	Toux	Test1	Test2	Test3	Test4
Jean	1	0	1	0	0	0
Marie	1	0	1	0	1	0
Eric	1	1	0	0	0	0

# ***Variables binaires : exemple***

- Attributs asymétriques
  - Pour chaque attribut,  
la co-occurrence de 0 n'est pas informative mais  
la co-occurrence de 1 est informative
  - Coefficient de Jaccard

$$d(\text{Jean}, \text{Marie}) = \frac{0+1}{2+0+1} = 0.33$$

$$d(\text{Jean}, \text{Eric}) = \frac{1+1}{1+1+1} = 0.67$$

$$d(\text{Eric}, \text{Marie}) = \frac{1+2}{1+1+2} = 0.75$$

- Objets les plus proches : Jean et Marie
- Objets les plus distants : Eric et Marie

# ***Variables nominales***

- Généralisation des variables binaires : plus de deux états possibles
  - Ex : couleur  $\in$  {vert, jaune, rouge, bleu}, numéro de département
- Méthode 1 : mesure simple de la similarité

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

- $m$  : nombre de valeurs similaires pour  $i$  et  $j$
  - $p$  : nombre total de variables nominales
- Méthode 2 : utiliser plusieurs variables binaires
  - Créer une variable binaire pour chacune des  $M$  valeurs possibles pour chaque variable
  - Utiliser une mesure non-invariante de la similarité

# ***Variables ordinales***

- Variable ordinaire discrète
  - Variable nominale dont les valeurs sont ordonnées
    - Ex : faible, moyen, fort
- Variable ordinaire continue
  - L'ordre des valeurs est important, pas les valeurs elles-mêmes
    - Ex : résultats d'une épreuve sportive, le classement est plus important

# *Variables ordinales*

- Traitées comme des variables numériques
  - Remplacer  $x_{if}$  par son rang  $r_{if} \in \{1, 2, \dots, M_f\}$
  - Discrétiser les rangs sur  $[0, 1]$  pour assurer l'indépendance par rapport au nombre d'états :

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- Calcul de dissimilitude identique à celui des variables numériques

# ***Variables par ratio***

- Mesures sur une échelle non-linéaire (exponentielle/logarithmique)
  - Ex :  $A.e^{Bt}$  ou  $A.e^{-Bt}$
- Les traiter comme des variables numériques n'est pas un choix judicieux
- Méthode 1
  - Les traiter comme des variables ordinales continues
  - Traiter les rangs comme des valeurs numériques continues

# *Variables par ratio*

- Méthode 2
  - Appliquer une transformation logarithmique
    - Ex :  $y_{if} = \log(x_{if})$
  - Traiter les résultats comme des variables numériques
  - Selon le cas, une autre transformation peut être plus appliquée
    - Ex :  $\log(\log(x_{if}))$
  - Nécessite de connaître le calcul

# Variables hétérogènes

- Formule simple de combinaisons des mesures
  - Objets :  $i$  et  $j$
  - Nombre total de variables :  $n$
  - Distance entre  $i$  et  $j$  pour la variable  $V$  :  $d_V(i,j)$
  - Pondérateur pour la variable  $V$  :  $P_V(i,j)$

$$d(i, j) = \frac{\sum_{V=1}^{V=n} P_V(i, j) \times d_V(i, j)}{\sum_{V=1}^{V=n} P_V(i, j)}$$

- $P_V(i, j) = 0$  si
  - $V(i)$  ou  $V(j)$  indéterminées
  - $V(i) = V(j) = 0$  et  $V$  est binaire asymétrique
- $P_V(i, j) = 1$  sinon

# Variables hétérogènes

- $V$  binaire ou nominale
  - $d_V(i,j) = 0$  si  $V(i) = V(j)$  sinon  $d_V(i,j) = 1$
- $V$  par ratios (numérique exponentielle)
  - Appliquer une transformation logarithmique (ex. :  $\log(V(i))$ )
- $V$  ordinale ou par ratios (numérique exponentielle)
  - Calculer le rang  $R_V(i)$ 

$$Z_V(i) = \frac{R_V(i) - 1}{M_V - 1}$$
  - et traiter  $Z_V(i)$  comme une variable numérique continue
- $V$  numérique continue
  - Distance normalisée

# Évaluation du résultat

## Clusters



## Distance

$$d(2, 4)$$

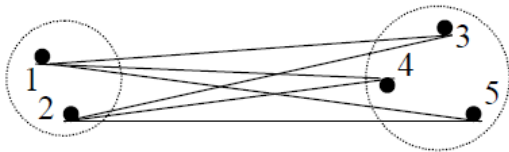
## Mesure

Saut minimal /  
Lien simple



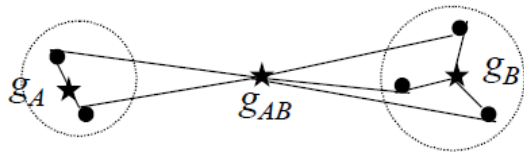
$$d(1, 5)$$

Saut maximal /  
Lien complet



$$\frac{1}{6}(d_{13} + d_{14} + d_{15} + d_{23} + d_{24} + d_{25})$$

Distance  
moyenne



$$I_{AB} - (I_A + I_B)$$

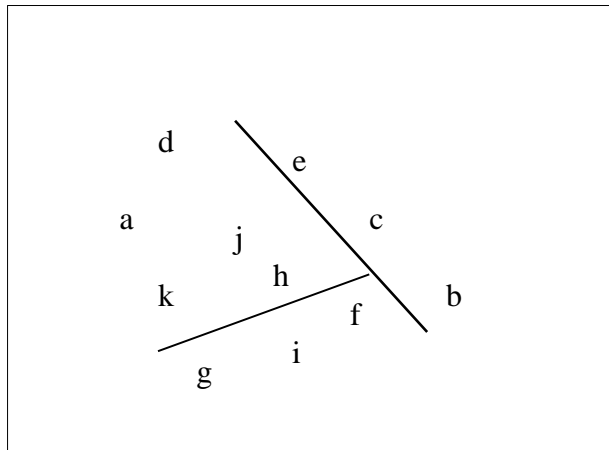
Inertie =  
inertie inter-clusters –  
inerties intra-clusters

# Mesure d'inertie

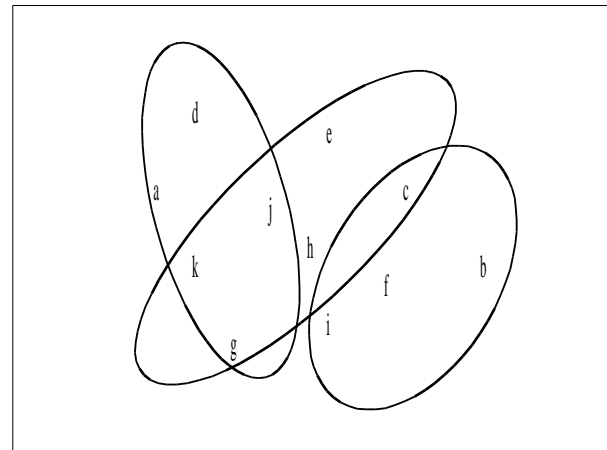
- Inertie d'une population  $I$ 
  - Moyenne des carrés des distances des individus au barycentre
- Inertie intra-clusters  $IA$ 
  - Somme des inerties des clusters (idem inertie d'une population)
  - Minimiser pour obtenir des clusters homogènes
- Inertie inter-clusters  $IR$ 
  - Moyenne (pondérée par l'effectif du cluster) des carrés des distances des barycentres de chaque cluster au barycentre global
  - Maximiser pour obtenir des clusters bien séparés

# *Types de clustering*

- Pour données numériques et symboliques
- Déterministes vs. probabilistes
- Recouvrants ou exclusifs
  - Les clusters peuvent se chevaucher ou non, i.e. un objet peut appartenir à plusieurs clusters ou non



Exclusifs

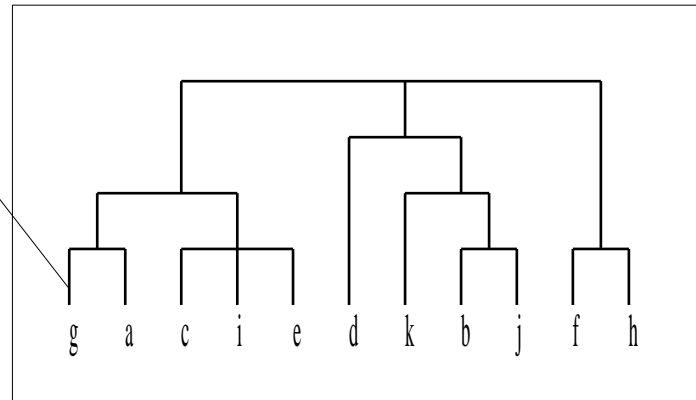


Recouvrants

# *Types de clustering*

- Hiérarchiques ou non
  - Plusieurs niveaux de détail
  - Au niveau inférieur, un cluster est décomposé en plusieurs clusters

Longueurs des arcs = distances



- Méthodes hiérarchiques ascendantes ou descendantes

# *Méthodes de clustering*

- Par partitionnement
  - Construisent différentes partitions, puis les évaluent selon certains critères
- Hiérarchiques
  - Construisent une organisation hiérarchique des objets
- Basées sur la densité
  - Utilisent des fonctions de densité et de connectivité
- Basées sur un modèle
  - Le meilleur modèle de structure est recherché pour chaque cluster

# *Méthodes par partitionnement*

- Principe
  - Construire une partition de l'ensemble des données en  $K$  parties dont chacune représente une classe
- Méthode
  - Initialisation : création d'une partition initiale
  - Puis : itérations d'un processus de remplacement qui optimise le partitionnement en déplaçant les objets d'une classe à l'autre
- Algorithmes
  - K-means : cluster représenté par son centre de gravité
  - K-médoïds : cluster représenté par un objet « central »
  - Nuées dynamiques : cluster représenté par son noyau (objets « centraux »)

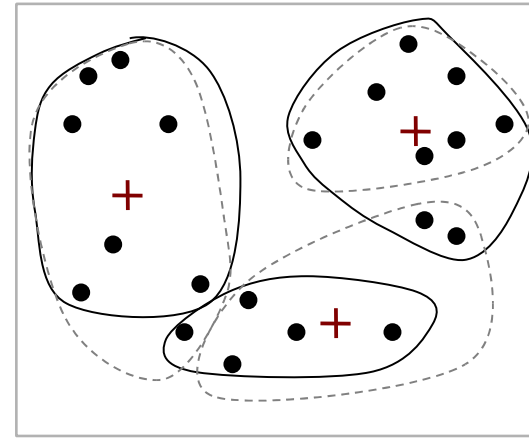
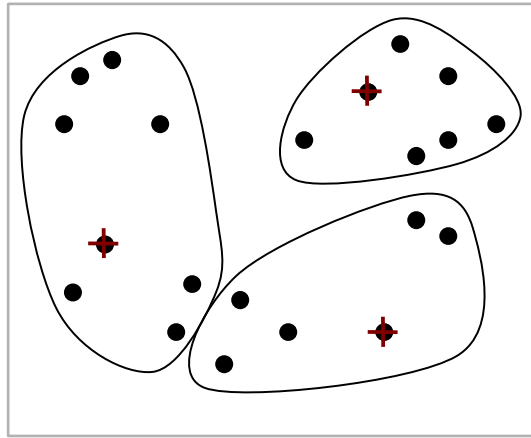
# ***K-moyennes : algorithme***

- Anglo-saxon : K-means

1. Sélectionner aléatoirement  $K$  objets comme centroïdes des clusters initiaux
2. Répéter
3.     Assigner chaque objet  $x$  au cluster dont le centroïde est le plus proche de  $x$
4.     Pour chaque cluster  $C$
5.         Recalculer son centroïde comme moyenne arithmétique des objets de  $C$
6.     Fin pour
7. Jusqu'à ce que les clusters soient stables

# *K-moyennes : exemple*

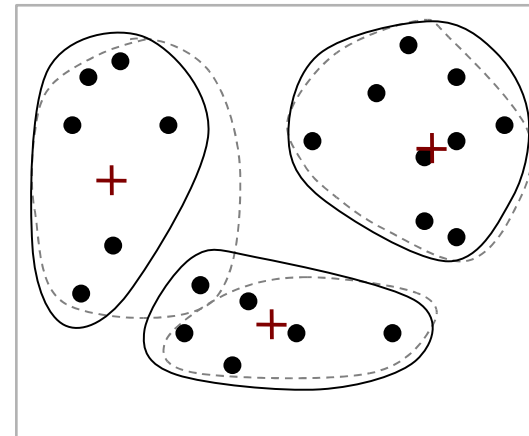
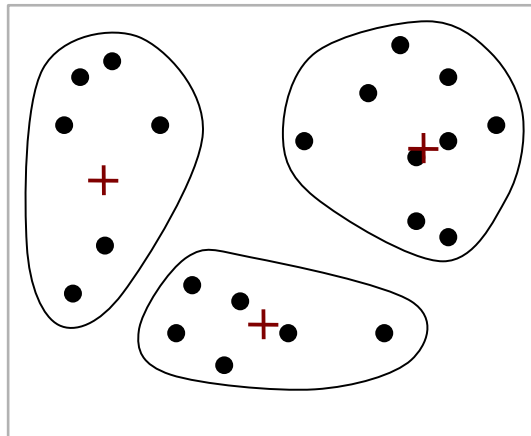
Choix aléatoire de  $k$  objets centres initiaux et calcul des clusters



Calcul des centres des clusters et mise-à-jour des clusters



Arrêt lorsque les clusters sont stables (critère stable)



Mise-à-jour des centres des clusters et mise-à-jour des clusters

# ***K-moyennes : caractéristiques***

- Avantages
  - Efficace : complexité en  $O(knt)$ 
    - $k$  : nbr clusters,  $n$  : nbr objets,  $t$  : nbr itérations
    - En général  $k \ll t \ll n$
  - Interprétation aisée des résultats
    - Centroïde caractérise le cluster
- Inconvénients
  - Nécessité de fixer  $k$
  - Sensible aux exceptions et aux données bruitées
  - Clusters convexes exclusivement
  - Variables numériques seules

# ***K-médoïdes : approche***

- Permet de traiter les données non numériques
- Chaque cluster est représenté par un de ses objets « centraux » : le médoïde
- Médoïde : objet d'un cluster dont la distance avec les autres objets du cluster est minimale
- Principe :
  - Remplacement itératif des médoïdes par un objet non-médoïde si le critère de qualité est amélioré
  - Critère de qualité
- K-medoids, K-modes

# ***K-médoïdes : algorithme***

1. Sélectionner aléatoirement  $K$  objets comme médoïdes initiaux
2. Répéter
3.     Assigner chaque objet restant au médoïde le plus proche
4.     Choisir aléatoirement un objet  $O_r$
5.     Pour chaque médoïde  $O_m$
6.         Calculer le coût  $C$  du remplacement de  $O_m$  par  $O_r$
7.         Si  $C < 0$  alors
8.             Remplacer  $O_m$  par  $O_r$
9.             Calculer les nouveaux clusters
10.         Fin si
11.     Fin pour
12. Jusqu'à ce que les clusters soient stables

# *K-médoïdes vs. K-moyennes*

- K-médoïdes : utilise la médiane au lieu de la moyenne de chaque cluster
  - Moyenne {1, 3, 5, 7, 9} = 5
  - Moyenne {1, 3, 5, 7, 1009} = 205
  - Médiane {1, 3, 5, 7, 1009} = 5
- Avantages
  - La médiane n'est pas affectée par les valeurs extrêmes (exceptions, déviations)
    - Moins sensible aux données bruitées
  - Chaque cluster est représenté par un objet réel : son médoïde

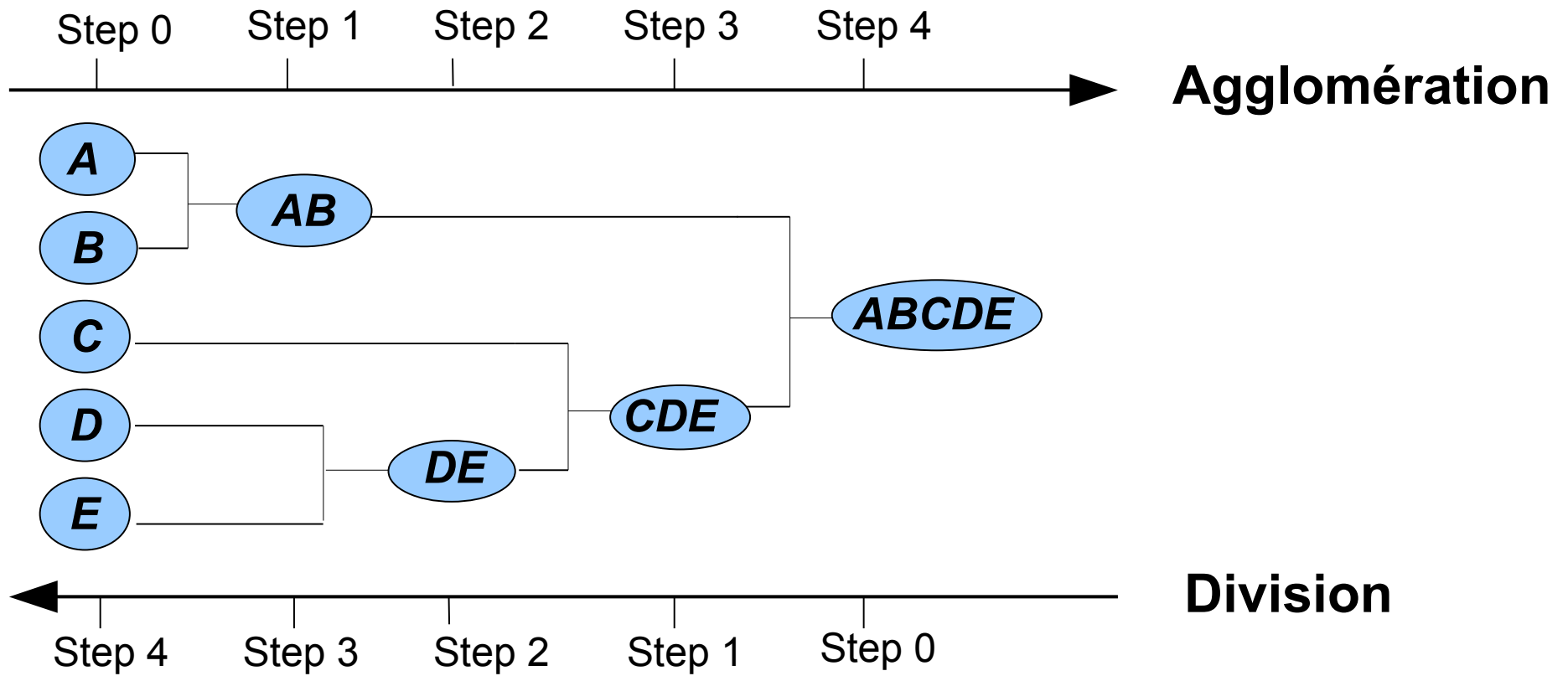
# ***Nuées dynamiques***

- Extension des K-médoïdes
- Chaque cluster est représenté par un ensemble d'objets centraux
- Avantages
  - Plus grande stabilité
  - Moins dépendant du choix initial des médoïdes
- Inconvénients
  - Moins bonnes performances (temps de calcul)

# Méthodes hiérarchiques

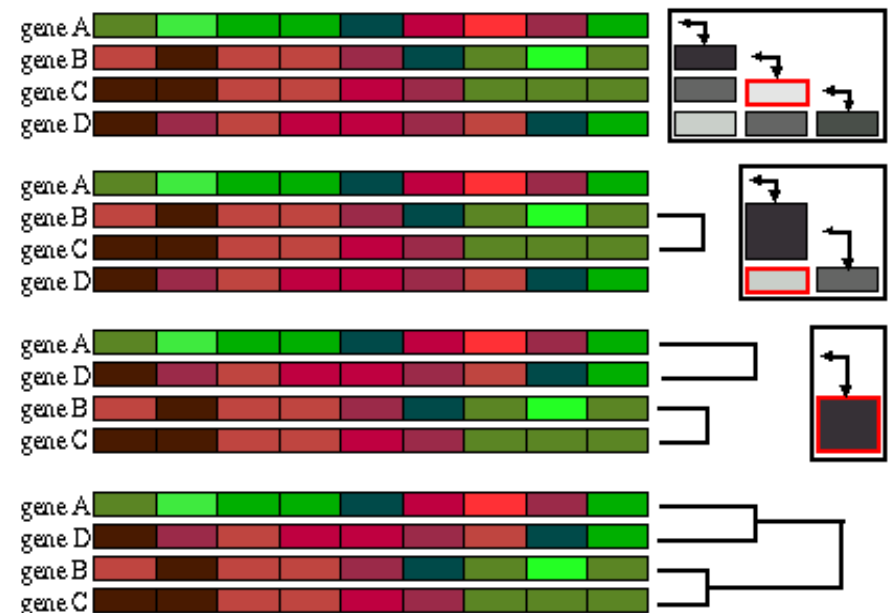
- Créent une décomposition hiérarchique des objets
- Méthodes par agglomérations (*bottom-up*) : AGNES, UPGMA
  - Départ : un cluster par objet
  - Regroupement des clusters les plus proches
  - Arrêt : un seul cluster ou condition d'arrêt vérifiée (ex :  $k$  clusters, inertie  $> \delta$ )
- Méthodes par divisions (*top-down*) : DIANA
  - Départ : un unique cluster contenant tous les objets
  - Séparer les objets ou clusters les plus dissemblables
  - Arrêt : un cluster par objet ou condition d'arrêt vérifiée

# Méthodes hiérarchiques



# Méthodes hiérarchiques

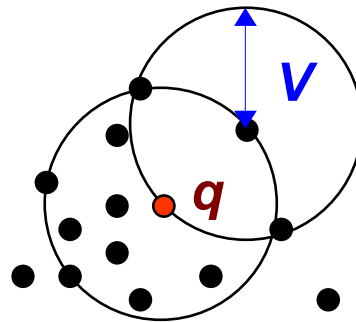
- Représentation hiérarchique des clusters utile pour un certain nombre de problèmes
- Exemple : Génomique
  - Identifier les gènes intervenant dans les processus biologiques (cicatrisation, cancers, etc.)
  - Résultat : dendrogramme
  - Longueur des branches proportionnelle à l'inertie
  - Le nombre de clusters varie selon l'endroit où l'on « coupe »



# Méthodes basées sur la densité

- Objets : points dans l'espace des données
  - Clusters : régions denses séparées par des régions peu denses
- Paramètres
  - $V$  = distance maximale de voisinage
  - $N$  = nombre minimal d'objets dans le voisinage d'un objet coeur
  - Centre d'une zone dense

**Point  
coeur  $q$**



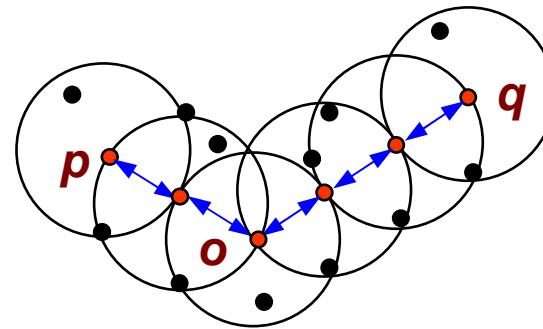
$N = 5$

# Méthodes basées sur la densité

- Algorithme

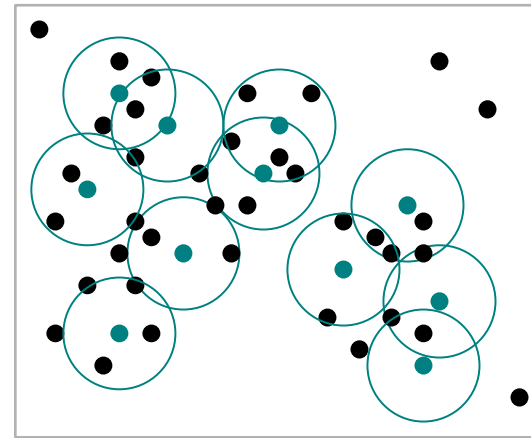
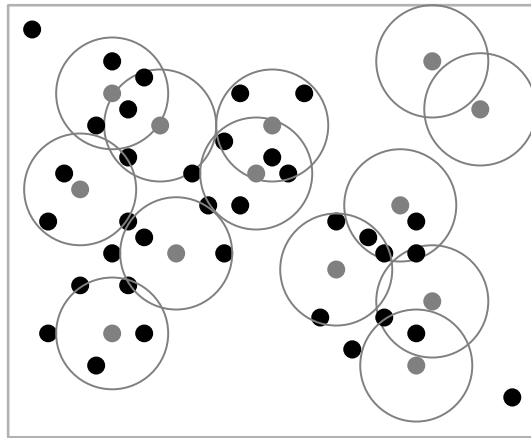
1. Choisir aléatoirement un ensemble d'objets et calculer leur voisinage
2. Identifier les objets coeurs
3. Construire un cluster pour chaque objet coeur
4. Fusionner les clusters d'objets coeurs mutuellement atteignables

**Cluster de points mutuellement atteignables**



# Méthodes basées sur la densité

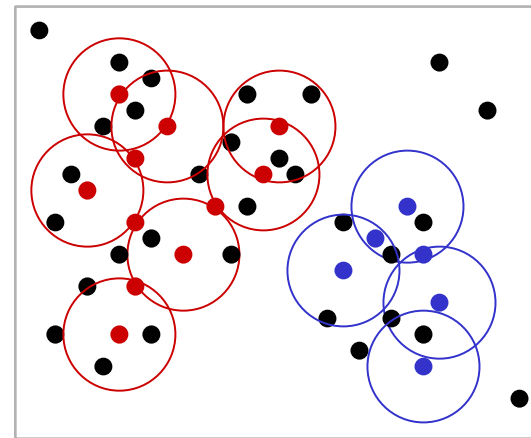
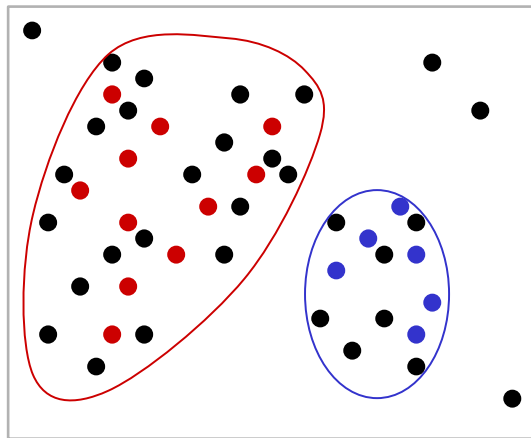
Choix aléatoire de points de départ et calcul de la taille de leur voisinage



Détermination des *objets cœurs*



On obtient des clusters de tailles et de formes différentes



Fusion des clusters dont les objets cœurs sont mutuellement atteignables

# *Méthodes basées sur la densité*

- Avantages
  - Robuste aux données bruitées
    - Apparaissent comme des points isolés
  - Clusters non convexes
    - Peuvent avoir n'importe quelle forme
- Inconvénients
  - Efficace si on dispose d'un index spatial :  $O(n \log(n))$
  - Sinon complexité :  $O(n^2)$
  - Peu adaptée aux attributs symboliques

# ***Construction de grilles***

- Améliore l'efficacité du clustering par densité
- Grille multi-dimensionnelle des données
  - Espace des données découpé en un nombre fini de cellules
  - Cellules : grille multi-dimensionnelle
  - Chaque attribut est une dimension
  - Objets représentés comme des points dans la grille
- La densité de chaque cellule est évaluée
  - Cellule dense si  $\text{nbr objets} > N$
- Les cellules denses connectées forment les clusters

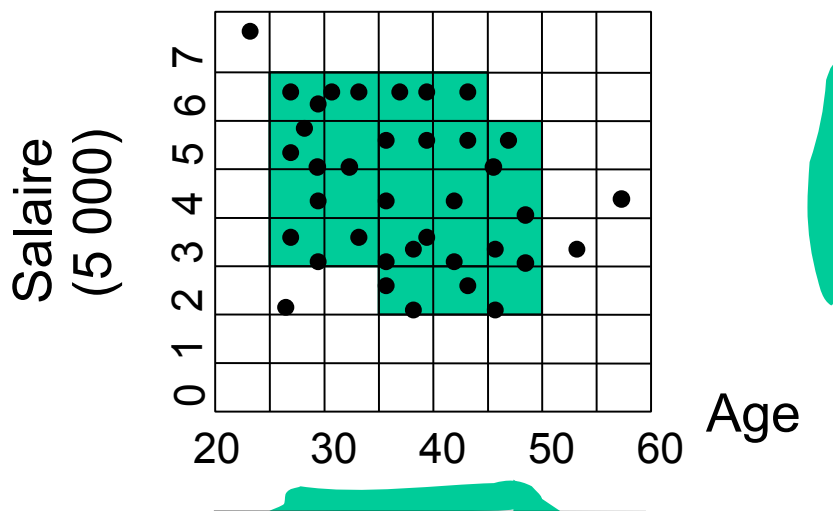
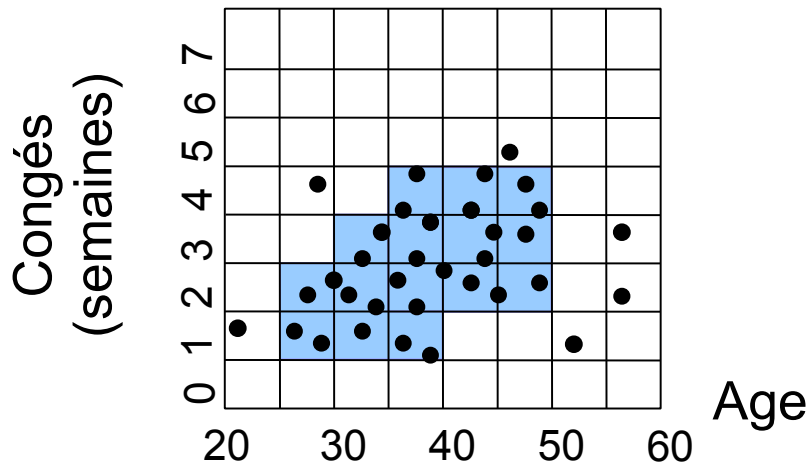
# Construction de grilles

- Algorithme

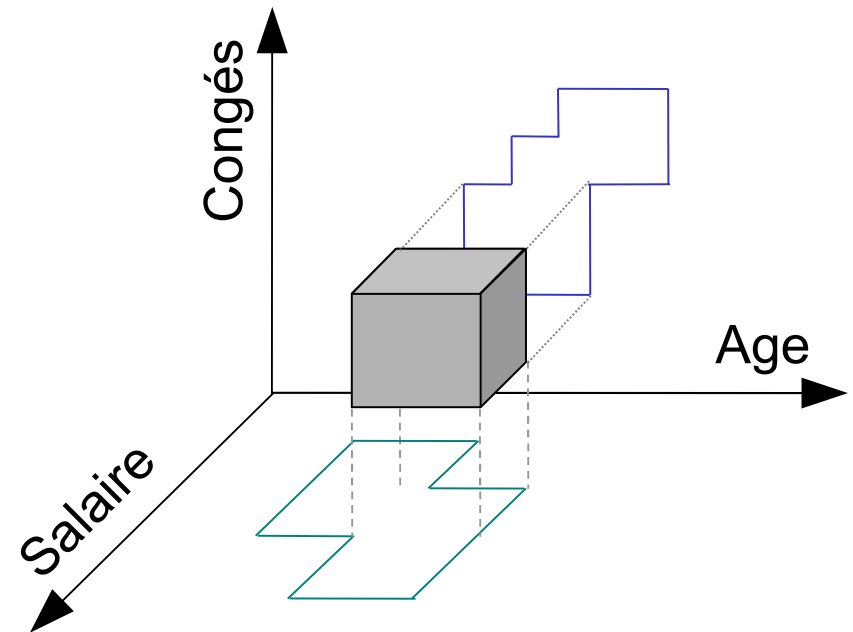
1. Partitionner chaque dimension en intervalles égaux
2. Répéter
3.     Sélectionner les cellules denses
4.     Ajouter une dimension
5. Tant que il existe une dimension non traitée
6. Identifier les régions maximales de cellules denses connectés

# Construction de grilles

- $N = 2$



## Espace 3-dimensionnel



# *Construction de grilles*

- Avantages
  - Bonnes performances : linéaires par rapport au nombre d'objets
  - Bonnes propriétés de croissance par rapport au nombre d'attributs
  - Insensible à l'ordre des objets
  - Processus facile à paralléliser
- Inconvénients
  - Précision dépendante de la méthode de formation des grilles (taille des cellules)
  - Peu adaptée aux attributs symboliques

# *Construction de modèles*

- Méthode adaptée aux attributs symboliques
- Effectue un clustering conceptuel par arbre de décision
  - Identifie des classes (clusters)
  - En donne une description
- Arbre de décision
  - Chaque nœud représente un concept et contient une description probabiliste de celui-ci
  - La description caractérise les objets classés dans ce nœud : probabilité du concept et probabilités conditionnelles sur les valeurs des attributs
  - Les nœuds frères d'un niveau forment une partition

# Construction de modèles

- Démarre à la racine
- Insère les instances une à une
- Met à jour l'arbre à chaque étape
- A chaque étape, 4 actions sont possibles:
  - Fusionner deux nœuds, éclater un nœud, créer un nœud, créer un sous-nœud
  - Objectif : optimiser l'indice Category utility
  - Fonction quadratique définie sur les probabilités conditionnelles

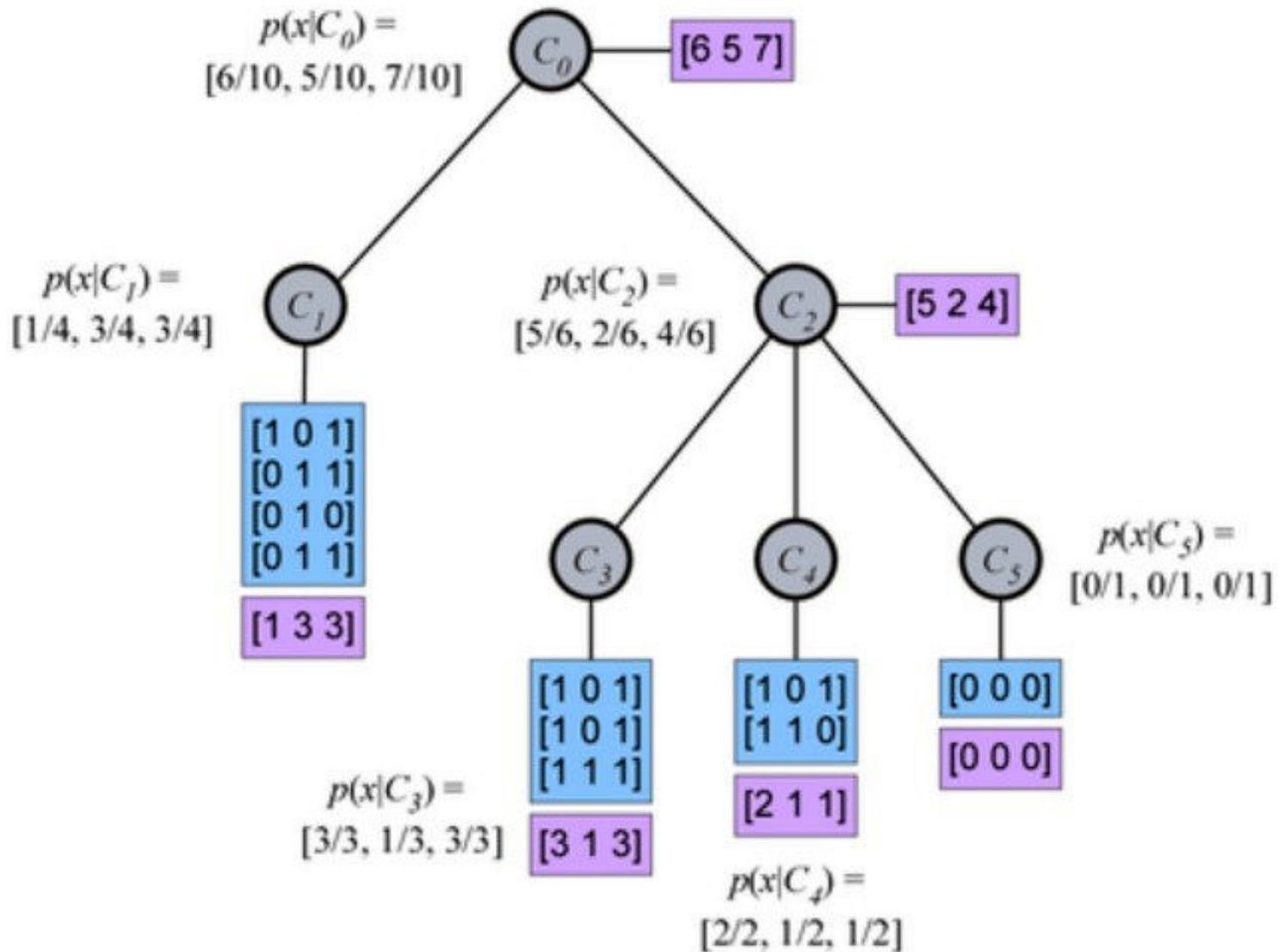
$$CU(C_1, C_2, \dots, C_k) = \frac{\sum_l \Pr[C_l] \sum_i \sum_j (\Pr[a_i = v_{ij} | C_l]^2 - \Pr[a_i = v_{ij}]^2)}{k}$$

# ***Cobweb : exemple 1***

- Jeu de données : description d'animaux

<b>Objet</b>	<b>Mâle</b>	<b>Ailé</b>	<b>Nocturne</b>
<b>A1</b>	1	0	1
<b>A2</b>	0	1	1
<b>A3</b>	1	0	1
<b>A4</b>	1	0	1
<b>A5</b>	0	0	0
<b>A6</b>	1	1	0
<b>A7</b>	1	0	1
<b>A8</b>	0	1	0
<b>A9</b>	1	1	1
<b>A10</b>	0	1	1

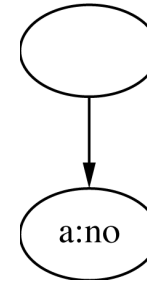
# Cobweb : exemple 1



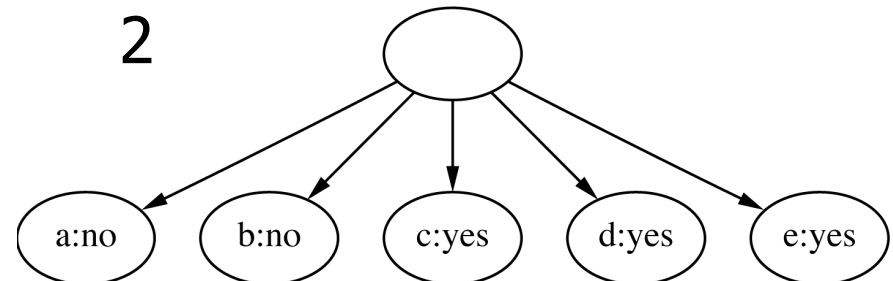
# Cobweb : exemple 2

ID	outlook	temperature	humidity	windy	play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no

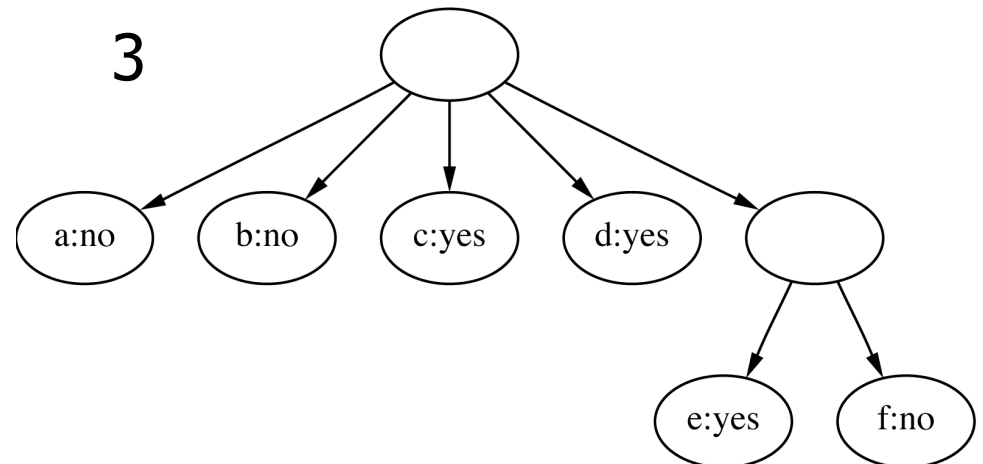
1



2

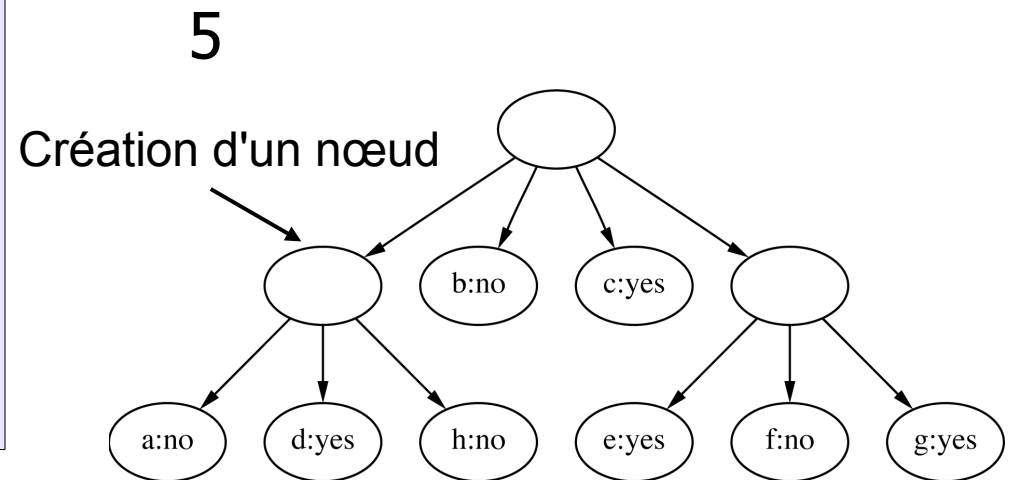
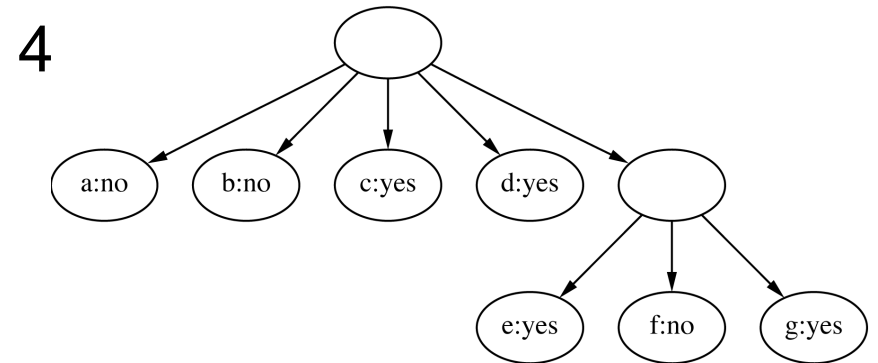


3



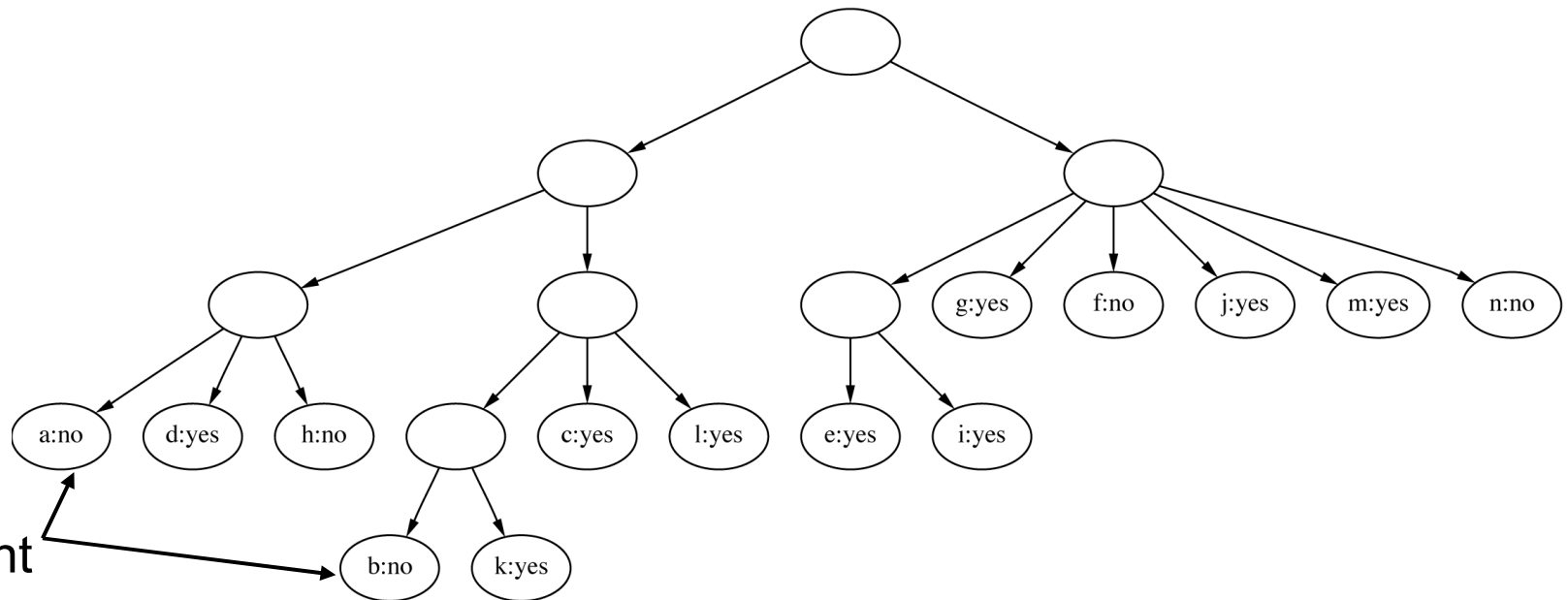
# Cobweb : exemple 2

ID	outlook	temperature	humidity	windy	play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no



# Cobweb : exemple 2

ID	outlook	temperature	humidity	windy	play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
...	...	...	...	...	...



a et b sont  
très  
similaires

# *Choix d'un algorithme*

- Doit tenir compte des caractéristiques de l'application et des données
  - Passage à l'échelle
    - Volume de données à traiter (nbr objets et nbr attributs)
  - Présence de différents types d'attributs
    - Numériques
    - Symboliques (nominaux, ordinaux, binaires)
  - Présence de bruit et déviations dans les données
  - Donner des résultats interprétables
    - Interprétation par l'expert
    - Utilisés dans un processus automatisé

# ***Résumé des méthodes***

- Méthodes par partitionnement
  - Méthodes simples et efficaces
  - Sensibles au bruitage des données
- Méthodes hiérarchiques
  - Clustering à diverses niveaux de détail
  - Applicables seulement aux jeux de tailles réduites
- Méthodes basées sur la densité
  - Méthodes efficaces pour les données bruitées
  - Plus adaptées pour les données spatiales (ex : SIG)
- Construction de modèle
  - Efficace et résultat facile à interpréter
  - Sensible à l'ordre des données