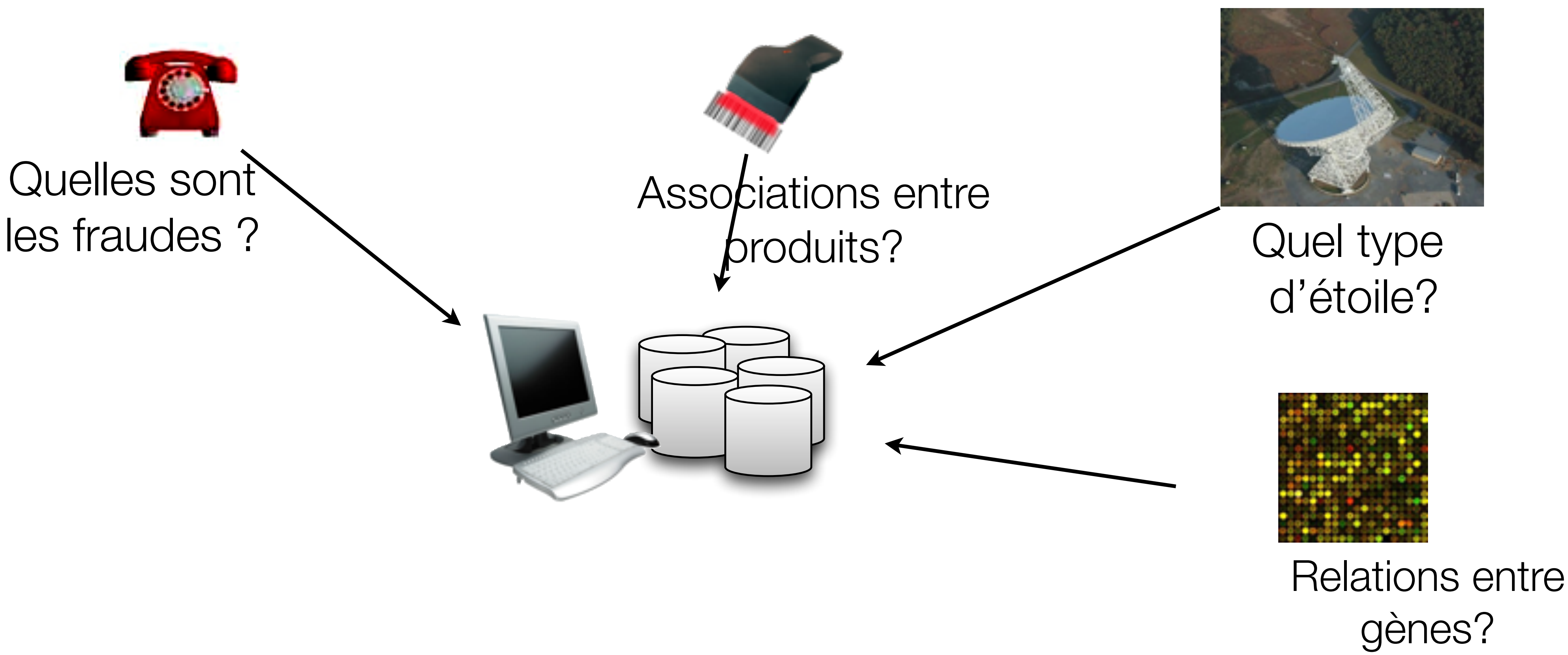
Fouille de Données

Processus ECD, fouille de motifs, clustering

Motivations

- Développement des TICs
    - Gestion et collection de très grands volumes de données

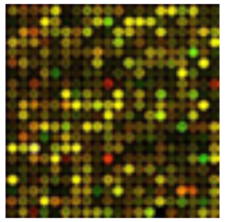


Motivations

- Développement des TICs
  - Gestion et collection de très grands volumes de données

Quelles sont les fraudes ?

pe
le?



Relations entre gènes?

Motivations

- Développement des TICs
  - Gestion et collection de très grands volumes de données

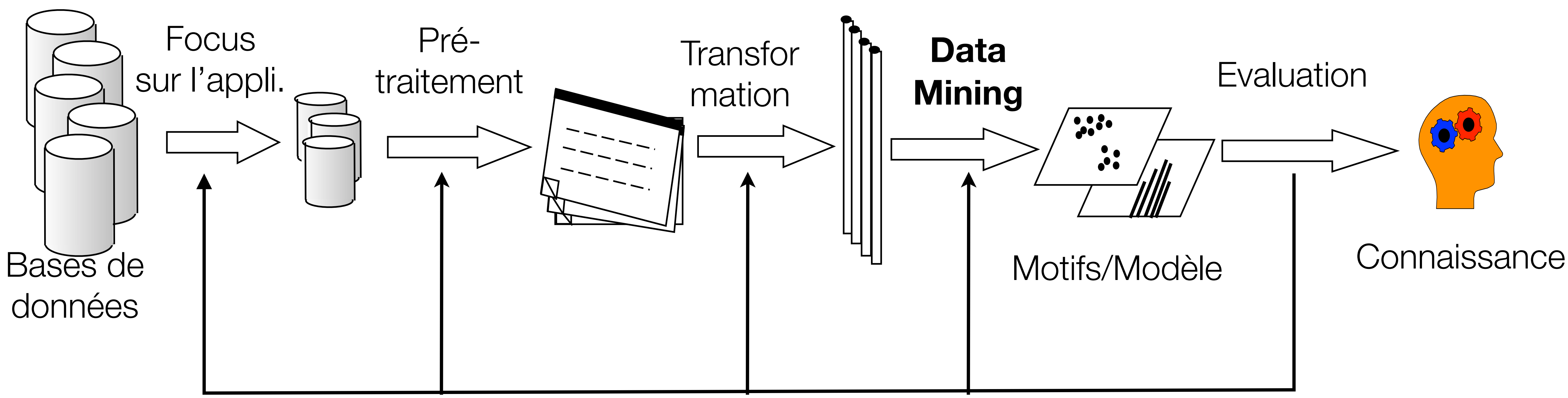
Evolution des sciences

- Avant 1600 : science empirique
- 1600-1950 : science théorique
- Années 50 - Années 90 : «Computational science»
  - Depuis plus de 50 ans, beaucoup de disciplines se sont développées sur une 3ème branche - le calcul - comme en physique, ...
  - Simulation : trouver des modèles proches de la réalité
- 1990 - Aujourd'hui : «data science»
  - Données omniprésentes (nouveaux instruments, simulations)
  - capacité à gérer et stocker des volumes gigantesques.
  - Internet
  - **La fouille de données est devenu un challenge majeur !!!**

Knowlege Discovery from Database (KDD)

- KDD (Extraction de Connaissances à partir des Données) est un processus (semi)- automatique d’extraction de connaissances à partir de bases de données où les connaissances sont :
    - valides
    - non connues a priori
    - potentiellement utiles [Fayad et al., 96]
- Remarques :
    - semi-automatique : différent d’une analyse manuelle mais souvent une interaction avec un utilisateur est nécessaire
    - non connues a priori : pas des tautologies.
    - potentiellement utiles : pour une application donnée

Le processus KDD



Processus itératif et interactif

«Focussing»

- Comprendre l’application
  - Ex. : Etablir une nouvelle tarification.
- Définir l’objectif KDD
  - Ex. : Etablir des «profils de consommateurs»
- Acquisition des données
  - Ex. : Bases de données des factures
- Gestion des données
  - Système de fichiers ou SGBD ?
- Sélection des données pertinentes
  - Ex. : considérer les 100 000 clients les plus importants et tous leurs appels sur l’année 2009

Exemple d’application

Pré-traitement

- Intégration des données à partir de différentes sources
    - Conversion des noms d'attributs (CNo -> CustomerNumber)
    - Utilisation de la connaissance du domaine pour détecter les doublons (e.g., utiliser les codes postaux)
- Vérifier la cohérence des données :
    - des contraintes spécifiques à l'application
    - Résolution des incohérences
- «Completion»
    - Le cas des valeurs manquantes

- **Le pré-traitement des données est souvent la tâche la plus coûteuse dans le processus KDD!**

Pré-traitement

- Entrepôts de données
  - persistant
  - Intégration de données issues de plusieurs sources
  - Dans un but d'analyse ou de prise de décision

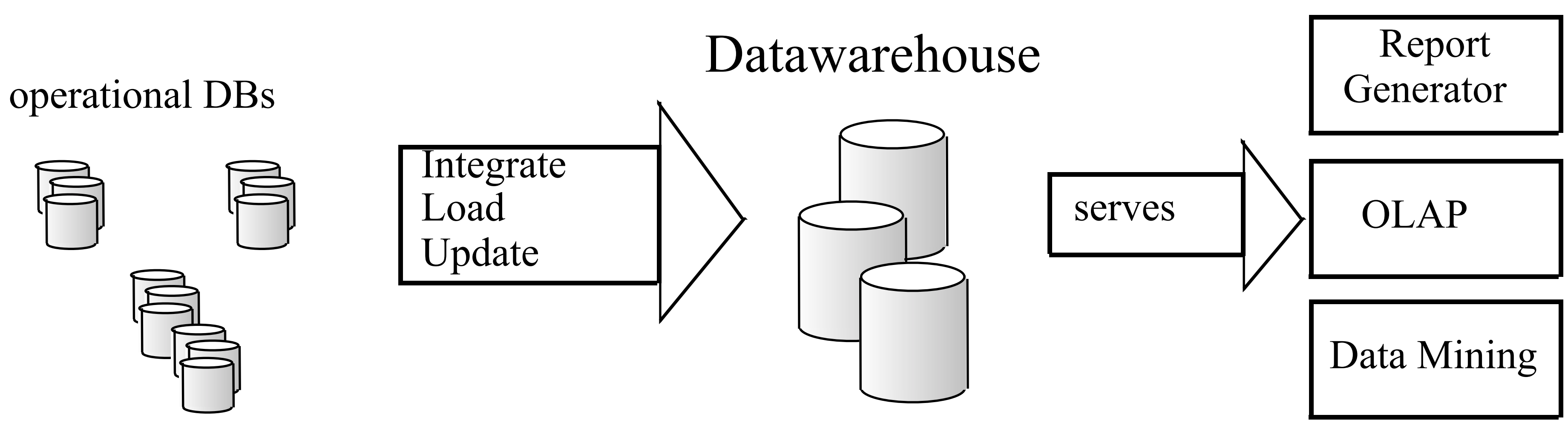


figure provenant de M. Ester

Transformation

- Discrétisation des attributs numériques
  - Indépendamment de la tâche de fouille de données
    - Ex. : partitionner le domaine des attributs en des intervalles de même longueur.
  - Spécifique de la tâche de fouille de données
    - Partitionner en des intervalles qui maximisent le gain d'information par rapport à la classe
  - Génération d'attributs dérivés :
    - Agrégation d'un ensembles d'attributs
        - Ex. : à partir d'appels
          - nb minutes par jour, semaine, appels locaux ...
    - Combinaison d'attributs :
      - Ex. : variation de revenu (revenu 2009 - revenu 2008)

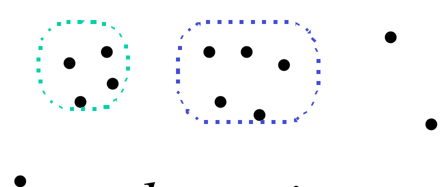
Transformation

- Sélection des attributs
  - manuellement : Si les connaissances du domaine sont disponibles pour les attributs.
  - de façon automatique :
- Trop d'attributs -> des répercussions sur l'étape de fouille de données
- Choix des attributs primordial :
  - Ex. : glace à la fraise

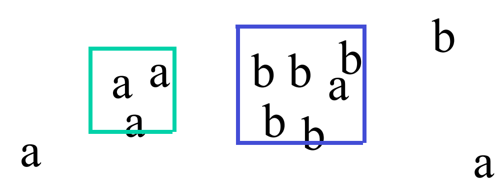
Data Mining

- Définition [Fayad et al. 96]
  - La fouille de données est l'application d'algorithmes efficaces qui identifient les motifs contenus dans une base de données

- Les différentes tâches de fouille :



clustering



classification

A and B → C

association rules

- Autres tâches : regression, détection d'outlier, etc.

Data Mining

- Applications
  - Clustering
    - Segmentation, structuration d'un ensemble de documents «web», déterminer des familles de protéines et des «super-familles», découvertes de communautés
  - Classification :
    - prédiction de la fonction d'une protéine, accorder un crédit, interpréter des images en astronomie, etc.
  - Règles d'association :
    - mise en rayon, promotion, améliorer la structure d'un site web ...

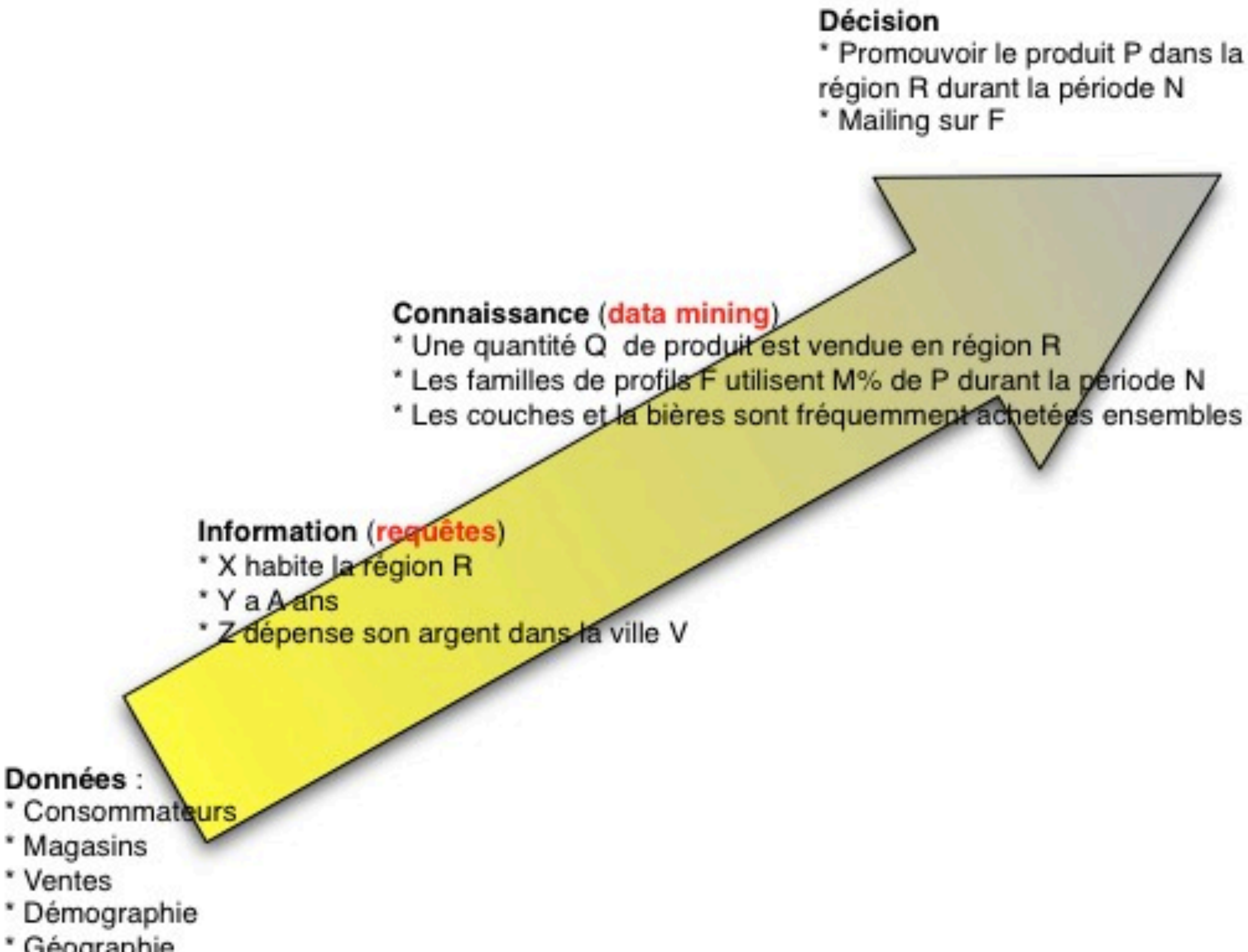
Evaluation

- Présentation des motifs découverts avec une visualisation appropriée
- Evaluation des motifs par l'utilisateur
- Si l'évaluation n'est pas satisfaisante, alors relancer la fouille avec :
  - des paramètres différents
  - d'autres méthodes
  - d'autres données
- Si l'évaluation est positive :
  - Intégrer les connaissances découvertes dans une base de connaissance
  - Utiliser ces connaissances dans les futures processus KDD

Evaluation

- Intérêt des motifs découverts :
    - motifs déjà connus ?
    - motifs surprenants ?
    - motifs pertinents par rapport à l’application ?
- Pouvoir prédictif
    - Quel est la précision du motif ?
    - Dans combien de cas se produit il ?
    - Peut-il se généraliser à d’autres cas non couverts ?

Données, information, connaissance

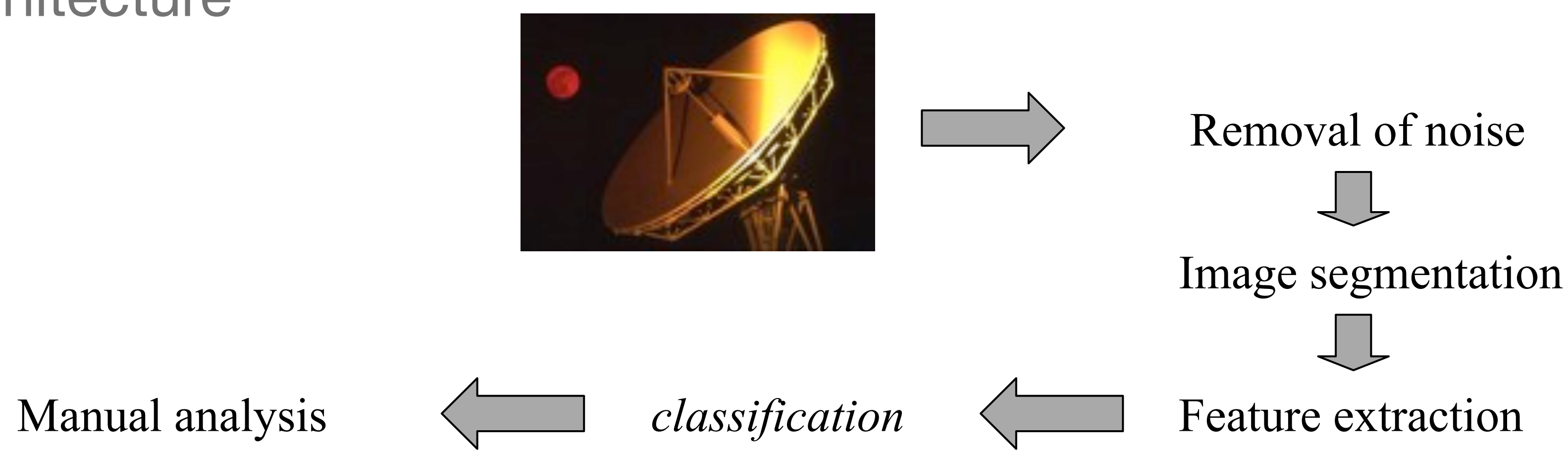


Data Mining ou non ?

- Recherche le salaire l’employé Dupont
- Interroger un moteur de recherche pour avoir des informations sur le data mining
- Regrouper un ensemble de documents retourner par un moteur de recherche en fonction de leur contenu
- Les personnes qui réalise l’action A réalise dans le mois qui suit l’action B

Applications KDD : Astronomie

- SKICAT System [Fayad et al 1996]
  - Architecture



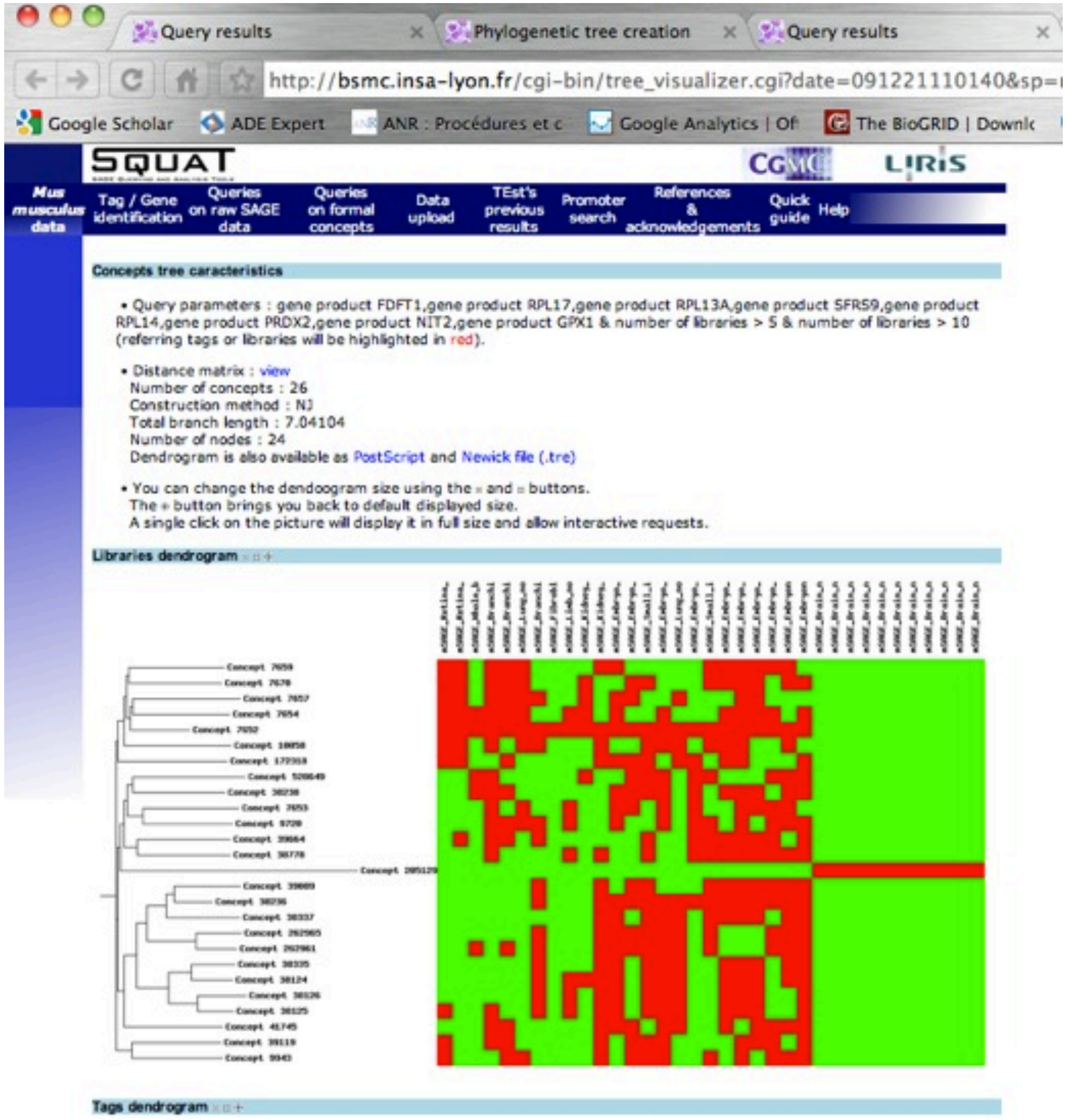
  - Méthode de classification : arbre de décision
  - Evaluation :
    - beaucoup plus rapide qu’une classification manuelle
    - Classifie aussi des objets célestes très petits

Applications KDD : Marketing

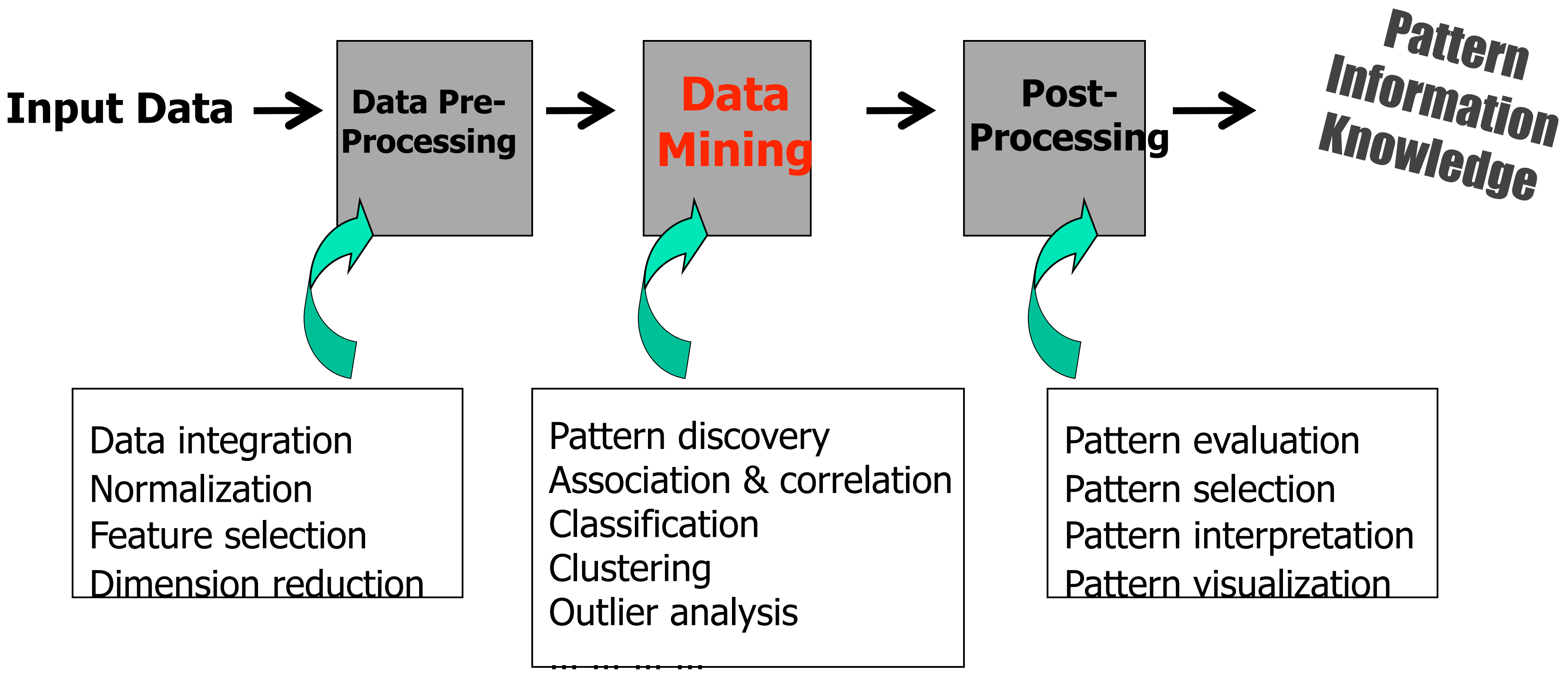
- Customer segmentation [Piatetsky-Shapiro et al 2000]
- But : partitionner les consommateurs par rapport à leurs achats
- Motivation :
  - product packages
  - établir une nouvelle politique tarifaire

Applications KDD : Commerce électronique

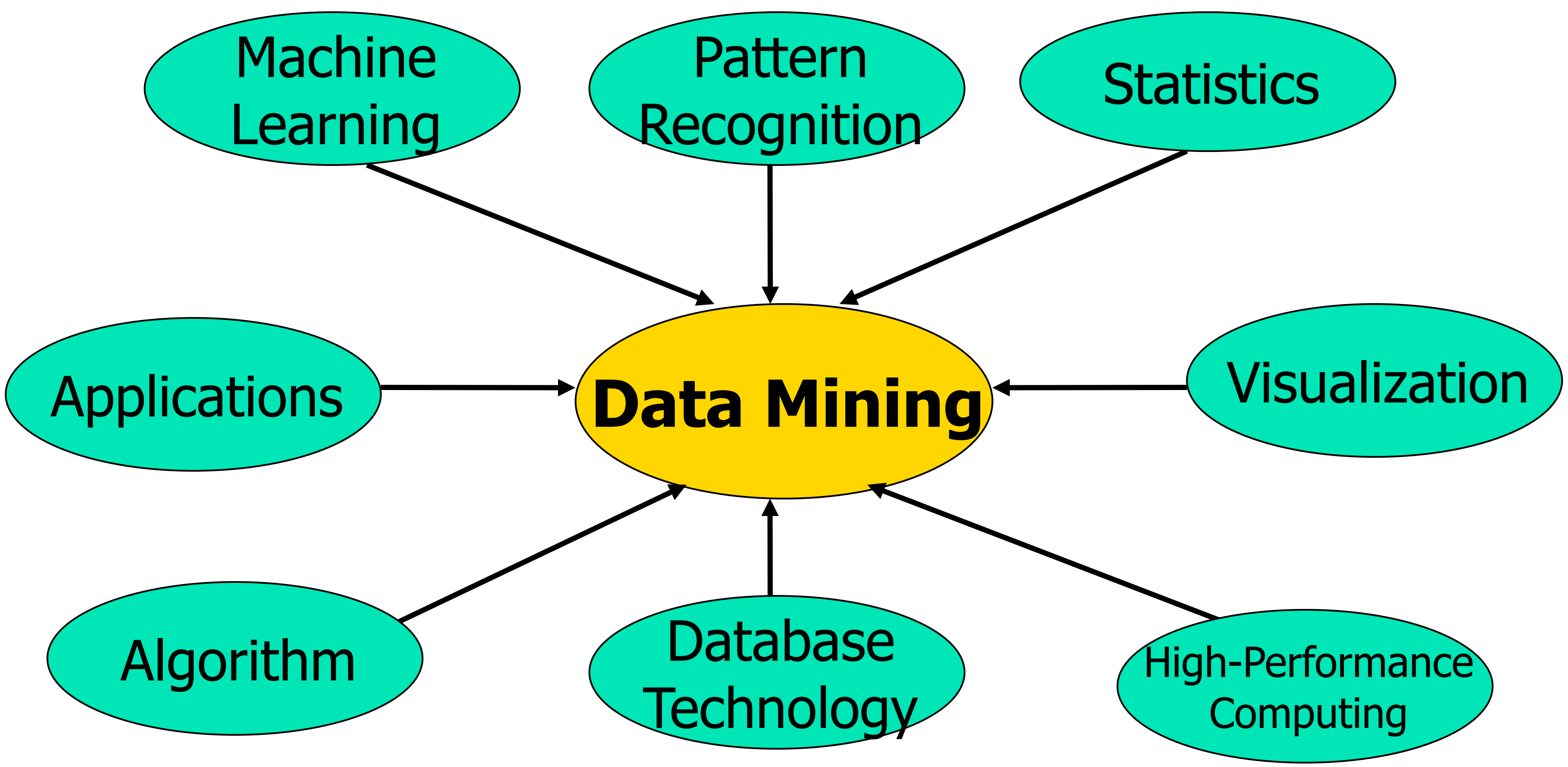
Applications KDD : puces ADN



Le data mining au centre du processus KDD



Data mining : à la confluence de nombreuses disciplines



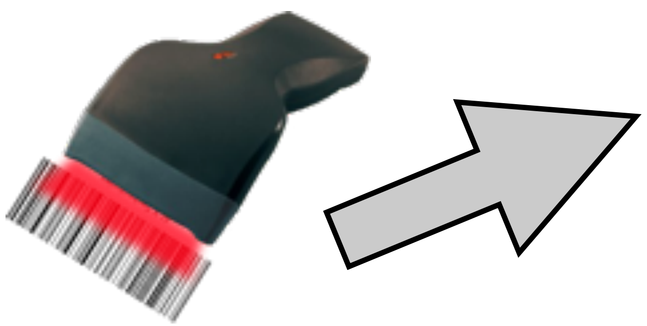
Plan du cours

- Fouille de motifs (MP)
  - Règle d’association, algorithme Apriori
  - Fouille de séquences
  - Fouille sous contraintes
  - Autres types de motifs et données
- Clustering (MP)
- Apprentissage supervisé (AA)

Motifs ensemblistes et règles d'association

Introduction

Motivations : chercher des régularités dans les données

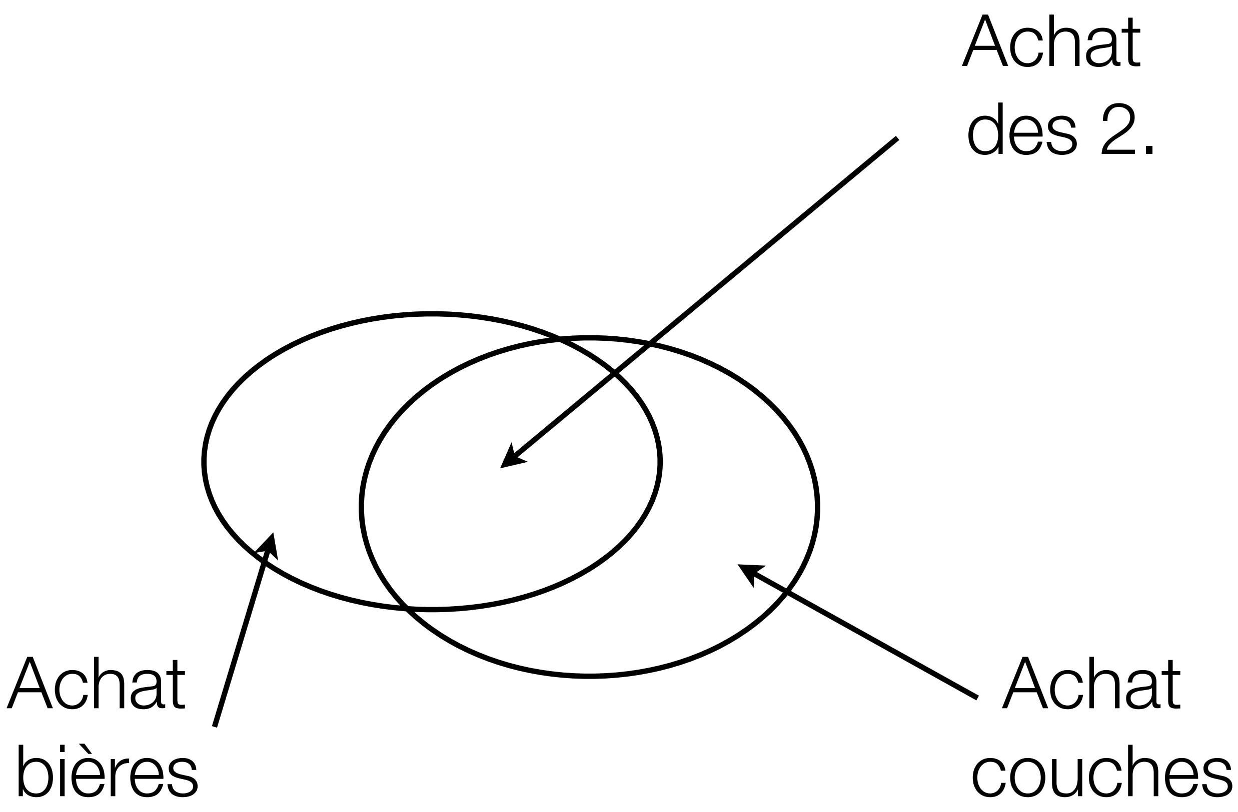


Base de données de transactions
beurre, lait, vin
oeufs, farine, coca
....
couches, bières, lait

- Analyse du «panier de la ménagère»
- Quels sont les produits qui sont fréquemment achetés ensembles ?
- Applications : rayonnage, mailing, cross marketing ...

Règles d'association

- Forme :
  - Corps -> Tête [support, confiance]
- Exemples :
  - couches -> bières [0.5, 0.7]
  - 98% des personnes qui achètent des pneus, prennent l'option montage.



Les règles d'association (plus formellement)

- Soit $\boldsymbol{I} = \{i_1, i_2, \ldots, i_n\}$ un ensemble de littéraux appelés **items.**
- Un itemset **X** : un ensemble d'items $\mathbf{X} \subseteq \boldsymbol{I}$
- Une base de données *D* consiste en un ensemble de transactions $\mathbf{T_i}$ t.q. $\mathbf{T_i} \subseteq \mathbf{I}$
- On dit que T contient X si $X \subseteq T$
- Les items d'une transaction ou d'un itemset sont triés suivant un ordre lexigographique
- Longueur d'un itemset = nombre d'items qu'il contient
- k-itemset : itemset de longueur k

Définitions

- **Support absolu** d'un itemset X dans *D* : nombre de transactions qui contiennent X
- **Support relatif** de X dans D : pourcentage de transactions de D qui contiennent X
- Itemset **fréquent** X dans D : itemset X avec un support ≥ **minsup**
- **Règle d'association : règle de la forme X -> Y avec**
  - $\mathbf{X} \subseteq \mathbf{I}$,
  - $\mathbf{Y} \subseteq \mathbf{I}$,
  - $\mathbf{X} \cap \mathbf{Y} = \emptyset$

Définitions

- Support d'une règle d'association X->Y dans D :
  - support de X U Y dans D

$$s = \frac{|\{T \in D \mid (X \cup Y) \subseteq T\}|}{|D|}$$

- Confiance d'une règle d'association X->Y dans D :
  - pourcentage de transactions contenant Y qui contiennent aussi X

$$c = \frac{|\{T \in D \mid X \cup Y \subseteq T\}|}{|\{T \in D \mid X \subseteq T\}|}$$

- Objectif : Etant donné un seuil de support minsup et un seuil de confiance mincong, découvrir toutes les règles d'association qui ont un support ≥ minsup et une confiance ≥ minconf

Exemple

TransactionID	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

minsup = 50%,
minconf = 50%

Support

(A): 75%, (B), (C): 50%, (D), (E), (F): 25%,

(A, C): 50%, (A, B), (A, D), (B, C), (B, E), (B, F), (E, F): 25%

Règles d'association

$A \Rightarrow C$ (support = 50%, confidence = 66.6%)

$C \Rightarrow A$ (support = 50%, confidence = 100%)

Découverte des règles d’association

- Deux étapes :
  - Découvrir tous les itemsets fréquents dans D
  - Générer les règles d’association à partir des itemsets fréquents :
    - Pour tous les itemsets fréquents X :
      - Pour tous les A ⊂ X : (qui satisfait la contrainte de support)
        - Générer la règle $A \Rightarrow (X - A)$ (qui satisfait la contrainte de support)
        - Verifier la confiance de la règle

Extraction des motifs fréquents (approche naïve)

- Générer tous les itemsets possibles, puis calculer leur support dans la base de données
- Problèmes :
  - Comment garder en mémoire un nombre important d'itemsets ?
    - 100 items => 2^{100} -1 itemsets possibles !!!!
  - Comment calculer le support d'un nombre important d'itemsets dans une grande base de données (100 million de transactions) ?

Extraction des motifs fréquents (approche naïve)

- Générer tous les itemsets possibles, puis calculer leur support dans la base de données

- Problèmes :

  - Comment garder en mémoire un nombre important …

    - 100 items => 2^{100} -1 itemsets possibles !!!!

  - Comment calculer le support d'un nombre importa… d'ite… une grande base de données (100 million de transactions) ?

Approche naïve non viable !!!

Extraction des motifs fréquents

- Propriété d’ anti-monotonie du support :
  - Tous les sous ensembles d’un itemset fréquent sont fréquents
  - Si un itemset X n’est pas fréquent alors il n’existe pas d’itemset Y t.q X C Y qui soit fréquent

Méthode

- Trouver les 1-itemsets fréquents, puis trouver les 2-itemsets fréquents ....
- Pour trouver les k+1-itemsets fréquents :
  - Seulement considérer les k+1-itemsets t.q. :
    - tous les k-sous-ensembles sont fréquents.
- Calcul du support :
  - Une passe sur la base de données pour compter le support de tous les itemsets pertinents.

Algorithme Apriori

C_k: set of *candidate* item sets of length k
L_k: set of all *frequent* item sets of length k

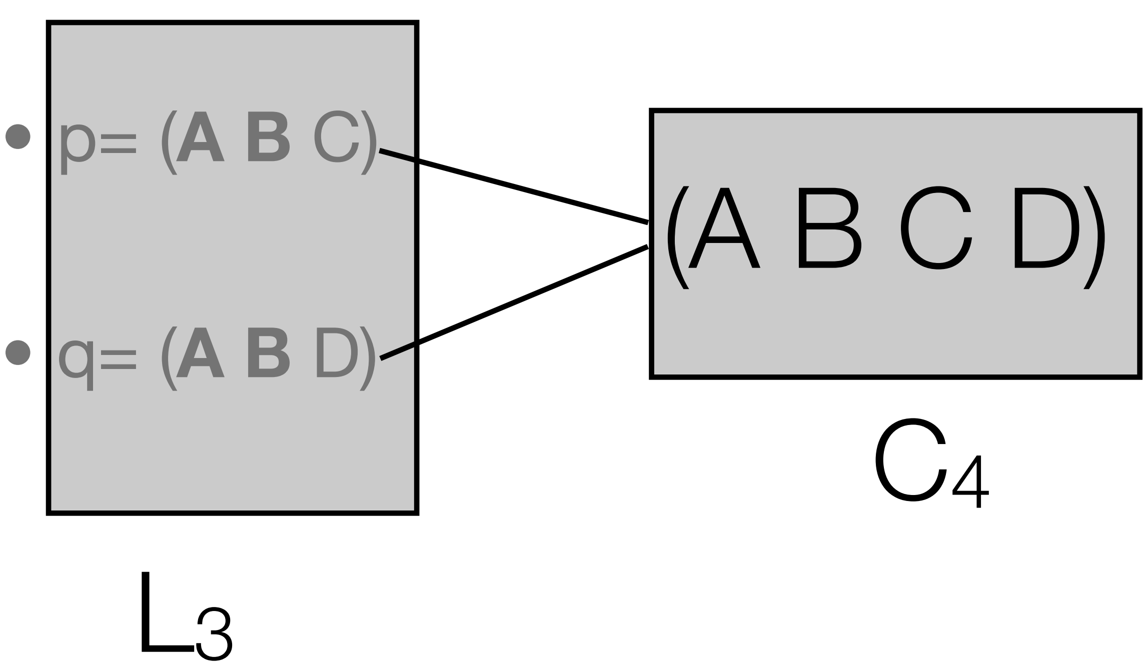
```
Apriori(D, minsup)
L_1 := {frequent 1-item sets in D};
k := 2;
while L_{k-1} ≠ ∅ do
   C_k := AprioriCandidateGeneration(L_{k-1});
   for each transaction T ∈ D do
      CT := subset(C_k, T); // all candidates from C_k, that are
                            // contained in transaction T;
      for each candidate c ∈ CT do c.count++;
   L_k := {c ∈ C_k | (c.count / |D|) ≥ minsup};
   k++;
return ∪_k L_k;
```

Génération de candidats

- Propriétés de l’ensemble C_k des k-itemsets candidats
  - Sur-ensemble de L_k
  - Significativement plus petit que tous k-itemsets possibles de I

Génération de candidats : la jointure

- Etape 1:
  - p et q : (k-1) itemsets fréquents
  - p et q sont joints si ils ont leurs (k-2) premières valeurs identiques
- Ex. :



Génération de candidats : élagage

- Etape 2 : l'élagage
  - Supprimer tous les éléments de C_k qui ont un (k-1) sous-ensemble qui n'appartient pas à L_{k-1}.
- Ex. : L_3 = {(1 2 3), (1 2 4), (1 3 4), (1 3 5), (2 3 4)}

Jointure : C_4 = {(1 2 3 4), (1 3 4 5)}

Elagage: suppression de (1 3 4 5) car (3 4 5) n'appartient pas à L_3

Au final : C_4 = {(1 2 3 4)}

Exemple d’une extraction complète

Au tableau avec un treillis et tout l’espace de recherche.

Exemple d’une extraction complète

Au tableau avec un treillis et tout l’espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Exemple d'une extraction complète

Au tableau avec un treillis et tout l'espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

Exemple d'une extraction complète

Au tableau avec un treillis et tout l'espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

→

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

Exemple d'une extraction complète

Au tableau avec un treillis et tout l'espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

→

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Exemple d'une extraction complète

Au tableau avec un treillis et tout l'espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

Exemple d'une extraction complète

Au tableau avec un treillis et tout l'espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Exemple d'une extraction complète

Au tableau avec un treillis et tout l'espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Exemple d'une extraction complète

Au tableau avec un treillis et tout l'espace de recherche.

minsup = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Espace de recherche (suite au tableau)

12345

1234 1235 1245 2345

123 124 125 234 235 345

12 13 14 15 23 24 25 34 35 45

1 2 3 4 5

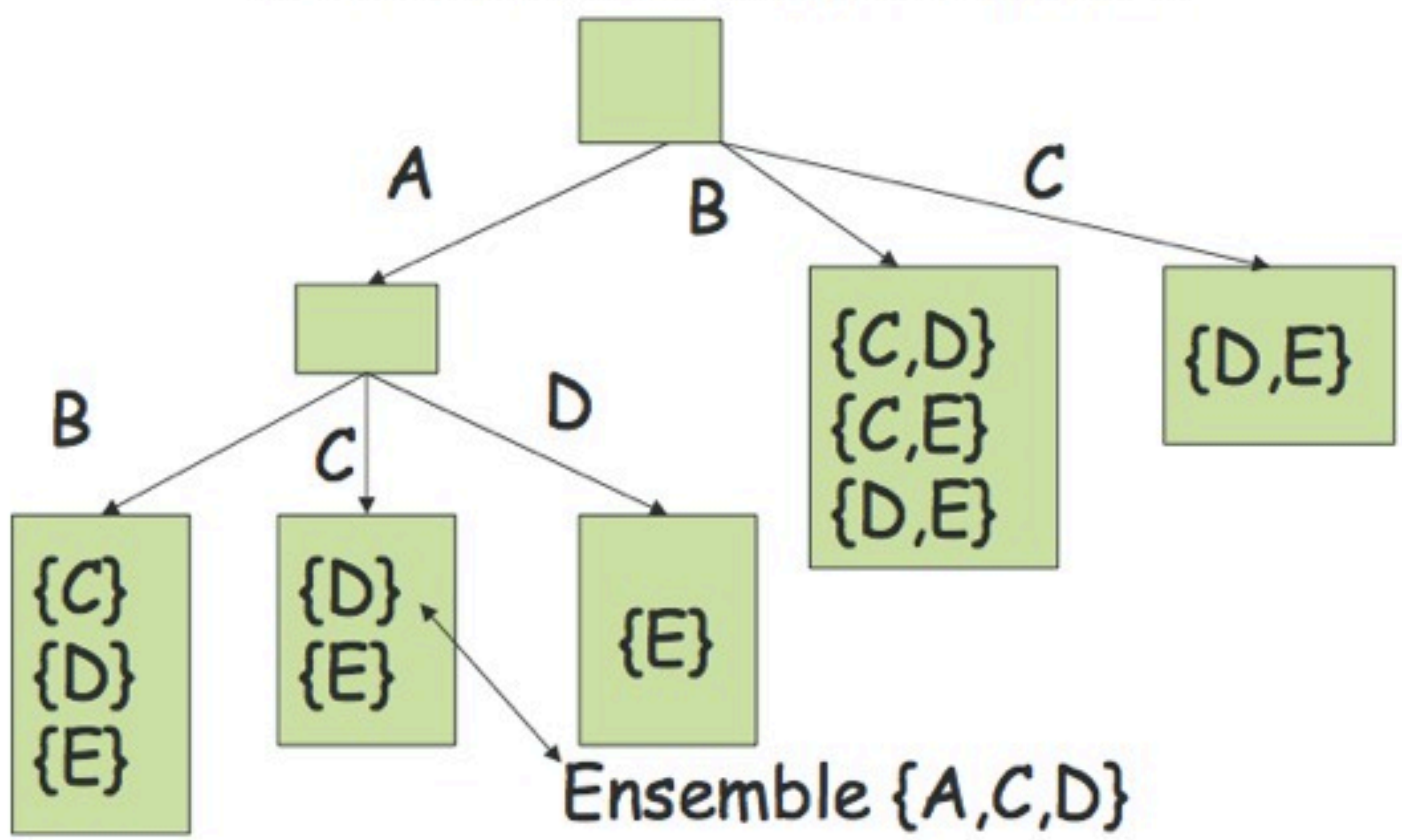
∅

Calcul du support efficace

- `Subset(`C_k`, `T`)`

  Tous les candidats de C_k qui sont contenus dans la transaction T
- Problèmes :
  - Très grand nombres d'itemsets candidats
  - Une transaction peut contenir un très grand nombre de candidats
- Structure de données Hash tree pour stocker C_k

structure de tous les 3-candidats possibles pour 5 items



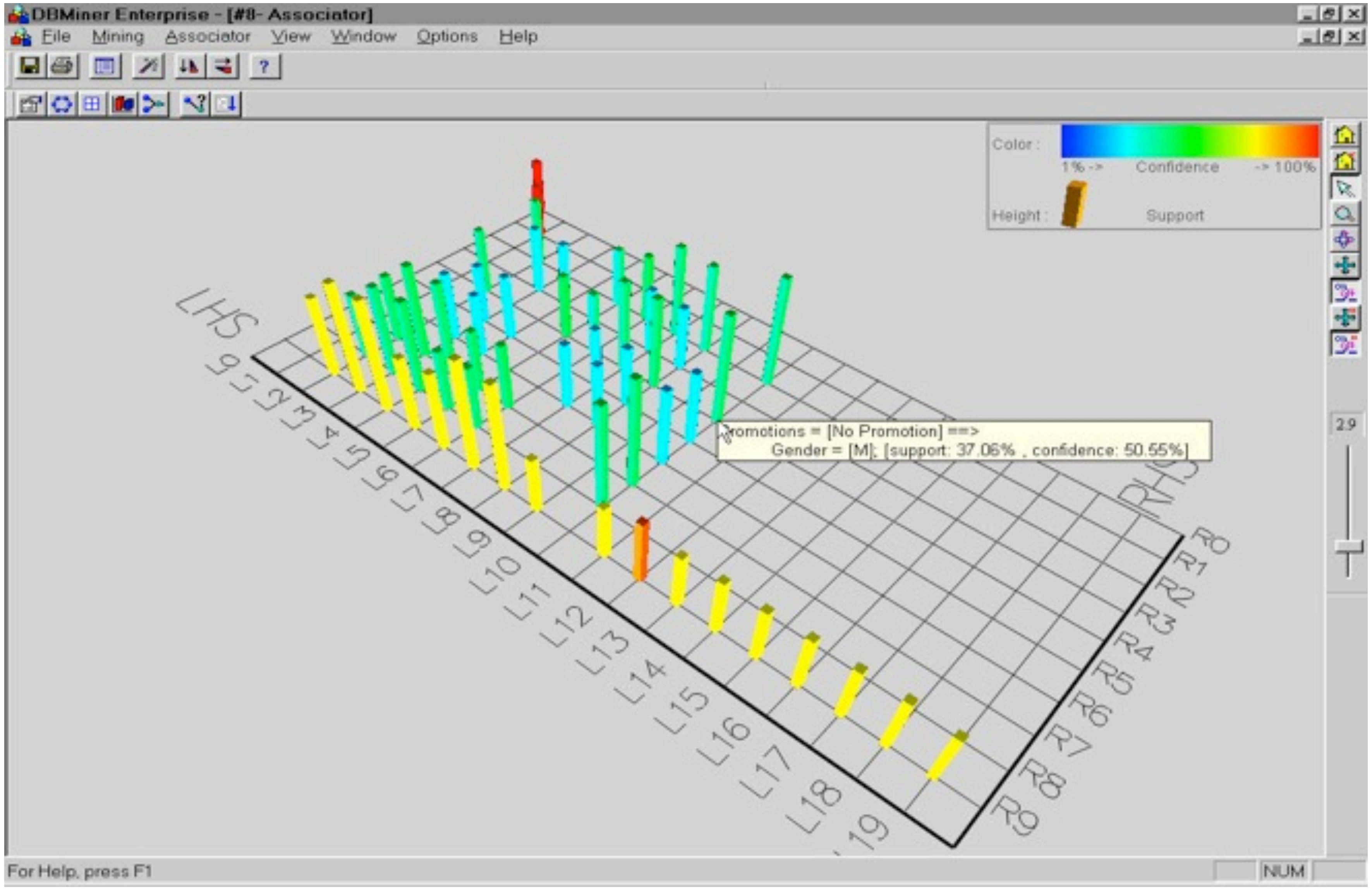
Génération des règles à partir des itemsets

- Pseudo-code :
  - Pour chaque itemset fréquent l :
    - Générer tous les sous-ensembles non vides X de l
    - Pour chaque X de l :
      - Si support(l)/support(X) ≥ minconf alors
        - produire la règle X => (l-X)

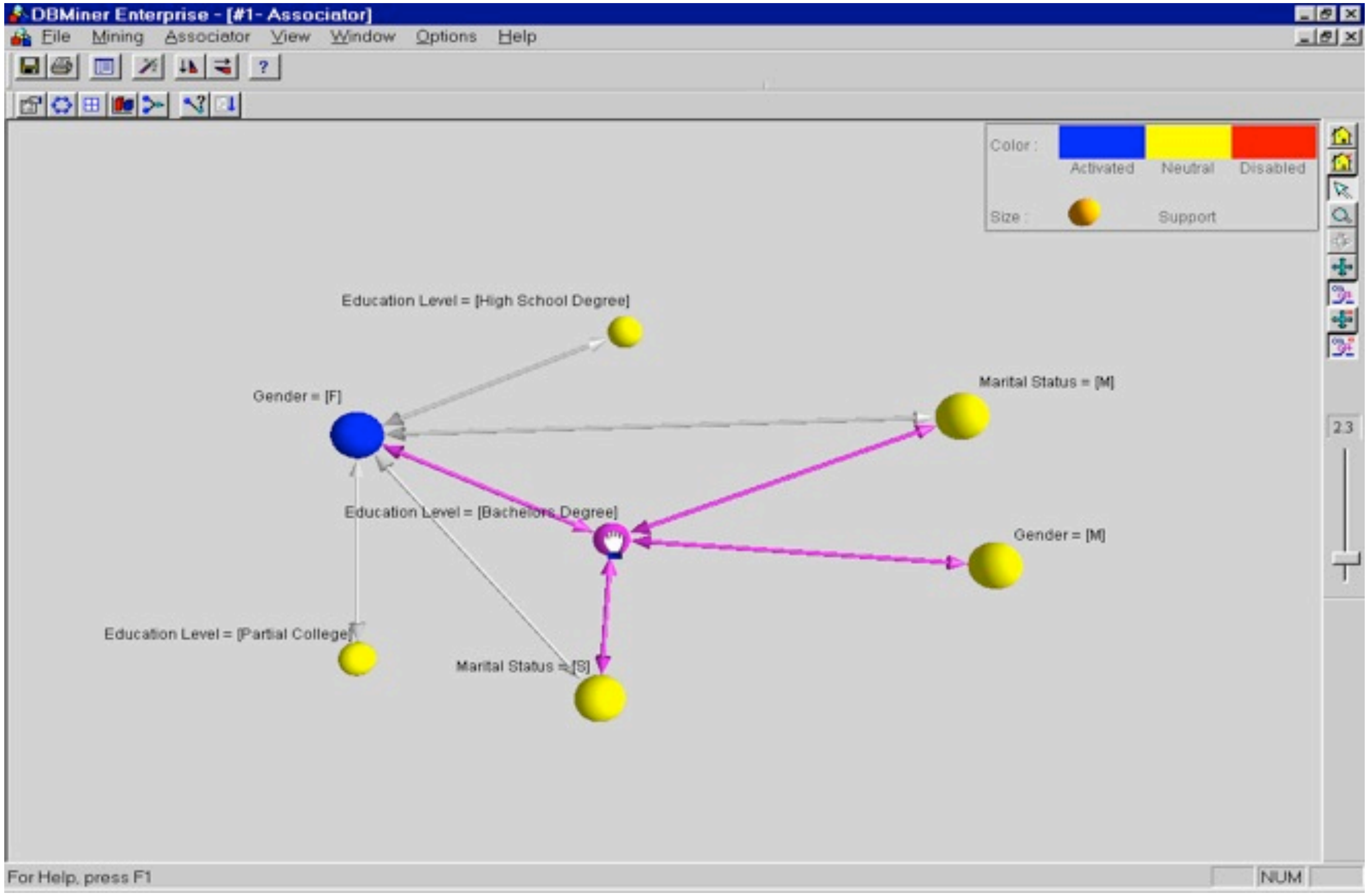
Exercice

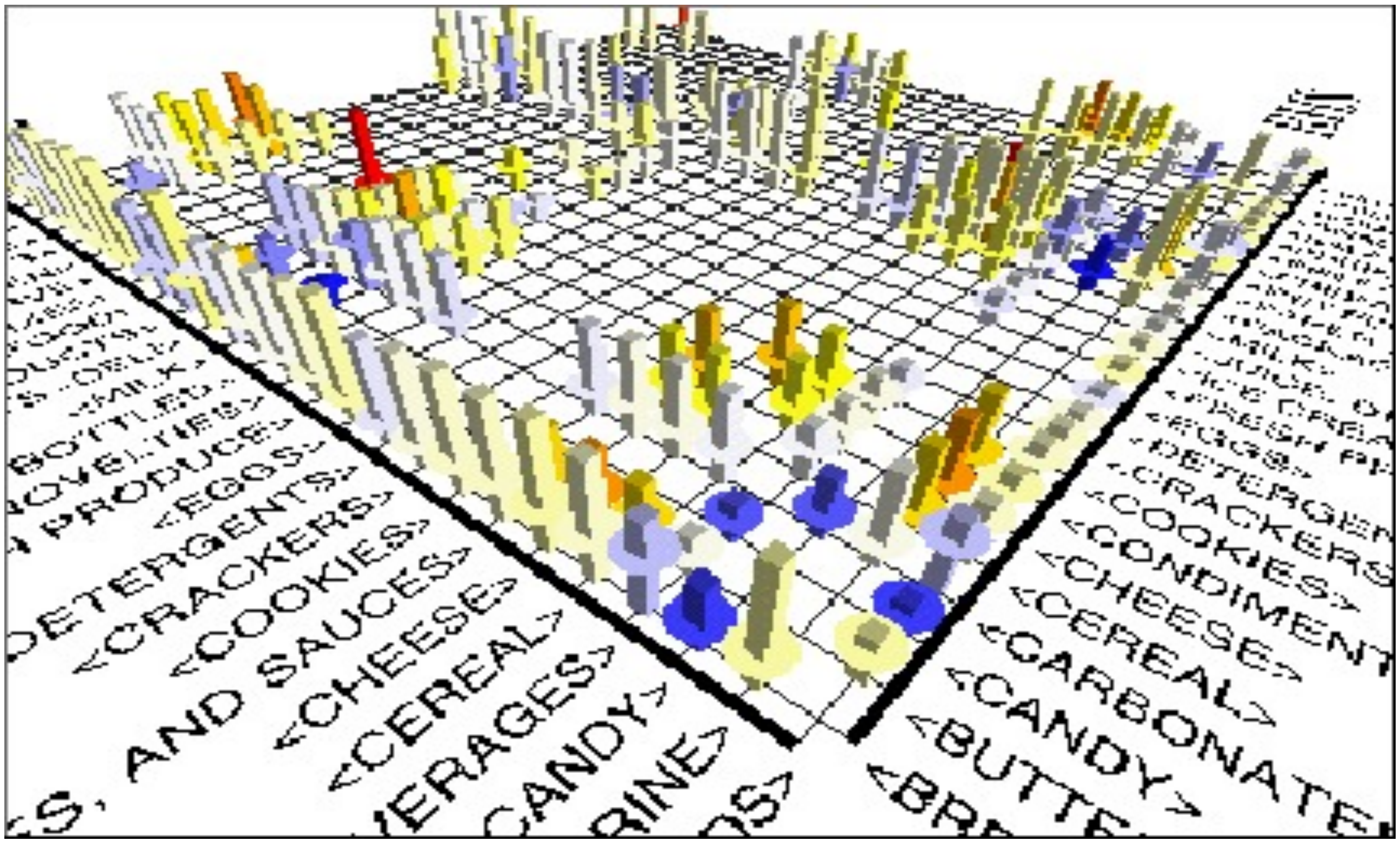
- Voir feuille de «TD»
  - Extraction de motifs fréquents
    - Propriétés d’Apriori
    - Bordures
  - Règles d’association
    - Propriétés
    - ...

Visualisation des règles : dans un plan



Graphe de règles



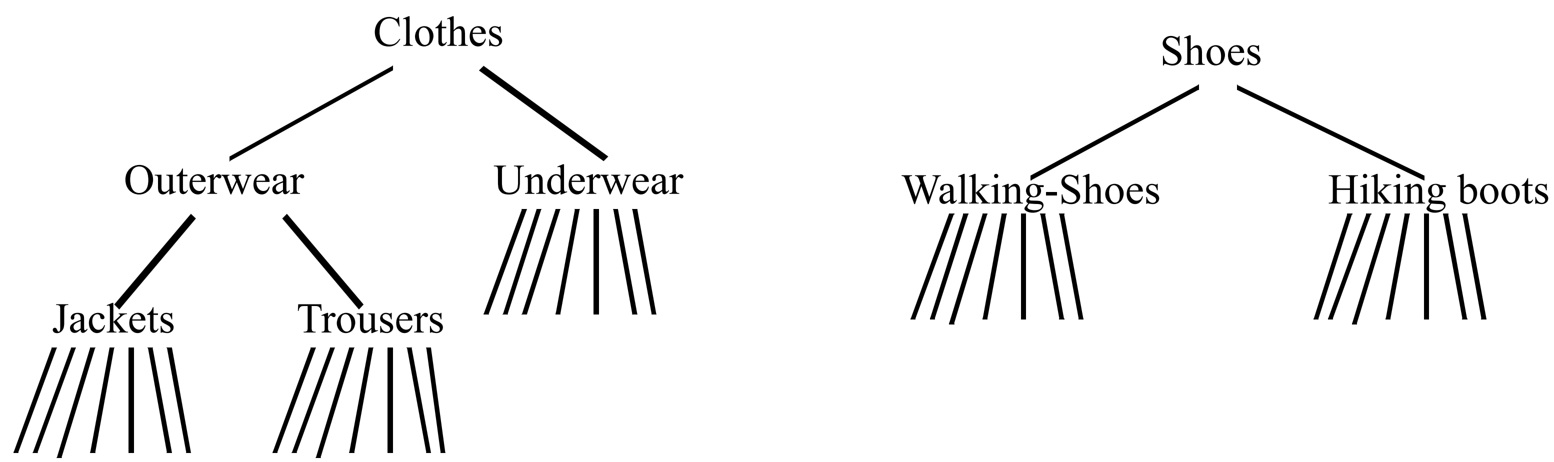

<EGGS>
<DETERGENTS>
<CRACKERS>
<COOKIES>
<CHEESE>
<CEREAL>
<CANDY>
<COOKIES>
<CHEESE>
<CEREAL>
<CANDY>

Différentes mesures

symbol	measure	range	formula
ϕ	ϕ-coefficient	-1…1	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	-1 … 1	$\frac{P(A,B)P(\overline{A},\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B})+P(A,\overline{B})P(\overline{A},B)}$
Y	Yule's Y	-1 … 1	$\frac{\sqrt{P(A,B)P(\overline{A},\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}}$
k	Cohen's	-1 … 1	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
PS	Piatetsky-Shapiro's	-0.25 … 0.25	$P(A,B)-P(A)P(B)$
F	Certainty factor	-1 … 1	$\max(\frac{P(B\mid A)-P(B)}{1-P(B)}, \frac{P(A\mid B)-P(A)}{1-P(A)})$
AV	added value	-0.5 … 1	$\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$
K	Klosgen's Q	-0.33 … 0.38	$\sqrt{P(A,B)}\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$
g	Goodman-kruskal's	0 … 1	$\frac{\Sigma_j \max_k P(A_j,B_k)+\Sigma_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-max_k P(B_k)}$
M	Mutual Information	0 … 1	$\frac{\Sigma_i \Sigma_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\Sigma_i P(A_i)\log P(A_i)\log P(A_i), -\Sigma_i P(B_i)\log P(B_i)\log P(B_i))}$
J	J-Measure	0 … 1	$\max(P(A,B)\log(\frac{P(B\mid A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\mid A)}{P(\overline{B})}))$ $P(A,B)\log(\frac{P(A\mid B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\mid B)}{P(\overline{A})})$
G	Gini index	0 … 1	$\max(P(A)[P(B\mid A)^2+P(\overline{B}\mid A)^2]+P(\overline{A}[P(B\mid\overline{A})^2+P(\overline{B}\mid\overline{A})^2]-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\mid B)^2+P(\overline{A}\mid B)^2]+P(\overline{B}[P(A\mid\overline{B})^2+P(\overline{A}\mid\overline{B})^2]-P(A)^2-P(\overline{A})^2)$
s	support	0 … 1	$P(A,B)$
c	confidence	0 … 1	$max(P(B\mid A), P(A\mid B))$
L	Laplace	0 … 1	$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$
IS	Cosine	0 … 1	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
γ	coherence(Jaccard)	0 … 1	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
α	all_confidence	0 … 1	$\frac{P(A,B)}{\max(P(A),P(B))}$
o	odds ratio	0 … ∞	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(\overline{A},B)P(A,\overline{B})}$
V	Conviction	0.5 … ∞	$\max(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})})$
λ	lift	0 … ∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 … ∞	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
χ^2	χ^2	0 … ∞	$\Sigma_i \frac{(P(A_i)-E_i)^2}{E_i}$

Règles d'association multi-niveaux

- Dans de nombreuses applications, des **hiérarchies** sont associées aux items.



- Recherche des R.A sur les feuilles : support risque d'être trop bas.

- Recherche des R.A aux niveaux supérieurs : motifs risquent de représenter des informations déjà connues.

- **Il faut donc chercher des R.A sur tous les niveaux.**

Exemple

Anorak ⇒ Hiking boots
Windcheater ⇒ Hiking boots
Support < minsup

Jacket ⇒ Hiking boots Support > minsup

Propriétés :

- Support (Jacket ⇒ Hiking boots) peut être différent de support(Anorak ⇒ Hiking boots) + support(Windcheater ⇒ Hiking boots)
- Si support(Jacket ⇒ Hiking boots) > minsup, alors support(Outerwear ⇒ Hiking boots) > minsup

Règles d'association multi-niveaux [Agrawal et Srikant, 1995]

- $I = \{i_1, ..., i_m\}$ un ensemble de littéraux (items)
- H un graphe orienté sans cycle (DAG) sur I
- Dans H, une arête de i vers j si :
  - i est une généralisation de j,
  - i est le père (prédécesseur direct) de j, j est un fils ou successeur direct
- Plus généralement, $\overline{X}$ prédécesseur de X s'il existe un chemin entre X et X dans H

• itemset $\overline{Z}$ est un prédécesseur d'un itemset Z : pour tout item de Z, on a au moins un pred dans $\overline{Z}$

- D est un ensemble de transactions T où T $\subseteq$ I
- Généralement, une transaction de D contient uniquement des items qui sont **feuilles** dans H
- Une transaction T supporte un item i si :
  - i $\in$ T ou i est un prédécesseur d'un item j de T
- T supporte un itemset X si tous les items de I sont supportés par T
  - ex. : (coca, chips) supporte (soda, chips)
- Support(X,D) = % de transaction de D supportant X

- Règle d'association multi-niveaux :
  - $X \Rightarrow Y$ où $X \subseteq I$, $Y \subseteq I$, $X \cap Y = \varnothing$ et aucun item de Y n'est prédécesseur d'un item de X dans H;
- Support($X \Rightarrow Y$,D) = support(X U Y, D)
- Confiance

Exemple

TransactionID	Items
1	Anorak
2	Windcheater, Hiking boot
3	Anorak, Hiking boot
4	Walking-shoes
5	Walking-shoes
6	Windcheater

- Support(Jackets): 4/6 = 67%
- Support(Jackets, Hiking boots): 2/6 = 33%
- Hiking-boots ⇒ Jackets : Support 33%, Confiance 100%
- Jackets ⇒ Hiking-boots : Support 33%, Confiance 50%

Déterminer les itemsets fréquents

- Idée originale (algorithme basique) :
  - Etendre la base de données avec tous les prédécesseurs des items contenus dans chaque transaction;
    - ex: T1: (coca,vin) =>T1’:(coca,soda,vin, alcool, boissons ...)
    - Ne pas insérer de duplications !
  - Ensuite, rechercher les itemsets fréquents (APriori);
- Optimisations : filtrage des prédécesseurs à ajouter, matérialisation des prédécesseurs.

- Suppression des itemsets redondants

- Soit *X* un *k*-itemset, *i* un item et *i'* un prédécesseur de *i*.
- X=(i,i',...)
- Support de *X* – {i'} = support de *X*
- *X* ne doit pas être considérer pendant la génération de candidats
- On n'a pas besoin de compter le support d'un *k*-itemset qui contient l'item *i* et son prédécesseur i'
- Algorithme *Cumulate*

Stratification

- Alternative à Apriori
- former des couches dans l'ensemble de candidats
- Propriété : si un itemset $\overline{X}$ n'est pas fréquent alors X non plus.
- Méthode :
  - Ne pas compter le support tous les k-itemsets en même temps
  - Compter en premier le support des itemsets les plus généraux et ensuite considérer le calcul des plus spécifiques si nécéssaire.

Exemple

- C_k = {(Clothes Shoes), (Outerwear Shoes), (Jackets Shoes)}
    - regarder d'abord (Clothes Shoes)
    - compter ensuite le support de (Outerwear Shoes) uniquement si nécessaire

Notations

- *Depth* d'un itemset
- (C_k^n): Set of item sets from C_k with depth n, $0 \leq n \leq$ maximal depth t

Algorithme Stratify

- Au tableau à partir d’un exemple simple.
- Représentation de l’espace de recherche («treillis déformé»)

Intérêt des règles d'association multi-niveaux

- $X \Rightarrow Y$ est un prédécesseur de $\bar{X} \Rightarrow \bar{Y}$
- Une règle d'association multi-niveaux est R-intéressante si :
  - Elle n'a pas de prédécesseur direct ou
  - Support (ou confiance) est R fois supérieur au support (confiance) attendue et que le prédécesseur direct est aussi R-intéressant.

Exemple

Item	Support
Clothes	20
Outerwear	10
Jackets	4

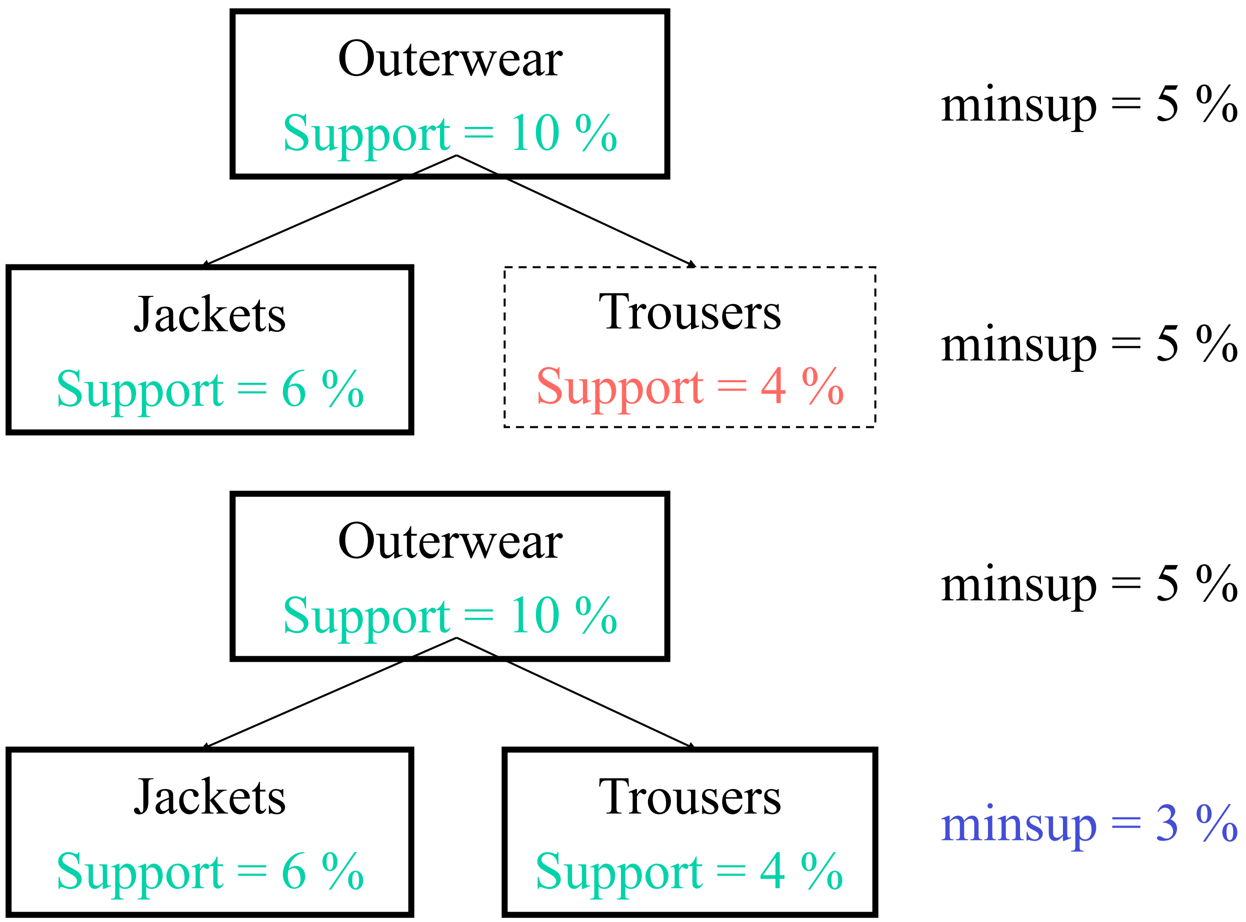
$R = 2$

Rule-No	Rule	Support	R-interesting ?
1	Clothes $\Rightarrow$ Shoes	10	yes, no predecessor
2	Outerwear $\Rightarrow$ Shoes	9	yes, support $\approx R$ * expected support (w.r.t. rule 1)
3	Jackets $\Rightarrow$ Shoes	4	no, support < R * expected support (w.r.t. rule 2)

Choix du support

Fix support

Variable support



Discussion

- Support fixe (même minsup pour tous les niveaux de H)
  - + : efficacité (suppression des successeurs non fréquents
  - - : réduit l’intérêt des règles
    - minsup trop haut : pas de règles de bas niveau
    - minsup trop bas : trop de règles de haut niveau
  - Support variable :
    - + : Intérêt : on trouve des règles aux niveaux appropriés
    - - : efficacité de l’extraction (pas de pruning des successeurs directs)

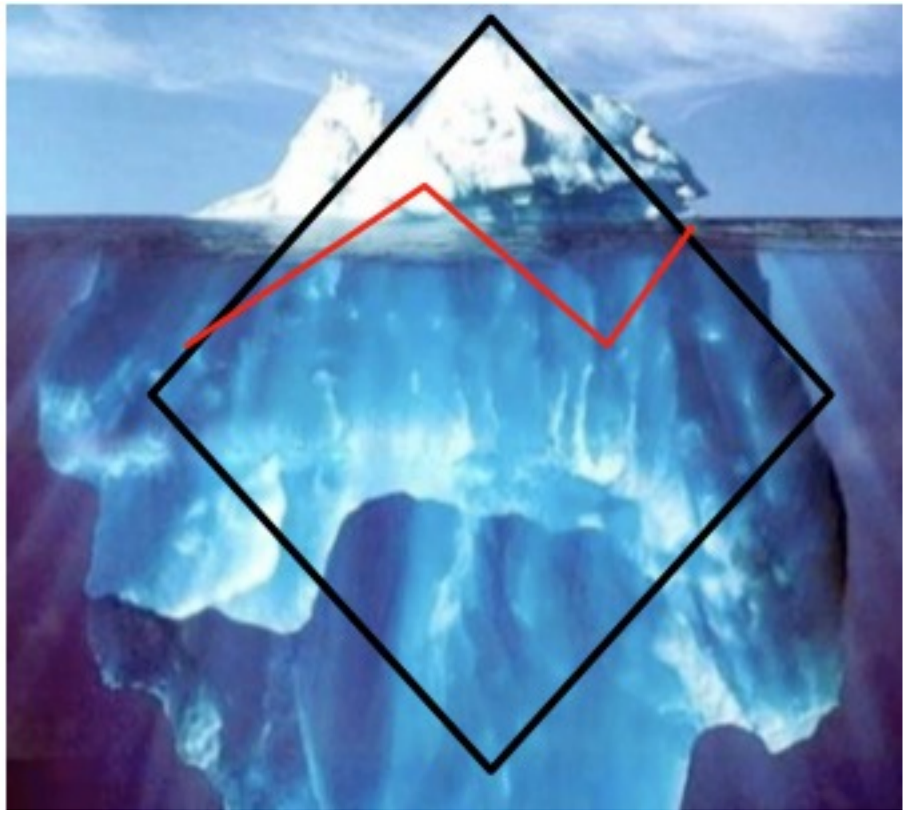
Règles multidimensionnelles

- Règle mono-dimensionnelle :
    - buys(X, “milk”) ⇒ buys(X, “bread”)
- Règle multidimensionnelle : ≥ 2 dimensions ou predicats
    - RA Inter-dimension (pas de prédicats répétés)
        - age(X,”19-25”) ∧ occupation(X,“student”) ⇒ buys(X, “coke”)
    - RA hybrides (prédicats répétés)
        - age(X,”19-25”) ∧ buys(X, “popcorn”) ⇒ buys(X, “coke”)
- Attributs catégoriels : Nombre finis de valeurs possibles, pas d’ordres parmi les valeurs — approche cube de données
- Attributs quantitatifs : Numérique, ordre implicite parmi les valeurs— discrétisation, clustering, ...

Autres applications de la recherche de motifs ensemblistes : les icebergs

Représenter que les cellules respectant une condition

SELECT A,B,C, Count(*),SUM(X)
FROM TableName
CUBE BY A,B,C
HAVING COUNT(*) $\geq$ minsup



Amélioration de l’algorithme Apriori

Rappel

Data base D

TID	Items
10	a, c, d
20	b, c, e
30	a, b, c, e
40	b, e

Min_sup=2

Scan D

1-candidates

Itemset	Sup
a	2
b	3
c	3
d	1
e	3

Freq 1-itemsets

Itemset	Sup
a	2
b	3
c	3
e	3

2-candidates

Itemset
ab
ac
ae
bc
be
ce

Scan D

Counting

Itemset	Sup
ab	1
ac	2
ae	1
bc	2
be	3
ce	2

Freq 2-itemsets

Itemset	Sup
ac	2
bc	2
be	3
ce	2

3-candidates

Itemset
bce

Scan D

Freq 3-itemsets

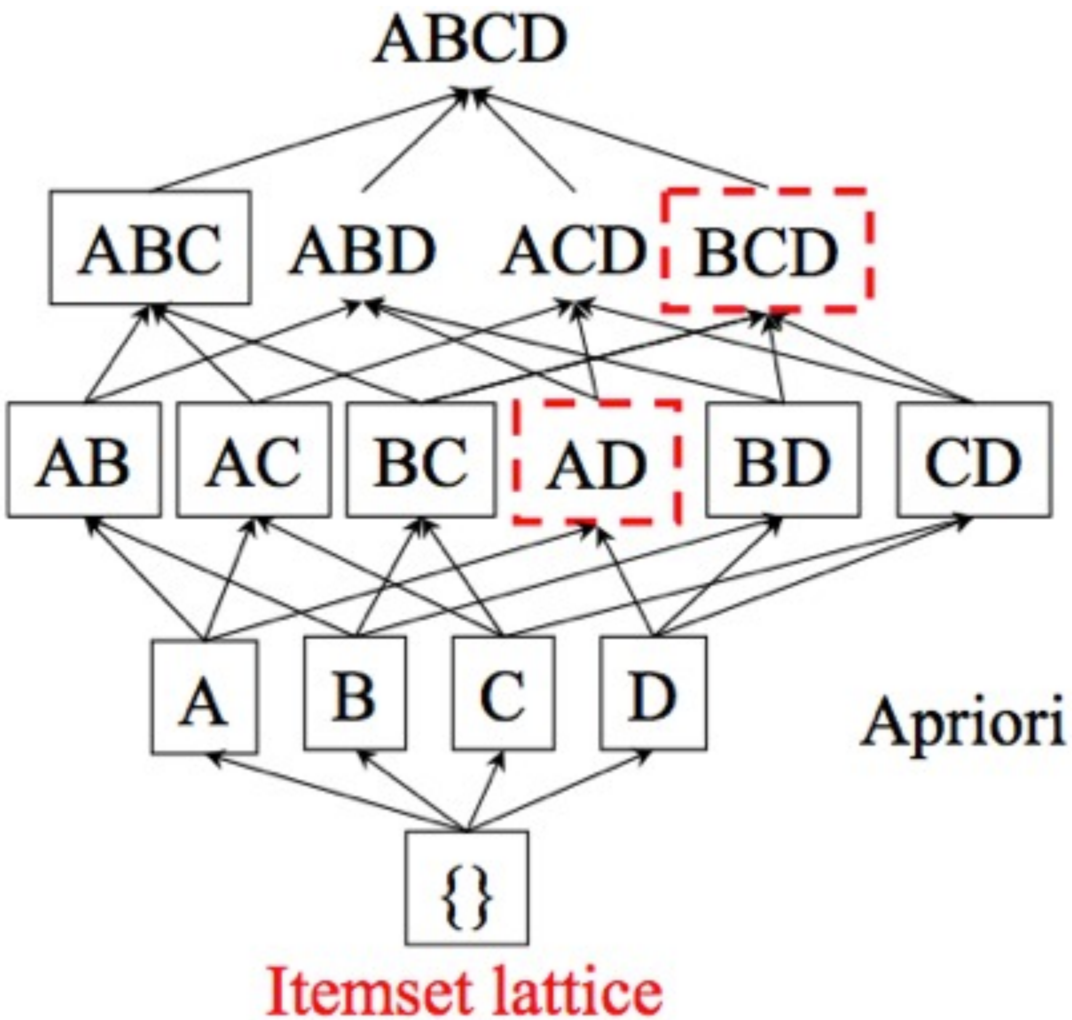
Itemset	Sup
bce	2

Les enjeux et les idées à développer

- Multiples passages sur la base de données
- Ensemble de candidats très importants
- Calcul du support d'un motif

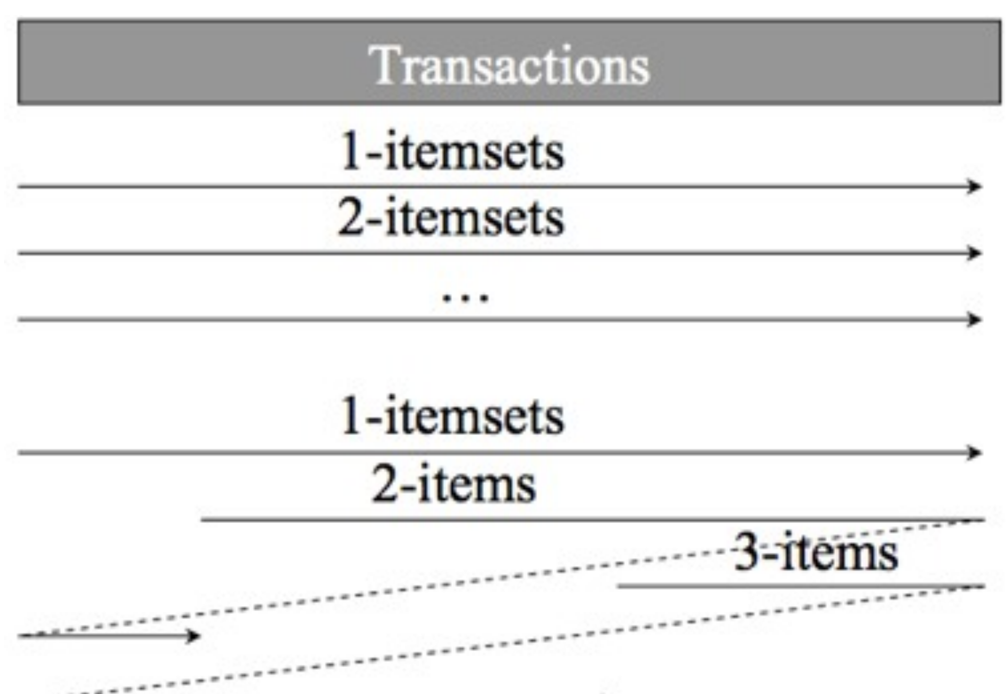
- Réduire le nombre de passages sur la base de données
- Réduire le nombre de candidats
- Faciliter le calcul du support

DIC : réduction du nombre de passage



S. Brin R. Motwani, J. Ullman, and S. Tsur, SIGMOD'97.
DIC: Dynamic Itemset Counting

- Dès que A et D sont déterminés comme fréquents, on peut commencer à calculer le support de (AD)
- Dès que les 2-sous-ensembles de (BCD) sont trouvés fréquents alors on peut commencer à chercher le support de (BCD)



DHP : Réduction du nombre de candidats

- A hashing bucket count < min_sup˽every candidate in the bucket is infrequent
  - Candidats: a, b, c, d, e
  - Hash entrées: {ab, ad, ae} {bd, be, de}
  - Large 1-itemset: a, b, d, e
  - La somme des supports de {ab, ad, ae} < min_sup
  - inutile de générer (ab)
- J. Park, M. Chen, and P. Yu, SIGMOD’95 – DHP: Direct Hashing and Pruning

Echantillonnage

- Choisir un échantillon et extraire des motifs fréquents à l'aide d'Apriori
- Un passage sur la BD pour vérifier les itemsets fréquents :
  - Seulement la bordure des itemsets fréquents est vérifiée
  - exemple : verif. abcd au lieu de abc, ab, bc, ...
- Un autre passage pour trouver les motifs qui manquent

Eclat/MaxEclat et VIPER

- Tid-list: la liste des transcations contenant un itemset
  - représentation verticale
- Opération principale : intersection des tid-lists
  - A : t1,t2,t3
  - B ; t2,t3
  - support de (AB) =2

FP-Growth

- Pas de génération de candidats
- Bases de données projetées
- Nous verrons ce concept de façon plus étendue avec l'algorithme prefixSpan d'extraction de motifs séquentiels

Fouille de motifs fréquents sous contraintes

Motivations

- Trouver tous les motifs qui apparaissent dans une base de données
  - Risque d'être trop nombreux et pas être intéressant
- La fouille de données doit être interactive
  - l'utilisateur exprime ce qui doit être extrait
- Fouille de motifs sous contraintes
  - Flexibilité pour l'utilisateur : modélise son intérêt
  - Optimisation : pousser des contraintes pour une fouille plus efficace

Contraintes en fouille de motifs

- Contraintes sur les données - requêtes SQL
  - trouver les produits vendus ensembles dans les magasins de N.Y.
- Contraintes sur les niveaux/dimensions
  - région, prix, marque, catégories de consommateurs
- Contraintes sur les motifs ou règles
  - les petits achats (<10$) qui provoquent de gros achats (>100$)
- Contraintes sur l'intérêt : (suppport, confiance, ...)

Extraction de motifs sous contraintes

- Solution naïve : post-traitement
- Approches plus efficaces :
  - Analyser les propriétés des contraintes
  - «pousser» des contraintes en même temps que la fouille des motifs fréquents.

Anti-monotonie

- Si un motif X viole une contrainte, alors tous ses super-motifs aussi.
  - sum(X.Price) ≤ v : antimonotone
  - sum(X.Price) > v : non antimonotone
- Exemple :
  - C: range(S.profit) ≤ 15
  - itemset (ab) viole C
  - comme tous les sur-ensemble de (ab)

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20

Contraintes anti-monotone

Constraint	Antimonotone
$v \in S$	No
$S \supseteq V$	no
$S \subseteq V$	yes
$\min(S) \leq v$	no
$\min(S) \geq v$	yes
$\max(S) \leq v$	yes
$\max(S) \geq v$	no
$\mathrm{count}(S) \leq v$	yes
$\mathrm{count}(S) \geq v$	no
$\mathrm{sum}(S) \leq v$ ($a \in S, a \geq 0$)	yes
$\mathrm{sum}(S) \geq v$ ($a \in S, a \geq 0$)	no
$\mathrm{range}(S) \leq v$	yes
$\mathrm{range}(S) \geq v$	no
$\mathrm{avg}(S)\ \theta\ v, \theta \in \{=, \leq, \geq\}$	convertible
$\mathrm{support}(S) \geq \xi$	yes
$\mathrm{support}(S) \leq \xi$	no

Contraintes monotones

- X satisfait la contrainte C alors tous ses super motifs aussi.
  - sum(X.Price)>v (monotone)

Contraintes monotones

Constraint	Monotone
$v \in S$	yes
$S \supseteq V$	yes
$S \subseteq V$	no
$min(S) \leq v$	yes
$min(S) \geq v$	no
$max(S) \leq v$	no
$max(S) \geq v$	yes
$count(S) \leq v$	no
$count(S) \geq v$	yes
$sum(S) \leq v$ ($a \in S, a \geq 0$)	no
$sum(S) \geq v$ ($a \in S, a \geq 0$)	yes
$range(S) \leq v$	no
$range(S) \geq v$	yes
$avg(S)\ \theta\ v, \theta \in \{=, \leq, \geq\}$	convertible
$support(S) \geq \xi$	no
$support(S) \leq \xi$	yes

Contraintes succinctes

- La vérification qu'itemset X satisfasse une contrainte C peut se faire sans regarder les transactions.
  - min(X.price) >v (succinct)
  - sum(X.price) > v (non-succinct)

contraintes succinctes

Constraint	Succinct
$v \in S$	yes
$S \supseteq V$	yes
$S \subseteq V$	yes
$\min(S) \leq v$	yes
$\min(S) \geq v$	yes
$\max(S) \leq v$	yes
$\max(S) \geq v$	yes
$count(S) \leq v$	weakly
$count(S) \geq v$	weakly
$sum(S) \leq v$ ($a \in S, a \geq 0$)	no
$sum(S) \geq v$ ($a \in S, a \geq 0$)	no
$range(S) \leq v$	no
$range(S) \geq v$	no
$avg(S)\ \theta\ v, \theta \in \{=, \leq, \geq\}$	no
$support(S) \geq \xi$	no
$support(S) \leq \xi$	no

Contraintes convertibles

- Convertir une contrainte en une contrainte monotone ou anti-monotone en redéfinissant un ordre sur les items.

- ex : C:avg(S.profit)≥25

  - Ordonné les items en par leur prix (décroissant)

  - <a,f,g,d,b,h,c,e>

  - si (afb) viole C

    - (afb*) aussi

TDB (min_sup=2)

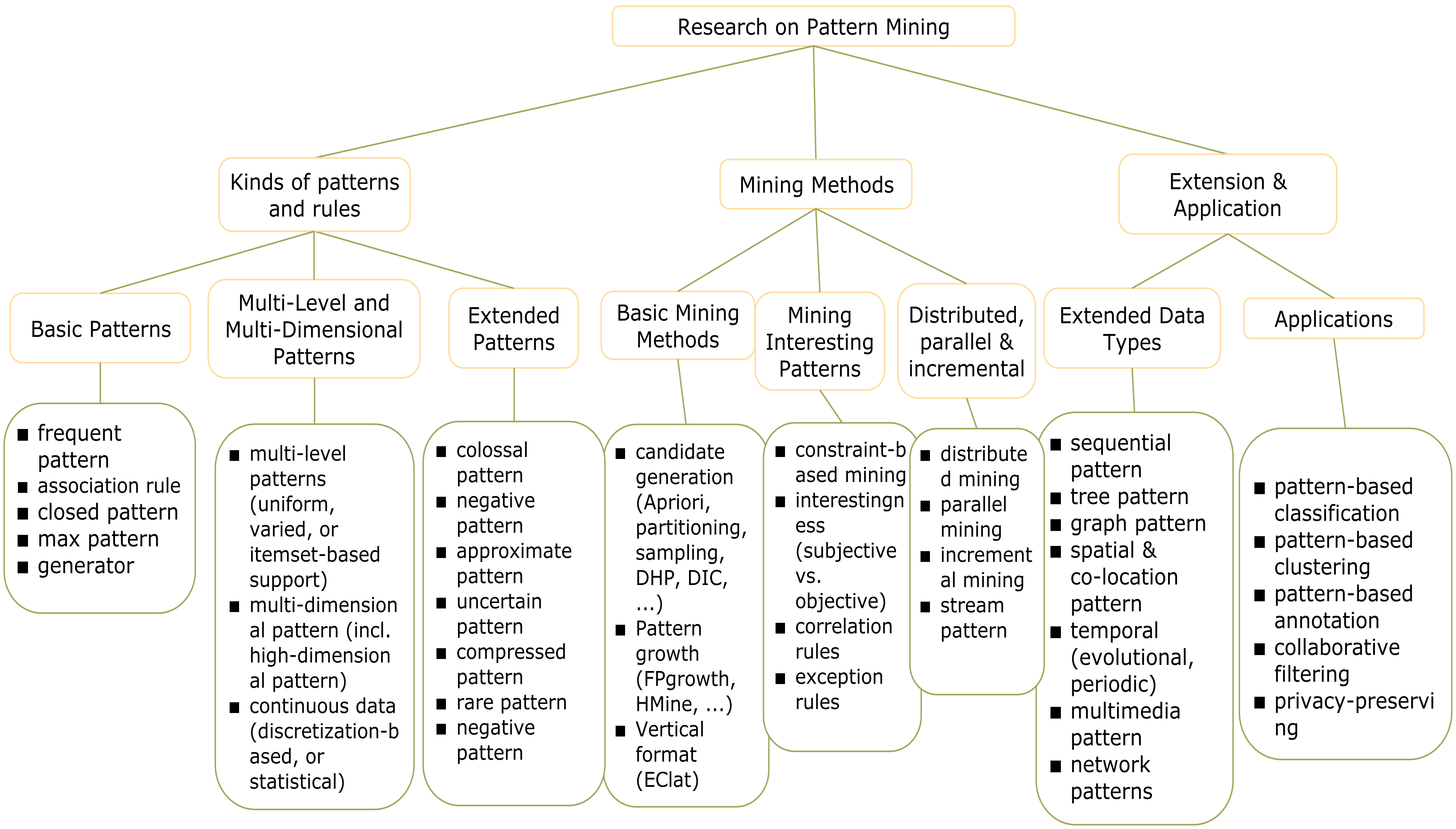
TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Introduction aux représentations condensées

Fouille de séquences (voir .pdf)

Fouille de motifs fréquents :



Clustering (regroupement, segmentation)

Différentes techniques de clustering

Problématique

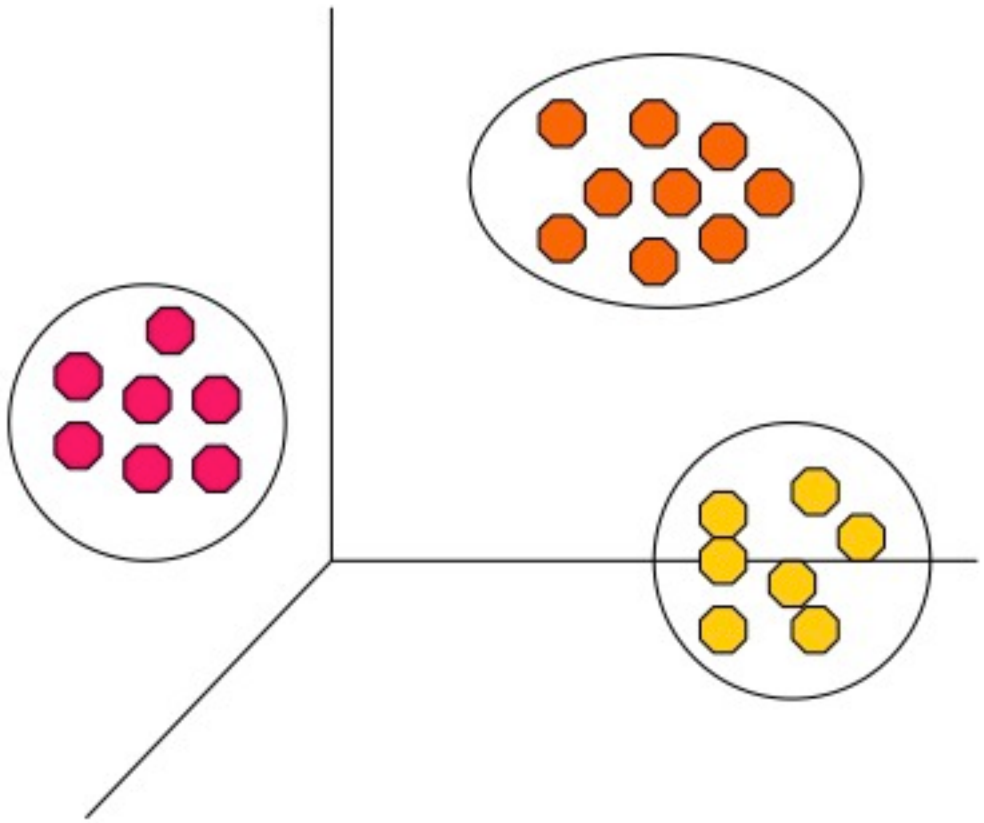
- Soient N instances de données à K attributs
- Trouver un partitionnement en c clusters (groupes) ayant du sens.
   - Affectation automatique de «labels» aux clusters
   - c peut être donné ou recherché
   - Les classes ne sont pas connues à l'avance (non supervisé)
   - Attributs de différents types

Qualité d’un clustering

- Une bonne méthode de regroupement permet de garantir
  - Une **grande similarité intra-groupe**
  - Une **faible similarité inter-groupe**
- La qualité d’un regroupement dépend donc de la **mesure de similarité** utilisée par la méthode et de son implémentation
- La **qualité d’une méthode** de clustering est évaluée par son abilité à découvrir certains ou tous les “patterns” cachés.

Objectif du clustering

- **Minimiser** les distances **intra**-clusters
- **Maximiser** les distances **inter**-clusters



Exemples d’applications

Marketing : segmentation du marché en découvrant des groupes de clients distincts à partir de bases de données d’achats.

- **Environnement :** identification des zones terrestres similaires (en termes d’utilisation) dans une base de données d’observation de la terre.

- **Assurance :** identification de groupes d’assurés distincts associés à un nombre important de déclarations.

- **Planification de villes :** identification de groupes d’habitations suivant le type d’habitation, valeur, localisation géographique, ...

- **Médecine** : Localisation de tumeurs dans le cerveau ␣ Nuage de points du cerveau fournis par le neurologue ␣ Identification des points définissant une tumeur :

Mesure de la similarité

- Pas de définition unique de la similarité entre objets
  - différentes mesures de distance d(x,y
- La définition de la similarité entre objets dépend de :
  - Le type de données considérées
  - Le type de similarité recherchée

Choix de la distance

- Propriété d'une distance :
  - $d(x,y) \geq 0$
  - $d(x,y) = 0$ ssi $x=y$
  - $d(x,y) = d(y,x)$
  - $d(x,z) \leq d(x,y) + d(y,z)$
- Définir une distance sur chacun des champs
- ex (numérique, e.g., âge, poids, taille, etc.): $d(x,y) = |x-y|$; $d(x,y) = |x-y|/dmax$ (dist. normalisée)

Types des variables

- Intervalles:
- Binaires:
- catégories, ordinales, ratio:
- Différents types:

Intervalle (discrètes)

- Standardiser les données
  - Calculer l'écart absolu moyen:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

  - où

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf}).$$

  - Calculer la mesure standardisée (z-score)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

Exemple

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

$$M_{Age} = 60 \qquad S_{Age} = 5$$

$$M_{salaire} = 11074 \qquad S_{salaire} = 148$$

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	2

Similarité entre objets

<u>Les distances</u> exprimment une similarité

Ex: la distance de Minkowski :

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)}$$

où i = (x_{i1}, x_{i2}, …, x_{ip}) et j = (x_{j1}, x_{j2}, …, x_{jp}) sont deux objets p-dimensionnels et q un entier positif

Si q = 1, d est la distance de Manhattan

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

Similarité entre objets(I)

Si q = 2, d est la distance Euclidienne :

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

Propriétés

$d(i,j) \geq 0$

$d(i,i) = 0$

$d(i,j) = d(j,i)$

$d(i,j) \leq d(i,k) + d(k,j)$

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

d(p1,p2)=120

d(p1,p3)=132

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

d(p1,p2)=120

d(p1,p3)=132

Conclusion: p1 ressemble plus à p2 qu'à p3!!! :-(

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

d(p1,p2)=120

d(p1,p3)=132

Conclusion: p1 ressemble plus à p2 qu’à p3!!! :-(

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	0

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

d(p1,p2)=120

d(p1,p3)=132

Conclusion: p1 ressemble plus à p2 qu'à p3!!! :-(

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	0

d(p1,p2)=4,675

d(p1,p3)=2,324

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

d(p1,p2)=120

d(p1,p3)=132

Conclusion: p1 ressemble plus à p2 qu'à p3!!! :-(

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	0

d(p1,p2)=4,675

d(p1,p3)=2,324

Conclusion: p1 ressemble plus à p3 qu'à p2 !!! :-)

Variables binaires

- Une table de contingence pour données binaires

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

a= nombre de positions où i a 1 et j a 1

- Exemple o_i=(1,1,0,1,0) et o_j=(1,0,0,0,1)
- a=1, b=2, c=1, d=2

Mesures de distances

- Coefficient d'appariement (matching) simple (invariant pour variables symétriques):

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

- Exemple o_i=(1,1,0,1,0) et o_j=(1,0,0,0,1)
- $d(o_i, o_j)$=3/5
- Coefficient de Jaccard
- $d(o_i, o_j)$=3/4

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

Variables binaires (I)

- Variable <u>symétrique</u>: Ex. le sexe d'une personne, i.e coder masculin par 1 et féminin par 0 c'est pareil que le codage inverse

- Variable <u>asymétrique:</u> Ex. Test HIV. Le test peut être positif ou négatif (0 ou 1) mais il y a une valeur qui sera plus présente que l'autre. Généralement, on code par 1 la modalité la moins fréquente
  - 2 personnes ayant la valeur 1 pour le test sont plus similaires que 2 personnes ayant 0 pour le test

Variables binaires(II)

- Exemple

Nom	Sexe	Fièvre	Toux	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

- Sexe est un attribut symétrique
- Les autres attributs sont asymétriques
- Y et P $\equiv$ 1, N $\equiv$ 0, la distance n'est mesurée que sur les asymétriques

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Les plus similaires sont Jack et Mary$\Rightarrow$atteints du même mal

Variables Nominales

- Une généralisation des variables binaires, ex: rouge, vert et bleu
- Méthode 1: Matching simple
  - m: # d’appariements, p: # total de variables

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

- Méthode 2: utiliser un grand nombre de variables binaires
  - Créer une variable binaire pour chaque modalité (ex: variable rouge qui prend les valeurs vrai ou faux)

Variables Ordinales

- Une variable ordinale peut être discrète ou continue
- L'ordre peut être important, ex: classement
- Peuvent être traitées comme les variables intervalles
  - remplacer x_{if} par son rang $r_{if} \in \{1, \ldots, M_f\}$
  - Remplacer le rang de chaque variable par une valeur dans [0, 1] en remplaçant la variable f dans l'objet I par

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

  - Utiliser une distance pour calculer la similarité

En Présence de Variables de différents Types

- Pour chaque type de variables utiliser une mesure adéquate. Problèmes: les clusters obtenus peuvent être différents
- On utilise une formule pondérée pour faire la combinaison

  - f est binaire ou nominale: $$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$
    - $d_{ij}^{(f)} = 0$ si $x_{if} = x_{jf}$, sinon $d_{ij}^{(f)} = 1$
  - f est de type intervalle: utiliser une distance normalisée
  - f est ordinale
    - calculer les rangs r_{if} et $$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$
    - Ensuite traiter z_{if} comme une variable de type intervalle

Approches de Clustering

- Algorithmes de Partitionnement: Construire plusieurs partitions puis les évaluer selon certains critères

- Algorithmes hiérarchiques: Créer une décomposition hiérarchique des objets selon certains critères

- Algorithmes basés sur la densité: basés sur des notions de connectivité et de densité

Méthodes de clustering : caractéristiques

- Extensibilité
- Abilité à traiter différents types de données
- Découverte de clusters de différents formes
- Connaissances requises (paramètres de l'algorithme)
- Abilité à traiter les données bruitées et isolées.

Algorithmes à partionnement

- Construire une partition à **k** clusters d'une base **D** de **n** objets
- Les k clusters doivent optimiser le critère choisi
  - Global optimal: Considérer toutes les k-partitions
  - Heuristic methods: Algorithmes k-means et k-medoids
  - k-means (MacQueen'67): Chaque cluster est représenté par son centre
  - k-medoids or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Chaque cluster est représenté par un de ses objets

La méthode des k-moyennes (K-Means)

- L'algorithme k-means est en 4 étapes :

1. Choisir k objets formant ainsi k clusters

2. (Ré)affecter chaque objet O au cluster C_i de centre M_i tel que dist(O,M_i) est minimal

3. Recalculer M_i de chaque cluster (le barycentre)

4. Aller à l'étape 2 si on vient de faire une affectation

K-Means :Exemple

- A={1,2,3,6,7,8,13,15,17}. Créer 3 clusters à partir de A

- On prend 3 objets au hasard. Supposons que c'est 1, 2 et 3. Ca donne C_1={1}, M_1=1, C_2={2}, M_2=2, C_3={3} et M_3=3

- Chaque objet O est affecté au cluster au milieu duquel, O est le plus proche. 6 est affecté à C_3 car dist(M_3,6)<dist (M_2,6) et dist(M_3,6)<dist(M_1,6); On a
  - C_1={1}, M_1=1,
  - C_2={2}, M_2=2
  - C_3={3, 6,7,8,13,15,17}, M3=69/7=9.86

K-Means :Exemple (suite)

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. $C_1=\{1\}$, $M_1=1$, $C_2=\{2,3\}$, $M_2=2.5$,$C_3=\{6,7,8,13,15,17\}$ et $M_3= 66/6=11$

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11
- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12
- dist(2,M1)<dist(2,M2)➔2 passe en C1. dist(7,M2)<dist(7,M3)➔ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

- dist(2,M1)<dist(2,M2)➔2 passe en C1. dist(7,M2)<dist(7,M3)➔ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

- dist(2,M1)<dist(2,M2)➔2 passe en C1. dist(7,M2)<dist(7,M3)➔ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

- dist(3,M1)<dist(3,M2)➔3 passe en 1. dist(8,M2)<dist(8,M3)➔8 passe en 2

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

- dist(2,M1)<dist(2,M2)➔2 passe en C1. dist(7,M2)<dist(7,M3)➔ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

- dist(3,M1)<dist(3,M2)➔3 passe en 1. dist(8,M2)<dist(8,M3)➔8 passe en 2
- C1={1,2,3}, M1=2, C2={6,7,8}, M2=7, C3={13,15,17}, M3=15

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

- dist(2,M1)<dist(2,M2)➔2 passe en C1. dist(7,M2)<dist(7,M3)➔ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

- dist(3,M1)<dist(3,M2)➔3 passe en 1. dist(8,M2)<dist(8,M3)➔8 passe en 2
- C1={1,2,3}, M1=2, C2={6,7,8}, M2=7, C3={13,15,17}, M3=15

K-Means :Exemple (suite)

- dist(3,M2)<dist(3,M3)➔3 passe dans C_2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3}, M_2=2.5,C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2. Tous les autres objets ne bougent pas. C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

- dist(2,M1)<dist(2,M2)➔2 passe en C1. dist(7,M2)<dist(7,M3)➔ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

- dist(3,M1)<dist(3,M2)➔3 passe en 1. dist(8,M2)<dist(8,M3)➔8 passe en 2
- C1={1,2,3}, M1=2, C2={6,7,8}, M2=7, C3={13,15,17}, M3=15

- **Plus rien ne bouge**

Illustration (1)

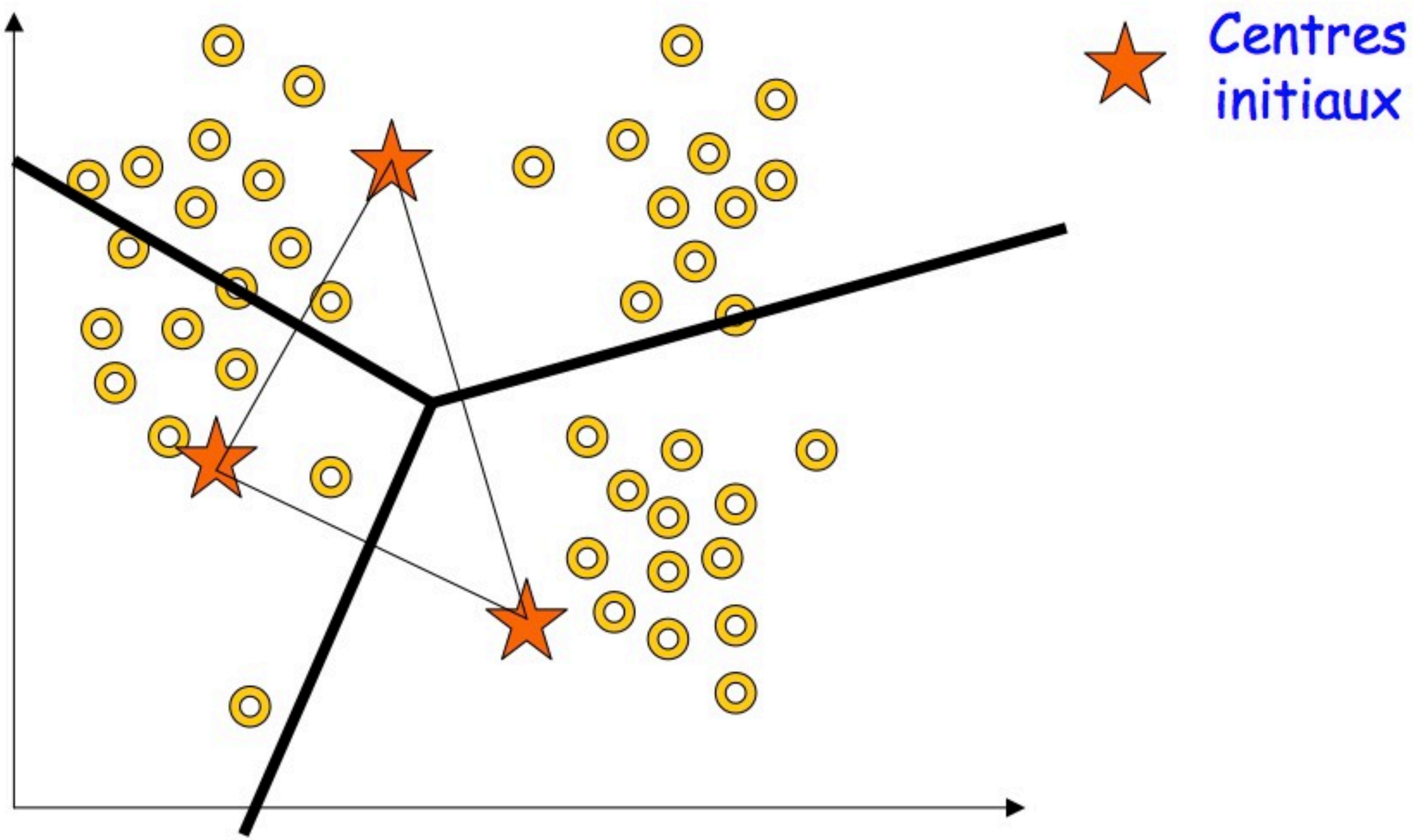


Illustration (2)

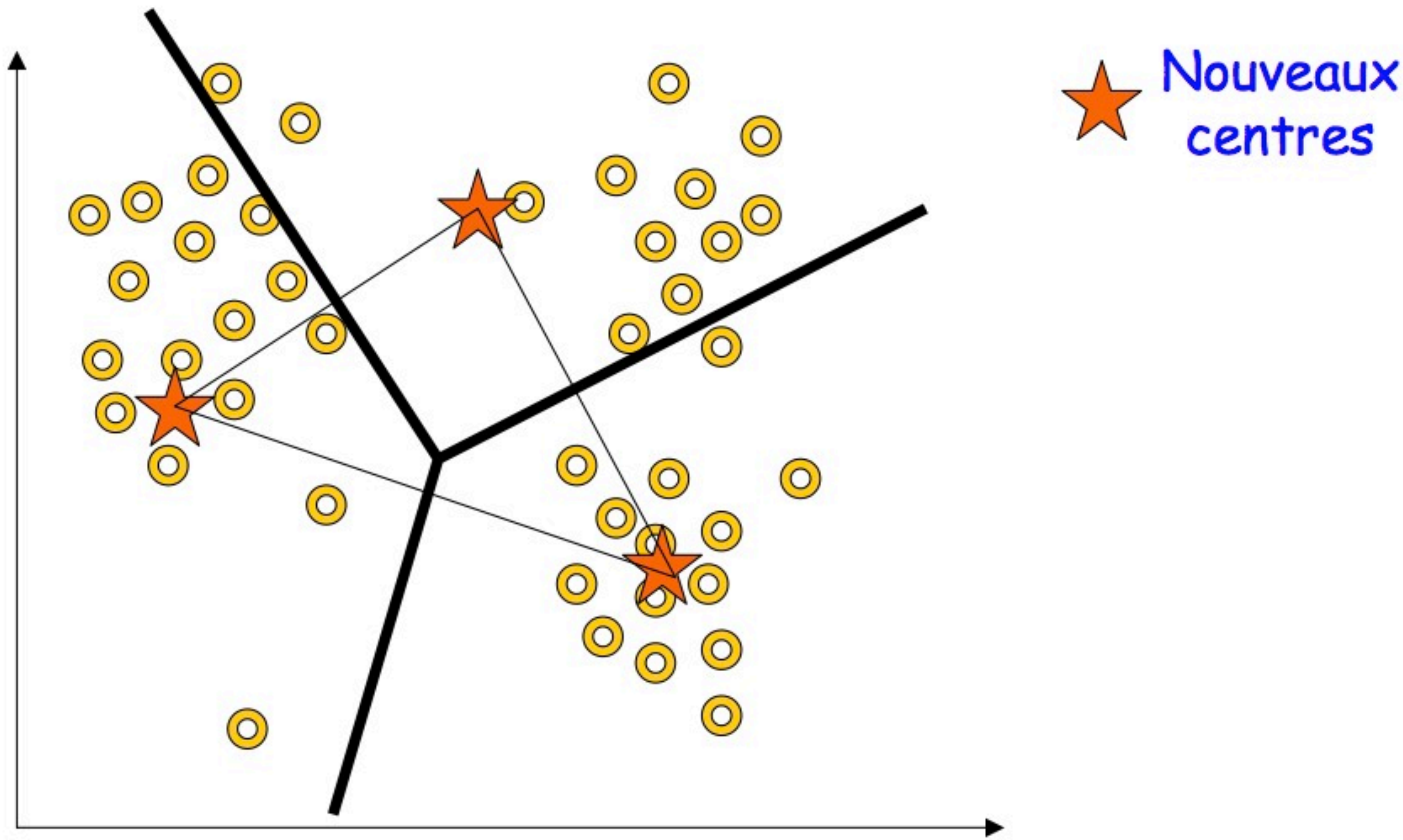
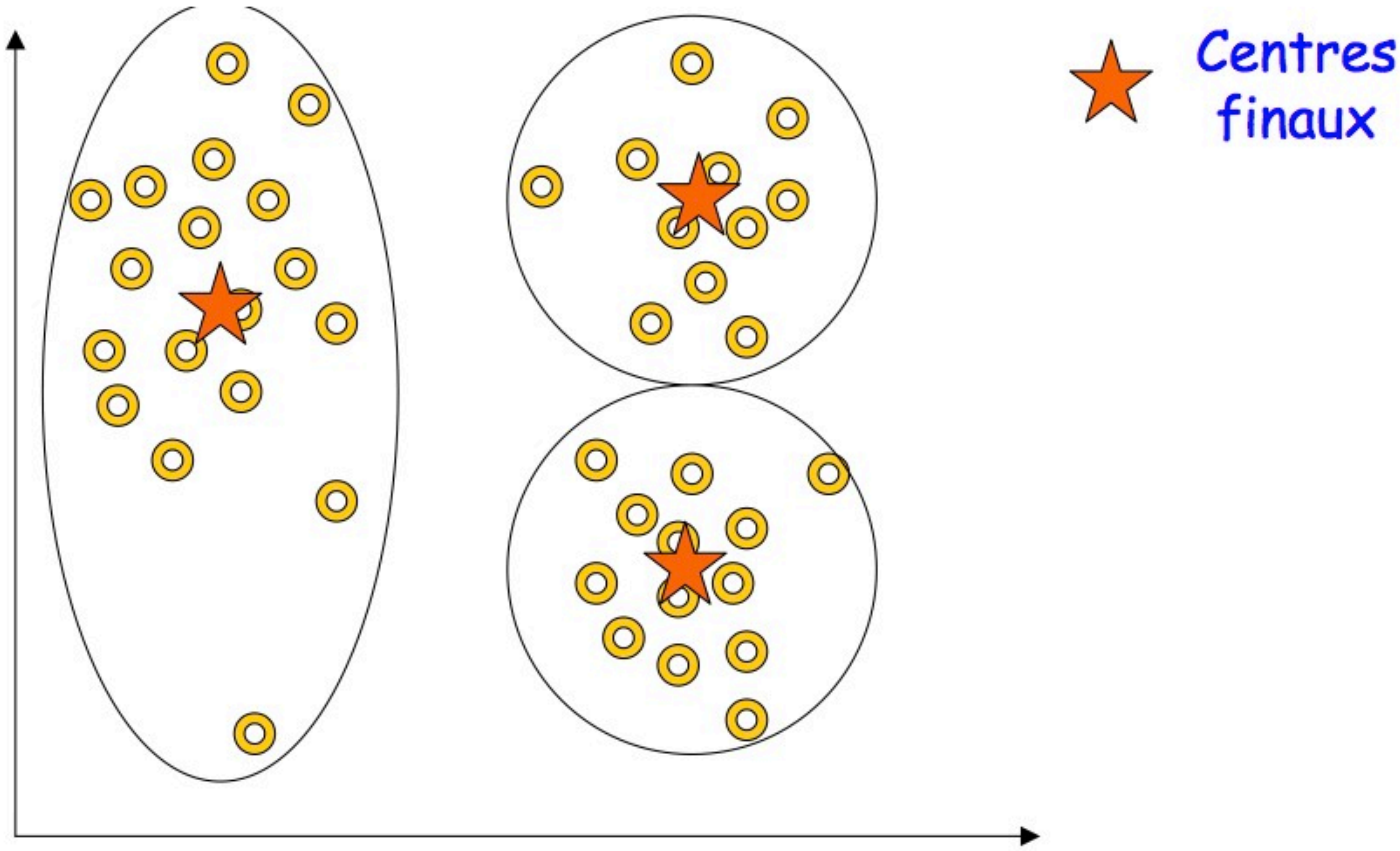


Illustration (3)



Exemple

- 8 points A, ..., H de l 'espace euclidéen 2D. k=2 (2 groupes)
- Tire aléatoirement 2 centres : B et D choisis.

points	Centre D(2,4), B(2,2)	Centre D(2,4), I(27/7,17/7)	Centre J(5/3,10/3), K(24/5,11/5)
A(1,3)	B	D	J
B(2,2)	B	I	J
C(2,3)	B	D	J
D(2,4)	D	D	J
E(4,2)	B	I	K
F(5,2)	B	I	K
G(6,2)	B	I	K
H(7,3)	B	I	K

Commentaires sur la méthode des K-Means

- Force
  - **Relativement extensible** dans le traitement d'ensembles de taille importante
  - **Relativement efficace** : O(t.k.n), où n représente # objets, k # clusters, et t # iterations. Normalement, k, t << n.
- Faiblesses
  - N'est pas applicable en présence d'attributs où la **moyenne** n'est pas définie
  - On doit spécifier **k** (nombre de clusters)
  - Incapable de traiter des données **bruitées**
  - Les clusters sont construits par rapports à des **objets inexistants** (les milieux)
  - Ne peut pas découvrir les **groupes non-convexes**
  - Les **outliers** sont mal gérés.

Variantes des K-means

- Sélection des centres initiaux
- Calcul des similarités
- Calcul des centres (K-medoids : [Kaufman & Rousseeuw'87])
- GMM : Variantes de K-moyennes basées sur les probabilités
- K-modes : données catégorielles [Huang'98]
- K-prototype : données mixtes (numériques et catégorielles)

Méthodes hiérarchiques

- **Une méthode hiérarchique :** construit une hiérarchie de clusters, non seulement une partition unique des objets.
- Le nombre de clusters **k** n'est pas exigé comme donnée
- Utilise une matrice de distances comme critère de clustering
- Une **condition de terminaison** peut être utilisée (ex. Nombre de clusters)

Méthodes hiérarchiques

Entrée : un échantillon de m enregistrements x1, ..., xm

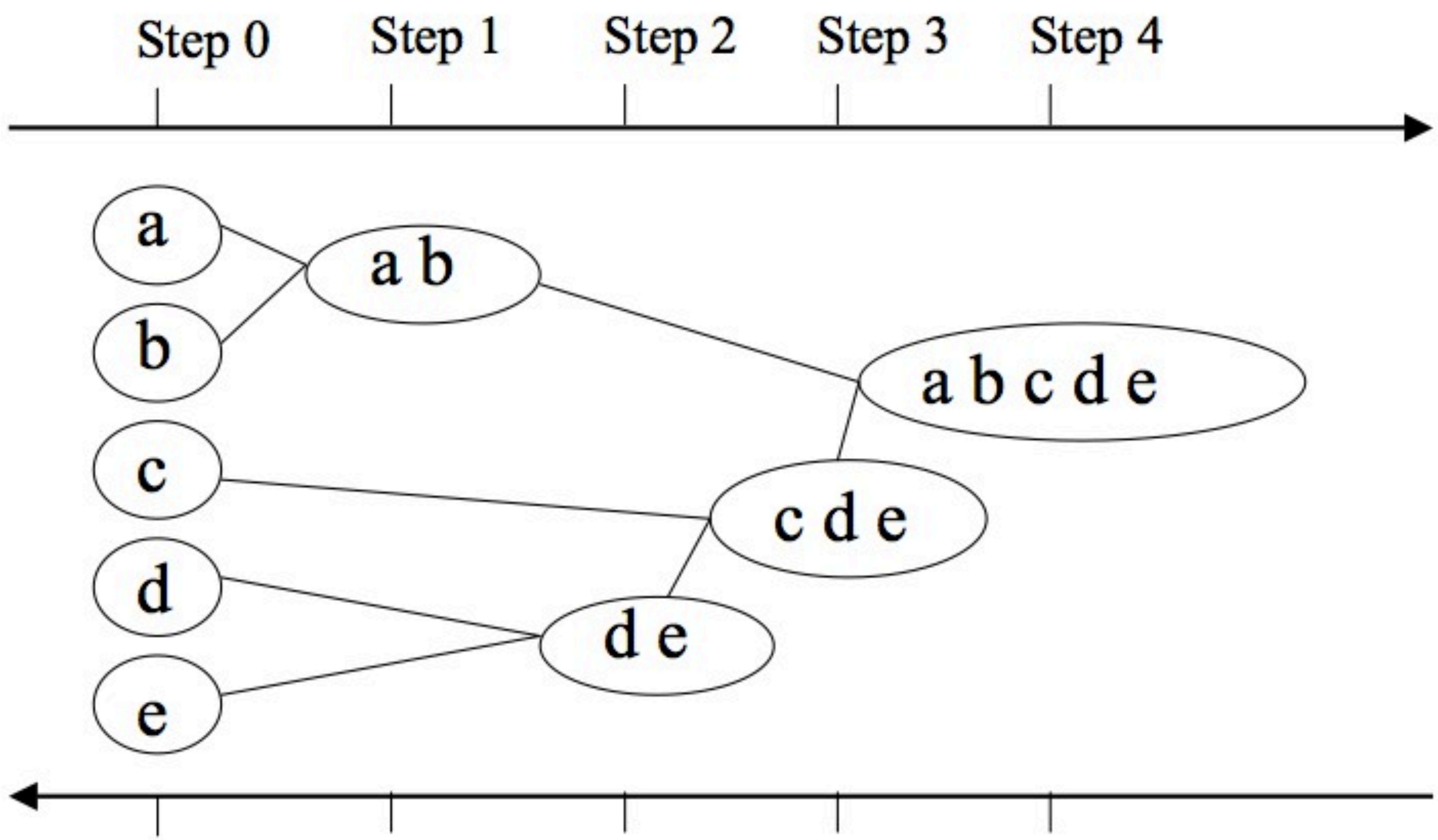
1. On commence avec m clusters (cluster = 1 enregistrement)

2. Grouper les deux clusters les plus « proches ».

3. S'arrêter lorsque tous les enregistrements sont membres d'un seul groupe

4. Aller en 2.

Arbre de clusters



- Résultat : Graphe hiérarchique qui peut être coupé à un niveau de **dissimilarité** pour former une **partition.**

- La hiérarchie de clusters est représentée comme un arbre de clusters, appelé **dendrogramme**

- Les **feuilles** de l’arbre représentent les **objets**

- Les noeuds intermédiaires de l’arbre représentent les **clusters**

Distances entre clusters

- Distance entre les centres des clusters (Centroid Method)

- Distance minimale entre toutes les paires de données des 2 clusters (Single Link Method) $d(i, j) = \min_{x \in C_i, y \in C_j} \{d(x, y)\}$

- Distance maximale entre toutes les paires de données des 2 clusters (Complete Link Method) $d(i, j) = \max_{x \in C_i, y \in C_j} \{d(x, y)\}$

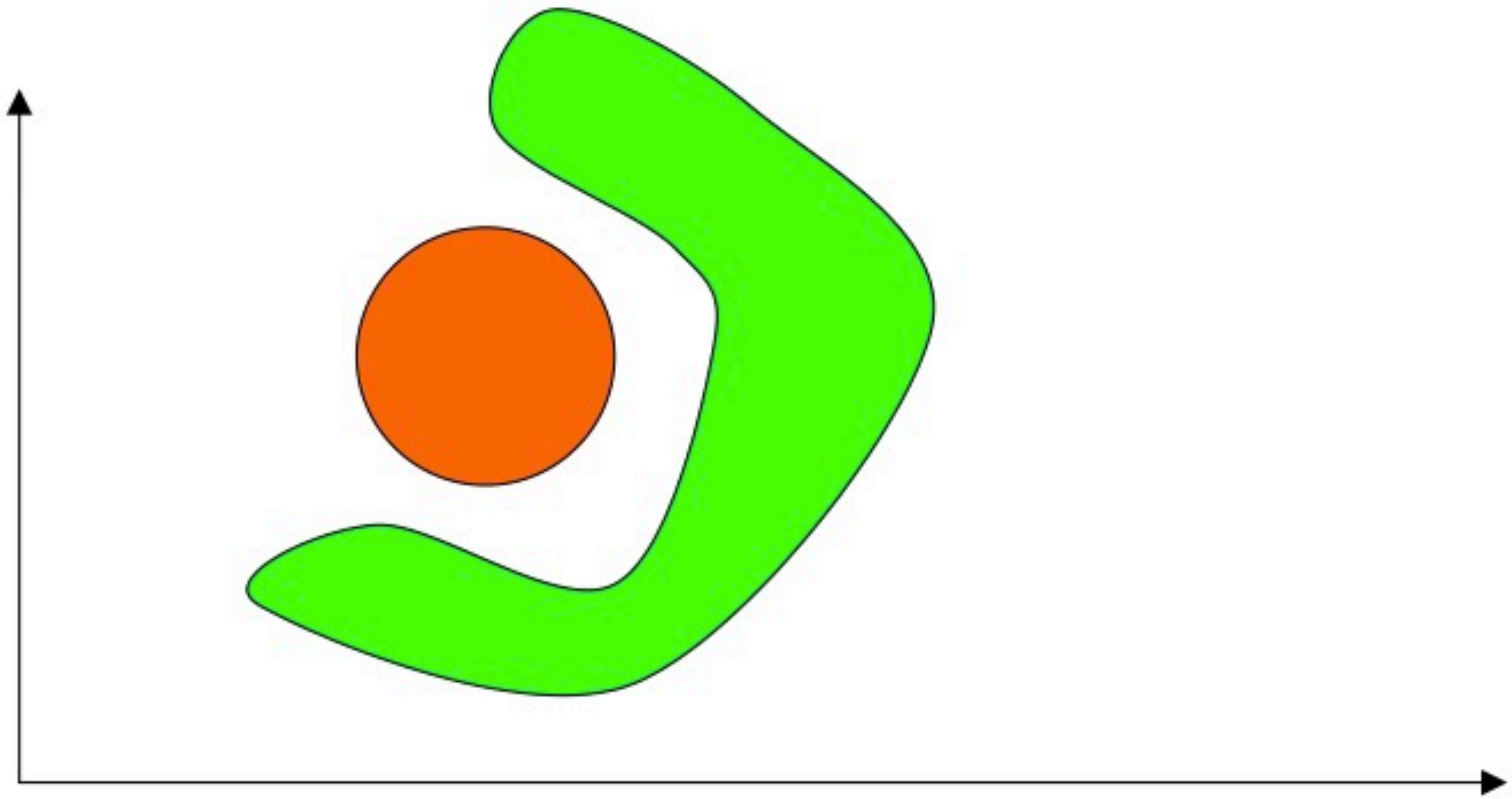
- Distance moyenne entre toutes la paires d'enregistrements (Average Linkage) $d(i, j) = avg_{x \in C_i, y \in C_j} \{d(x, y)\}$

+ et -

- Avantages :
  - **Conceptuellement simple**
  - **Propriétés théoriques** sont bien connues
  - Quand les clusters sont groupés, la décision est définitive => le nombre d'alternatives différentes à à examiner est réduit
- Inconvénients :
  - **Groupement** de clusters est **définitif =>** décisions erronnées sont **impossibles à modifier** ultérieurement
  - Méthodes **non extensibles** pour des ensembles de données de grandes tailles

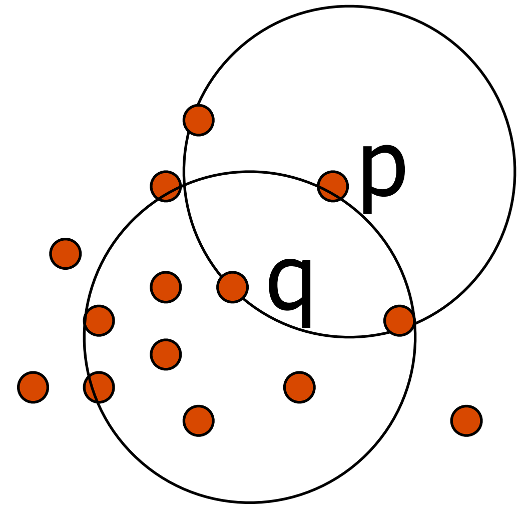
Méthode basée sur la densité

- Pour ce types de problèmes, l’utilisation de mesures de similarité (distance) est moins efficace que l’utilisation de **densité de voisinage.**



Clustering basé sur la densité

- Voit les clusters comme des régions denses séparées par des régions qui le sont moins (bruit)
- Deux paramètres**:**
  - **Eps**: Rayon maximum du voisinage
  - **MinPts**: Nombre minimum de points dans le voisinage-Eps d'un point
- **Voisinage : $\mathbf{V_{Eps}(p)}$**: **$\{q \in D \mid dist(p,q) <= Eps\}$**
- Un point **p** est directement densité-accessible à partir de **q** resp. à **Eps**, **MinPts** si
  - 1) **p** $\in$ **$V_{Eps}(q)$**
  - 2) **$|V_{Eps}(q)|$** >= **MinPts**

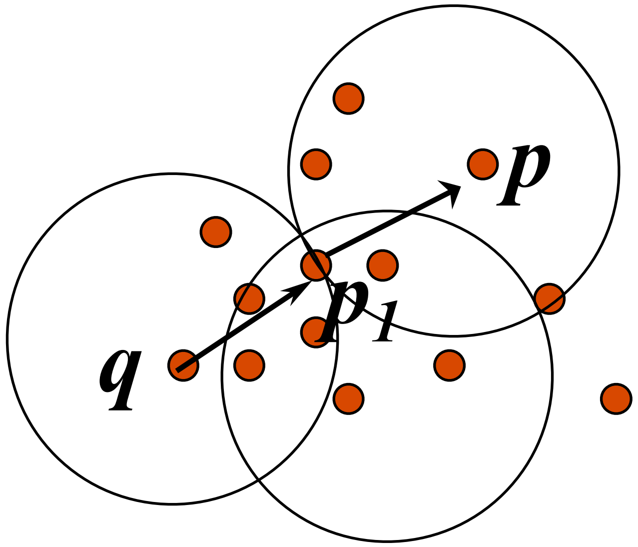


MinPts = 5

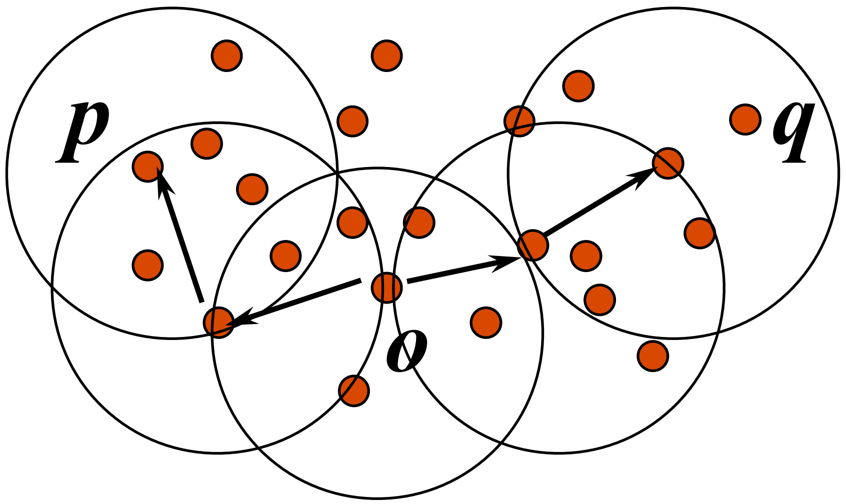
Eps = 1 cm

Clustering basé sur la densité

- Accessibilité:
  - p est accessible à partir de q resp. à Eps, MinPts si il existe p_1, …, p_n, $p_1 = q$, $p_n = p$ t.q p_{i+1} est directement densité accessible à partir de p_i

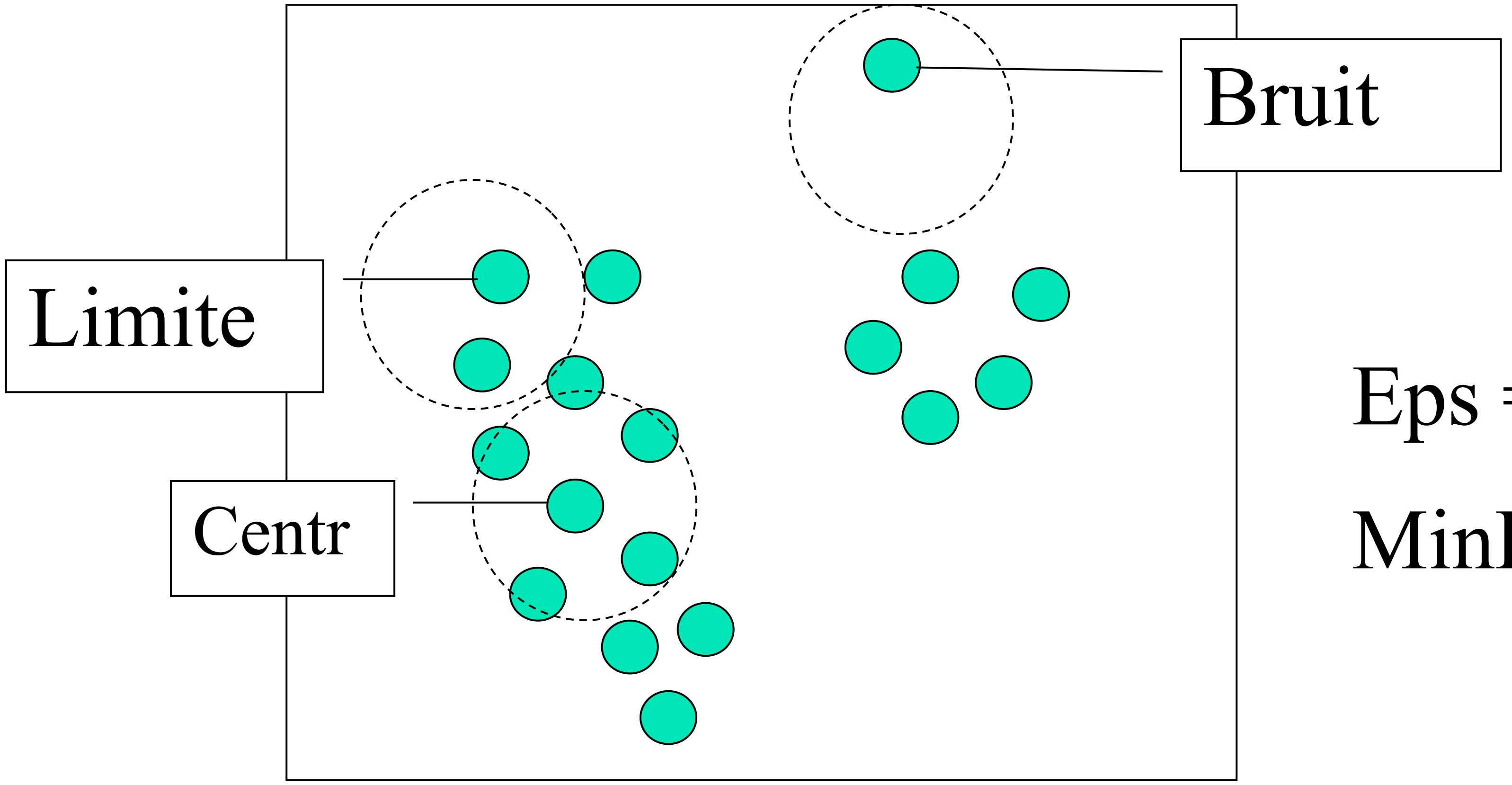


- Connexité
  - p est connecté à q resp. à Eps, MinPts si il existe un point o t.q p et q accessibles à partir de o resp. à Eps et MinPts.



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Un cluster est l'ensemble maximal de points connectés
- Découvre des clusters non nécessairement convexes



Eps = 1cm

MinPts = 5

DBSCAN: l'algorithme

- Choisir **p**
- Récupérer tous les points accessibles à partir de **p** resp. **Eps** et **MinPts**.
- Si **p** est un centre, un cluster est formé.
- si **p** est une limite, alors il n'y a pas de points accessibles de **p** : passer à un autre point
- Répéter le processus jusqu'à épuiser tous les points.

Résumé

- Le clustering groupe des objets en se basant sur leurs **similarités**.
- Le clustering possède plusieurs applications.
- La mesure de similarité peut être calculée pour **différents types** de données.
- La sélection de la **mesure de similarité** dépend des données utilisées et le type de similarité recherchée.

- Les méthodes de clustering peuvent être classées en :
    - Méthodes de partitionnement,
    - Méthodes hiérarchiques,
    - Méthodes à densité de voisinage
- Plusieurs travaux de recherche sur le clustering en cours et en perspective.
- Plusieurs applications en **perspective** : Génomique, Environnement, ...