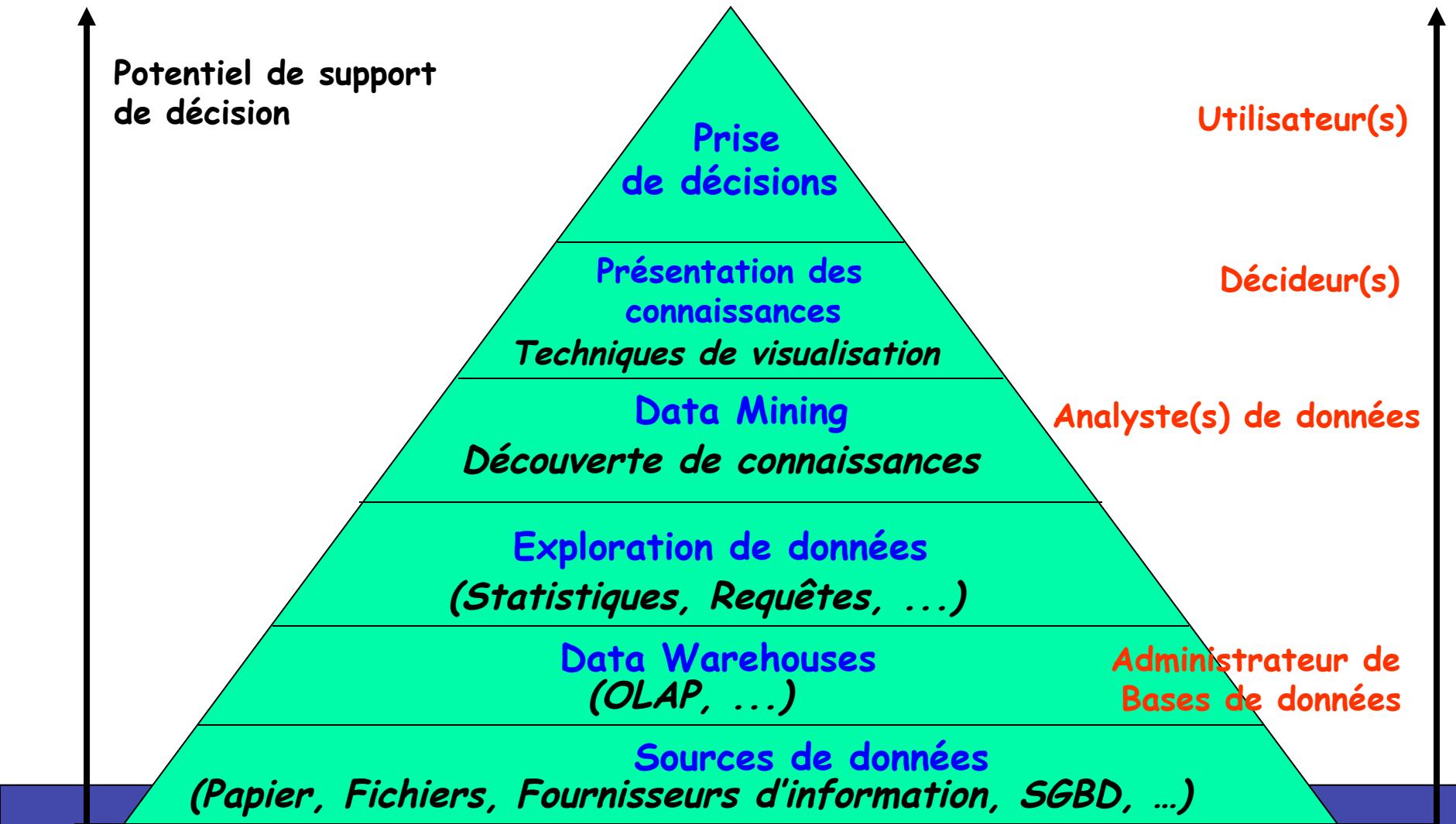


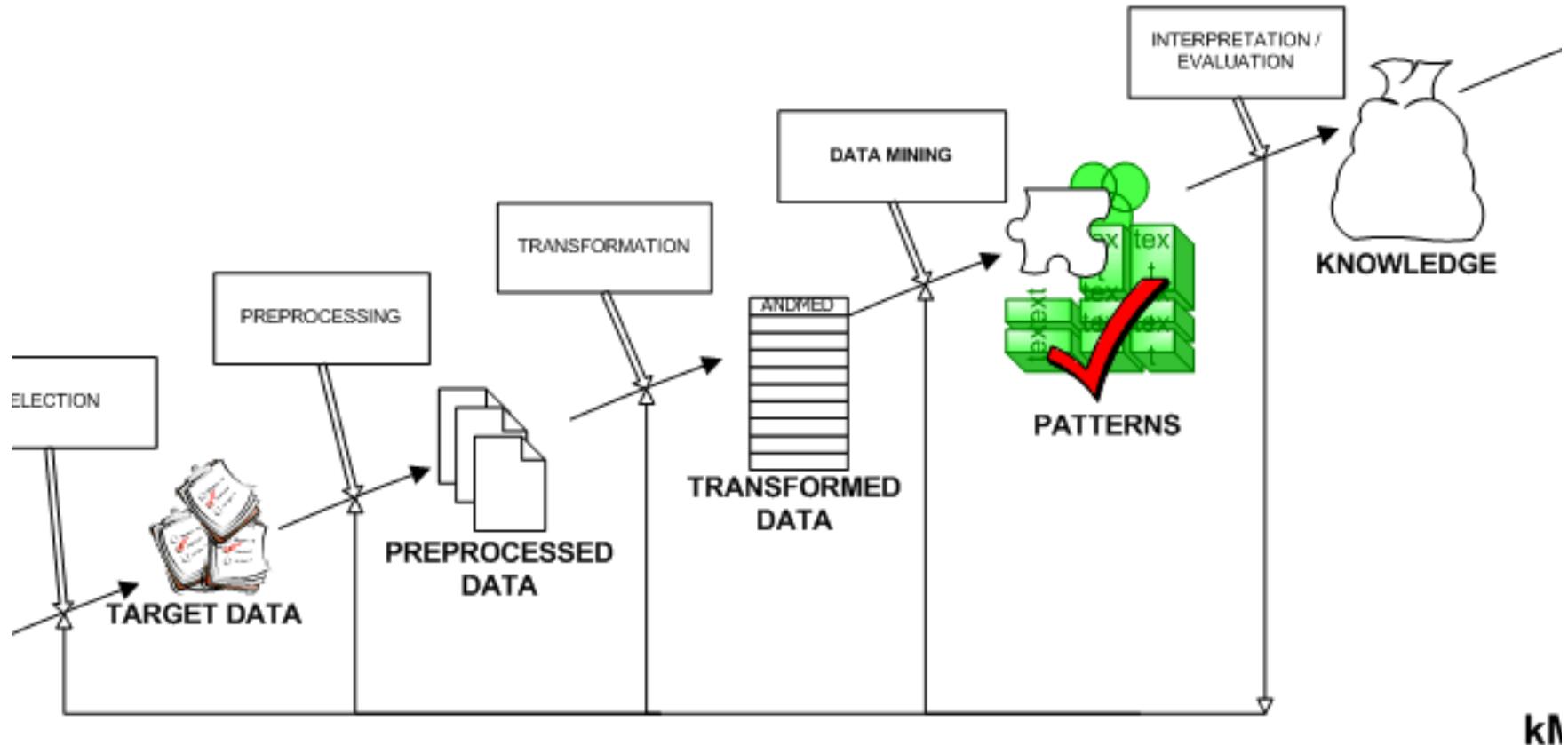
Fouille de données



Data Mining et aide à la décision



Un processus complet



La fouille de données = Datamining

Le **datamining** est l'ensemble des

- Techniques et méthodes
- ... Destinées à l'exploration et l'analyse
- ... De (souvent) grandes bases de données
- ... En vue de détecter dans ces données des règles, des associations, des tendances inconnues (non fixées a priori) des structures particulières restituant de façon concise l'essentiel de l'information utile
- ... Pour l'aide à la décision

Souvent on utilise le terme extraire de l'information de la donnée

Selon le MIT, le Datamining est l'une des 10 technologies émergentes qui changeront le monde au XXI siècle

Démarche méthodologique (1)

Comprendre l'application

- Connaissances *a priori*, objectifs, etc.

Sélectionner un échantillon de données

- Choisir une méthode d'échantillonnage

Nettoyage et transformation des données

- Supprimer le «bruit» : données superflues, marginales, données manquantes, etc.
- Effectuer une sélection d'attributs, réduire la dimension du problème, discréétisation des variables continues, etc.

Appliquer les techniques de fouille de données (DM)

- le cœur du KDD
- Choisir le bon modèle et le bon algorithme

Démarche méthodologique (2)

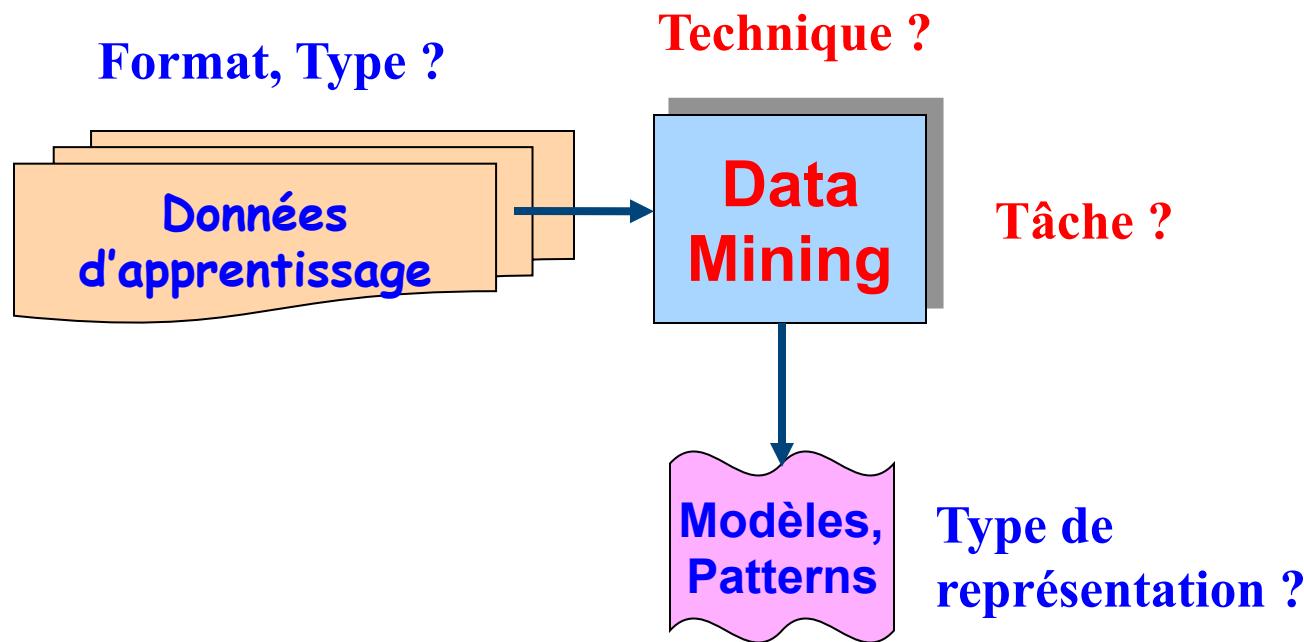
- Visualiser, évaluer et interpréter les modèles découverts

- Analyser la connaissance (intérêt, critères d'évaluation)
- Compréhensibilité souvent capitale
- Vérifier sa validité (sur le reste de la base de données)
- Réitérer le processus si nécessaire

- Gérer/déployer la connaissance découverte

- La mettre à la disposition des décideurs
- L'échanger avec d'autres applications (système expert, ...)
- etc.

Paramètres d'un processus KDD



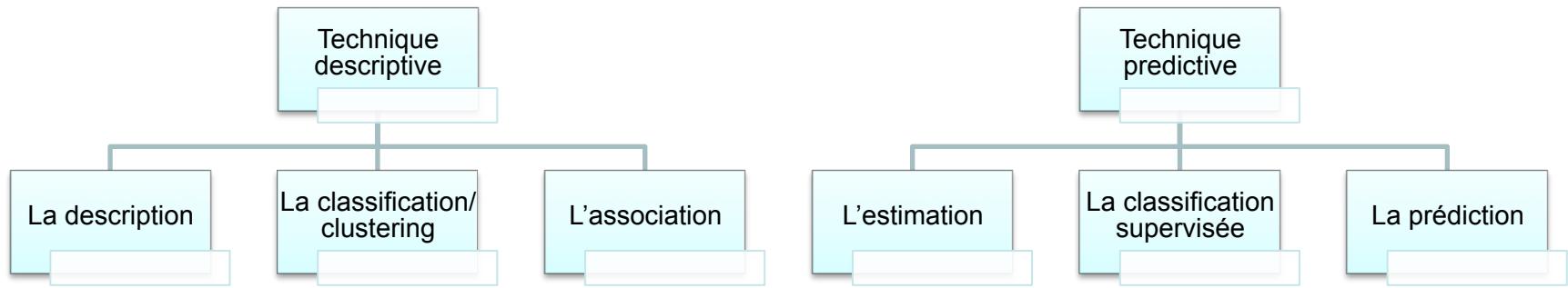
Les données : matière première

Valeurs des champs (p attributs ou variables) des enregistrements (n lignes ou cas) des tables de l'entrepôt (base de données, matrice n*p)

Types :

- Données discrètes : données binaires (sexe, ...), données énumératives (couleur, ...), énumératives ordonnées (réponses 1:très satisfait, 2:satisfait, ...).
- Données continues : données entières ou réelles (âge, salaire, ...)
- Dates
- Données textuelles
- Pages/liens web, Multimédia, ...

2 types de techniques



Les 2 types de techniques

- Les techniques descriptives (recherche de patterns) :
 - Visent à mettre en évidence des informations présentes mais cachées par le volume de données
 - Réduisent, résument et synthétisent les données
 - Il n'y a pas de variables à expliquer
- Les techniques prédictives (modélisation) :
 - Visent à extrapoler de nouvelles informations à partir des informations présentes (c'est le cas du scoring)
 - Expliquent les données
 - Il y a une variable à expliquer

1 : la description (technique descriptive)

Principe :

La description consiste à mettre au jour

- Pour une variable donnée : la répartition de ses valeurs (tri, histogramme, moyenne, minimum, maximum, etc.).
- Pour deux ou trois variables données : des liens entre les répartitions des valeurs des variables. Ces liens s'appellent des « tendances ».

Intérêt :

- Favoriser la connaissance et la compréhension des données.

Méthode :

- Méthodes graphiques pour la clarté : analyse exploratoire des données.

Exemples :

- Répartition des votes par âge (lien entre les variables « vote » et « âge »).

2 : la classification/clustering (technique descriptive)

Principe :

La classification (ou clustering ou segmentation) consiste à créer des classes/groupe (c'est-à-dire des sous-ensembles) de données similaires entre elles et différentes des données d'une autre Classe. Elle permet une vision générale de l'ensemble (de la clientèle, par exemple).

Intérêt :

- Favoriser, grâce à la métatypologie, la compréhension et la prédiction.
- Fixer des segments qui serviront d'ensemble de départ pour des analyses approfondies.
- Réduire les dimensions, c'est-à-dire le nombre d'attributs, quand il y en a trop au départ.

Méthodes :

- Classification hiérarchique
- Classification des K moyennes
- Réseaux de Kohonen.
- Règles d'association.

Exemples :

- Métatypologie d'une clientèle en fonction de l'âge, les revenus, le caractère urbain ou rural, la taille des villes, etc.

3 : l'association (technique descriptive)

Principe :

L'association consiste à trouver quelles valeurs des variables vont ensemble.

Par exemple, telle valeur d'une variable va avec telle valeur d'une autre variable.

Les règles d'association sont de la forme : si antécédent, alors conséquence.

L'association ne fixe pas de variable cible. Toute les variables peuvent à la fois être prédicteurs et variable cible.

On appelle aussi ce type d'analyse une « analyse d'affinité ».

Intérêt : Mieux connaître les comportements.

Méthodes : Algorithme a priori.

Exemples :

- Analyse du panier de la ménagère (si j'achète des fraises, alors j'achète des cerises).
- Étudier quelle configuration contractuelle d'un abonné d'une compagnie de téléphone

4 : l'estimation (technique prédictive)

Principe :

L'estimation consiste à définir le lien entre un ensemble de prédicteurs et une variable cible. Ce lien est défini à partir de données « complètes », c'est-à-dire dont les valeurs sont connues tant pour les prédicteurs que pour la variable cible. Ensuite, on peut déduire une variable cible inconnue de la connaissance des prédicteurs.

À la différence de la classification supervisée qui travaille sur une variable cible catégorielle, l'estimation travaille sur une variable cible numérique.

Intérêt : Permettre l'estimation de valeurs inconnues.

Méthodes :

- Analyse statistique classique : régression linéaire simple, corrélation, régression multiple, intervalle de confiance, estimation de points.
- Réseaux de neurones

Exemples :

- Estimer la pression sanguine à partir de l'âge, le sexe, le poids et le niveau de sodium dans le sang.
- Estimer les résultats dans les études supérieures en fonction de critères sociaux.

5 : la classification supervisée

(technique prédictive)

Principe :

C'est une estimation qui travaille sur une variable cible catégorielle.

Intérêt : Permettre l'estimation de valeurs inconnues.

Méthodes :

- Graphiques et nuages de points.
- Méthode des k plus proches voisins.
- Arbres de décision.
- Réseau de neurones.

Exemples :

- Segmentation par tranche de revenus : élevé, moyen et faible (3 segments). On cherche les caractéristiques qui conduisent à ces segments.
- Déterminer si un mode de remboursement présente un bon ou un mauvais niveau de risque crédit (deux segments).

6 : la prévision (technique prédictive)

Principe :

La prévision est similaire à l'estimation et à la segmentation mise à part que pour la prévision, les résultats portent sur le futur.

Intérêt : Permettre l'estimation de valeurs inconnues.

Méthodes : Celles de l'estimation ou de la segmentation.

Exemples :

- Prévoir le prix d'action à trois mois dans le futur.
- Prévoir le temps qu'il va faire.
- Prévoir le gagnant du championnat de football, par rapport à une comparaison des résultats des équipes.

Intérêt du Datamining

On ne veut pas simplement confirmer des intuitions a priori par des requêtes dans les BD mais détecter sans a priori les combinaisons de critères les plus discriminantes

- Ex: dans le domaine commercial, on ne veut pas savoir « Combien de clients ont acheté tel produit pendant telle période »

MAIS

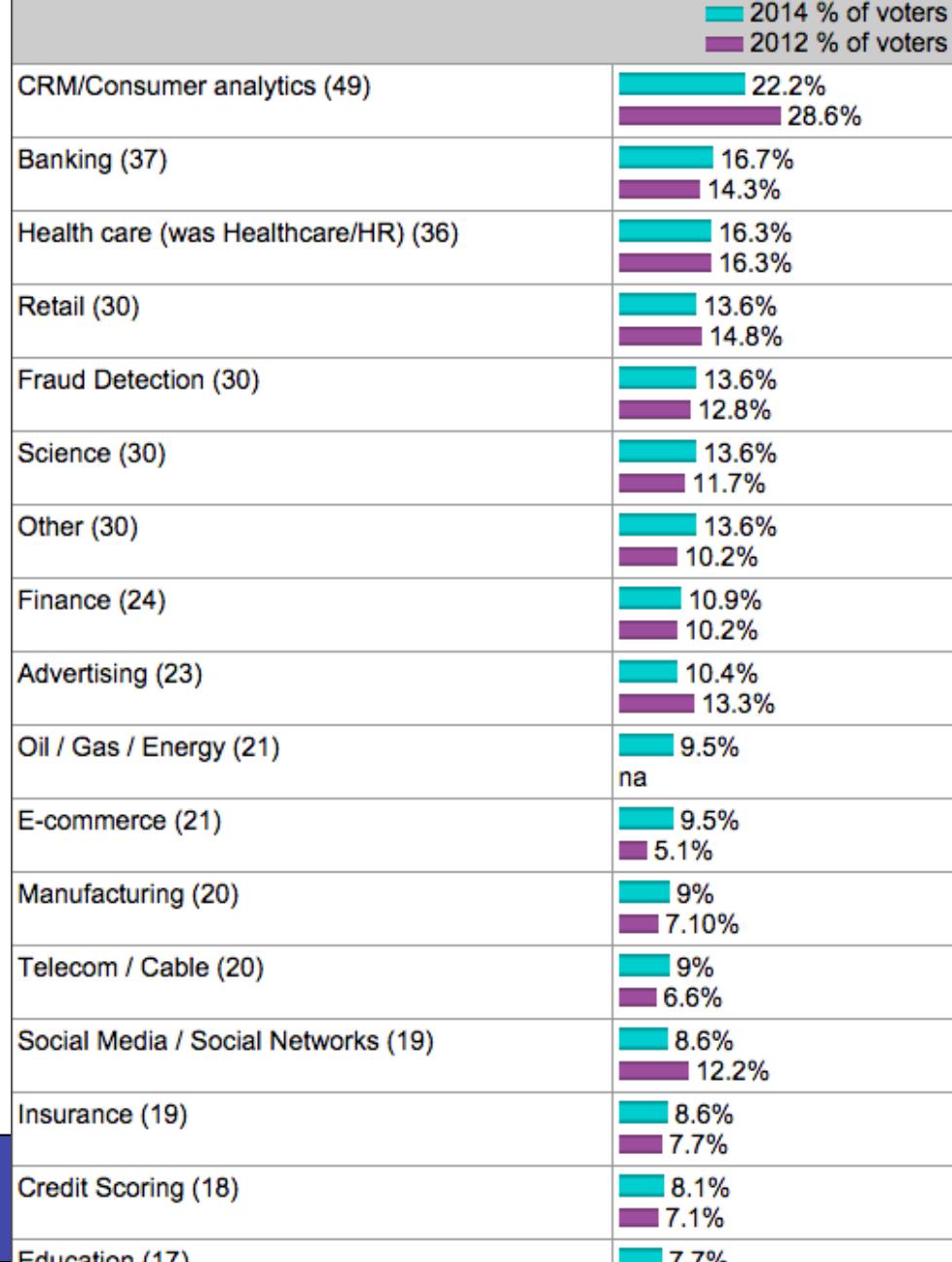
- « Quel est leur profil ? »
- « Quels autres produits les intéresseront ? »
- « Quand seront ils intéressés »

Les profils de clientèle à découvrir sont en général des profils complexes en opposition à des profils devinables par statistiques descriptives

Utilisation du datamining

<http://www.kdnuggets.com/polls/2014/industries-applied-analytics-data-mining-data-science.html>

Industries / Fields where you applied Analytics, Data Mining, Data Science in 2014? [221 voters]



Datamining et CRM

(gestion de la relation client)

19

- Mieux connaître le client
 - Pour mieux le servir
 - Pour augmenter sa satisfaction
 - Pour augmenter sa fidélité
 - Il est plus couteux d'acquérir un client que de le conserver
- La connaissance du client est encore plus utile dans le secteur tertiaire
 - Les produits se ressemblent entre établissements
 - Le prix n'est pas toujours déterminant
 - Ce sont surtout le service et la relation avec le client qui font la différence

Exemple du *credit scoring*

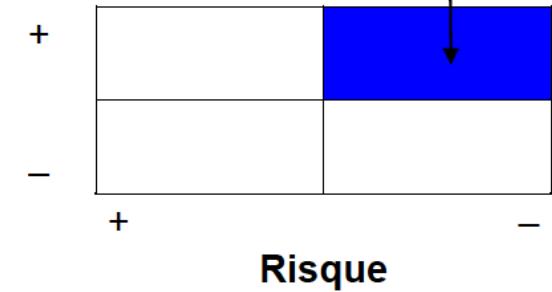
- ▶ Objectifs de la banque :
 - ▶ vendre plus
 - ▶ en maîtrisant les risques
 - ▶ en utilisant les bons canaux au bon moment
- ▶ Le crédit à la consommation
 - ▶ un produit standard
 - ▶ concurrence des sociétés spécialisées sur le lieu de vente (Cetelem...)
 - ▶ quand la banque a connaissance du projet du client, il est déjà trop tard

Conclusion :

il faut être pro-actif
détecter les besoins des clients et leur tendance à emprunter

> Faire des propositions commerciales aux bons clients, avant qu'ils n'en fassent la demande

Appétence



Le data mining dans la banque

- ▶ Naissance du score de risque en 1941 (David Durand)
- ▶ Multiples techniques appliquées à la banque de détail et la banque d'entreprise
- ▶ Surtout la banque de particuliers :
 - ▶ montants unitaires modérés
 - ▶ grand nombre de dossiers
 - ▶ dossiers relativement standards
- ▶ Essor dû à :
 - ▶ développement des nouvelles technologies
 - ▶ nouvelles attentes de qualité de service des clients
 - ▶ concurrence des nouveaux entrants (assureurs, grande distribution) et des sociétés de crédit
 - ▶ pression mondiale pour une plus grande rentabilité
 - ▶ surtout : ratio de solvabilité Bâle 2

Le data mining dans l'assurance de risque

- ▶ Des produits obligatoires (automobile, habitation) :
 - ▶ soit prendre un client à un concurrent
 - ▶ soit faire monter en gamme un client que l'on détient déjà
- ▶ D'où les sujets dominants :
 - ▶ attrition
 - ▶ ventes croisées (*cross-selling*)
 - ▶ montées en gamme (*up-selling*)
- ▶ Besoin de décisionnel dû à :
 - ▶ concurrence des nouveaux entrants (bancassurance)
 - ▶ bases clients des assureurs traditionnels mal organisées :
 - ▶ compartimentées par agent général
 - ▶ ou structurées par contrat et non par client

Le data mining dans la téléphonie

- ▶ Deux événements :
 - ▶ ouverture du monopole de France Télécom
 - ▶ arrivée à saturation du marché de la téléphonie mobile
- ▶ D'où les sujets dominants dans la téléphonie :
 - ▶ score d'attrition (*churn* = changement d'opérateur)
 - ▶ optimisation des campagnes marketing
 - ▶ *text mining* (pour analyser les lettres de réclamation)
- ▶ Problème du *churn* :
 - ▶ coût d'acquisition moyen en téléphonie mobile : 250 euros
 - ▶ plus d'un million d'utilisateurs changent chaque année d'opérateur
 - ▶ la loi Chatel (juin 2008) facilite le changement d'opérateur en diminuant le coût pour ceux qui ont dépassé 12 mois chez l'opérateur
 - ▶ la portabilité du numéro facilite le churn

Le data mining dans le commerce

- ▶ **Vente Par Correspondance**
 - ▶ utilise depuis longtemps des scores d'appétence
 - ▶ pour optimiser ses ciblages et en réduire les coûts
 - ▶ des centaines de millions de documents envoyés par an
- ▶ **e-commerce**
 - ▶ personnalisation des pages du site web de l'entreprise, en fonction du profil de chaque internaute
 - ▶ optimisation de la navigation sur un site web
- ▶ **Grande distribution**
 - ▶ analyse du ticket de caisse
 - ▶ détermination des meilleures implantations (géomarketing)

Autres exemples

- ▶ De l'∞ petit (génomique) à l'∞ grand (astrophysique pour le classement en étoile ou galaxie)
- ▶ Du plus quotidien (reconnaissance de l'écriture manuscrite sur les enveloppes) au moins quotidien (aide au pilotage aéronautique)
- ▶ Du plus ouvert (e-commerce) au plus sécuritaire (détection de la fraude dans la téléphonie mobile ou les cartes bancaires)
- ▶ Du plus industriel (contrôle qualité pour la recherche des facteurs expliquant les défauts de la production) au plus théorique (sciences humaines, biologie...)
- ▶ Du plus alimentaire (agronomie et agroalimentaire) au plus divertissant (prévisions d'audience TV)

Exemples médicaux

- ▶ Mettre en évidence des facteurs de risque ou de rémission dans certaines maladies (infarctus et des cancers) – Choisir le traitement le plus approprié – Ne pas prodiguer des soins inutiles
- ▶ Déterminer des segments de patients susceptibles d'être soumis à des protocoles thérapeutiques déterminés, chaque segment regroupant tous les patients réagissant identiquement
- ▶ Décryptage du génome
- ▶ Tests de médicaments, de cosmétiques
 - ▶ Prédire les effets sur la peau humaine de nouveaux cosmétiques, en limitant le nombre de tests sur les animaux

Clustering (Segmentation)

Problématique

Objectif = structuration des données

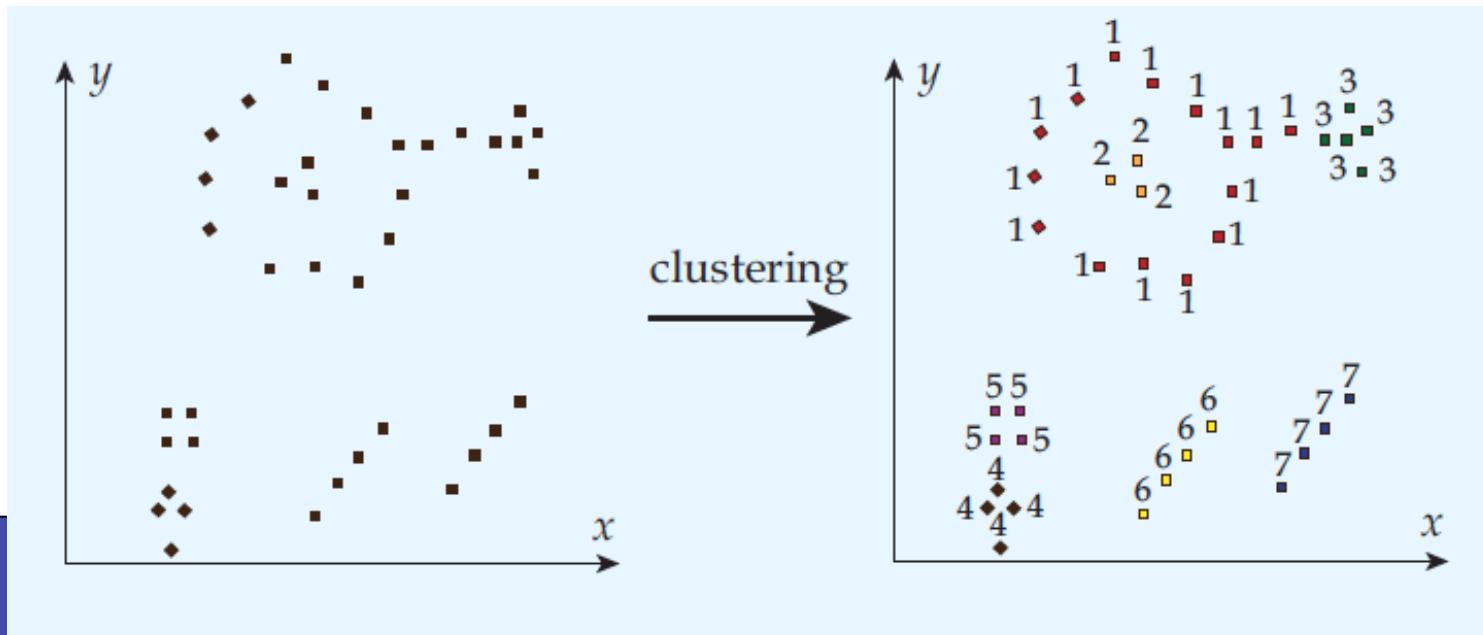


Clustering (en anglais) = Classification (en français) non supervisée

On cherche à regrouper les points proches en groupe ou classes ou cluster

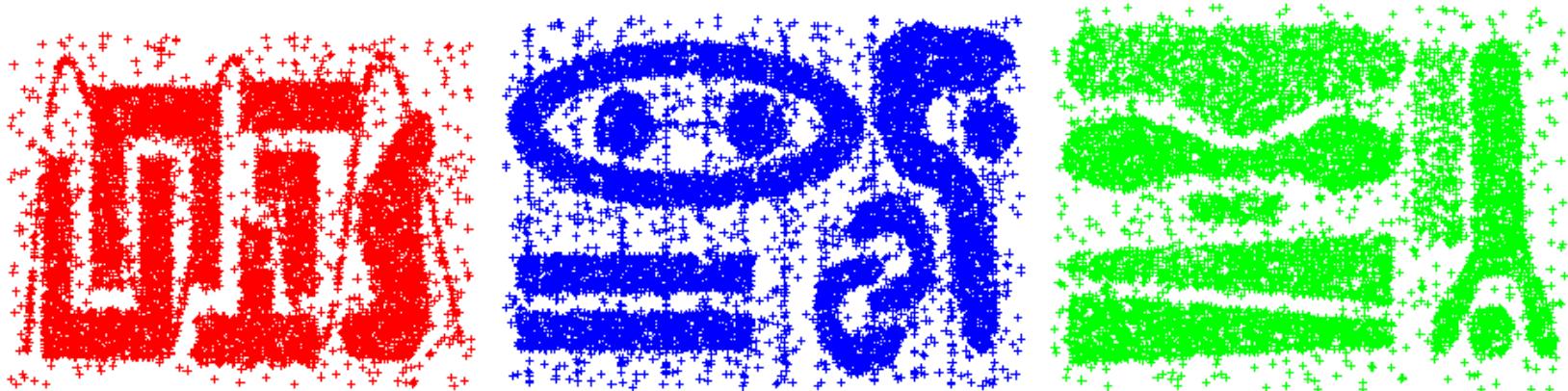
Les points ou objets différents appartiennent à des groupes différents

Les classes peuvent être assez bien définies



Problématique

Les classes peuvent aussi être assez imbriquées,
avoir des formes bizarres,
ou pire



Mais surtout ne pas être en 2 dimensions

Problématique

Soient N instances de données à k attributs,

Trouver un partitionnement en c clusters (groupes) ayant un sens
(Similitude)

Affectation automatique de “labels” aux clusters

c peut être donné, ou “découvert”

Plus difficile que la classification car les classes ne sont pas connues à l'avance (non supervisé)

Attributs

- Numériques (distance bien définie)
- Enumératifs ou mixtes (distance difficile à définir)

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.

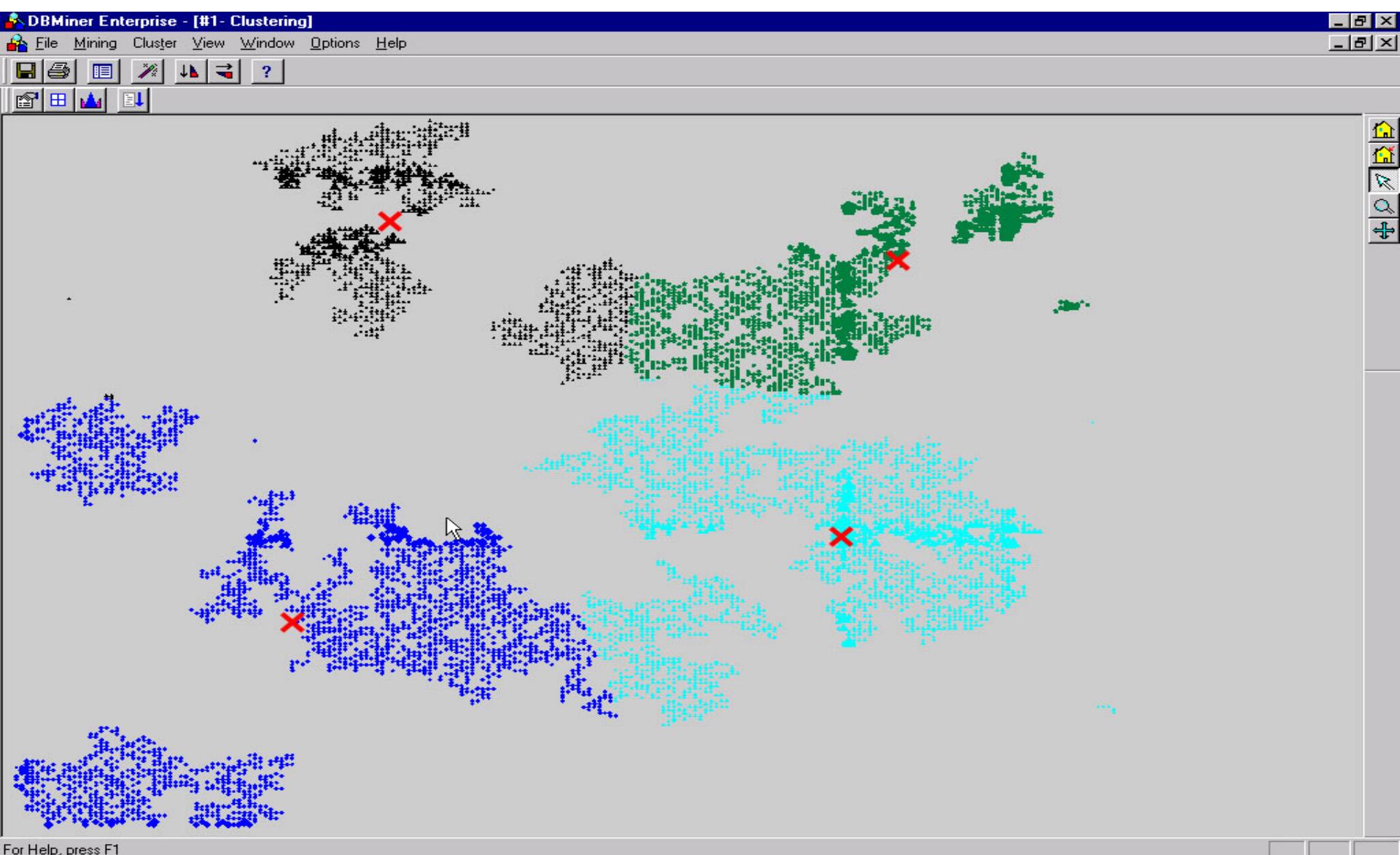
Assurance: identification de groupes d'assurés distincts associés à un nombre important de déclarations.

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

Exemple: segmentation de marchés



Qualité d'un clustering

Une bonne méthode de clustering produira des clusters d'excellente qualité avec :

- Similarité intra-classe importante
- Similarité inter-classe faible

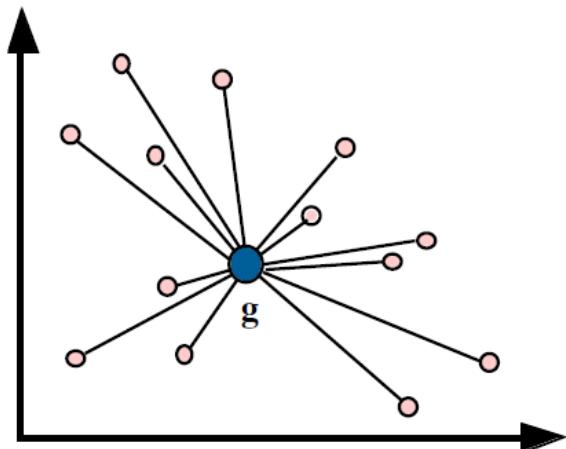
La qualité d'un clustering dépend de :

- La mesure de similarité utilisée
- L'implémentation de la mesure de similarité

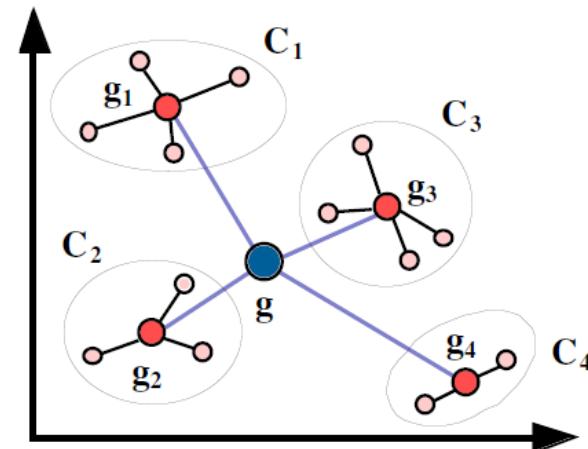
La qualité d'une méthode de clustering est évaluée par son habileté à découvrir certains ou tous les “patterns” cachés.

Objectifs du clustering

Maximiser les distances inter-clusters



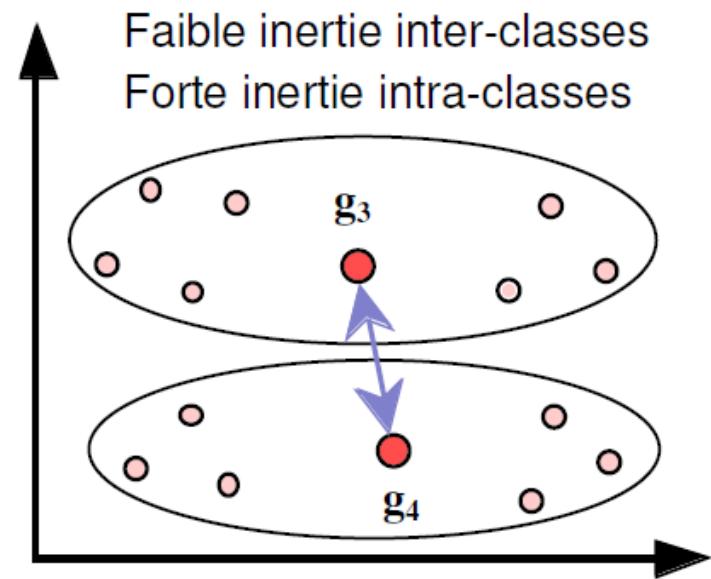
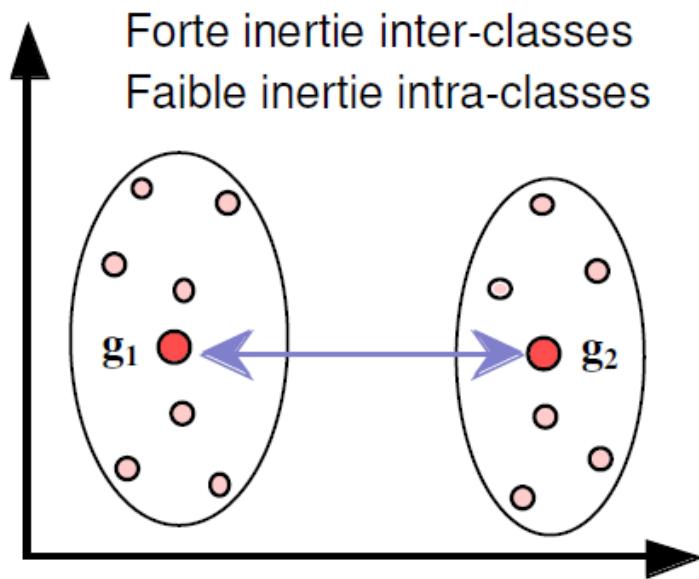
Minimiser les distances intra-cluster



$$\text{Inertie totale des points} = \text{Inertie Intra} + \text{Inter}$$

Bisson 2001

Objectifs du clustering



Notion de proximité

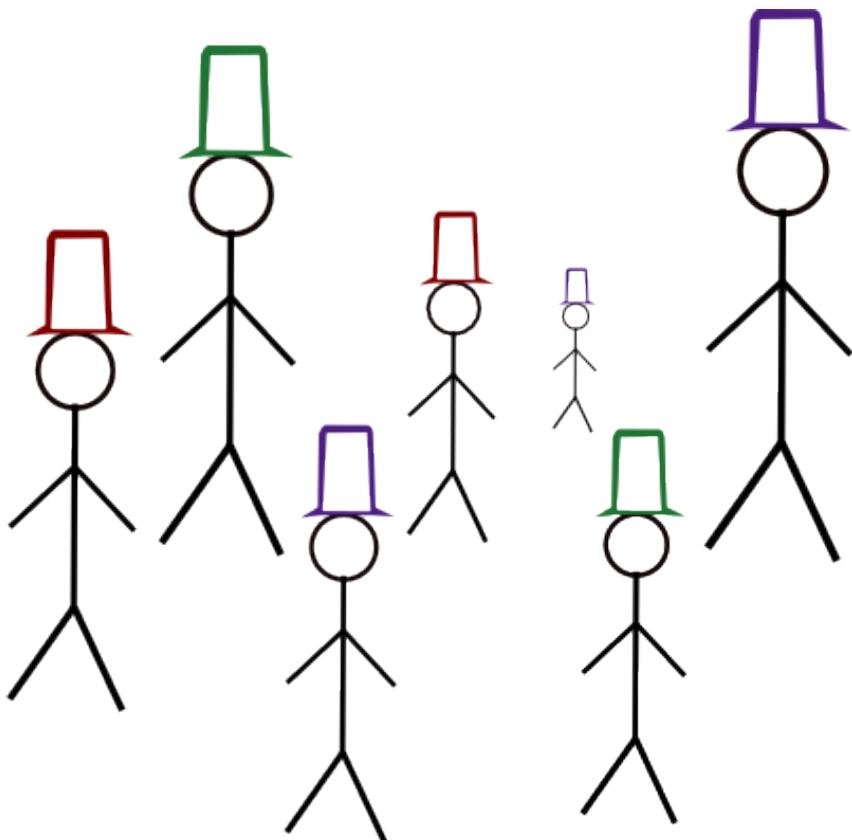
Vocabulaire

Mesure de dissimilarité (DM) : plus la mesure est faible plus les points sont similaires (~ distance)

Mesure de similarité (SM) : plus la mesure est grande, plus les points sont similaires

$$DM = \text{borne} - SM$$

Mesure de la similarité



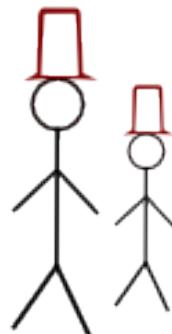
Il n'y a pas de définition unique de la similarité entre objets

- Différentes mesures de distances $d(x,y)$

La définition de la similarité entre objets dépend de :

- Le type des données considérées
- Le type de similarité recherchée

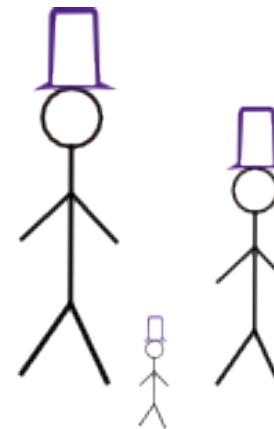
Mesure de similarité



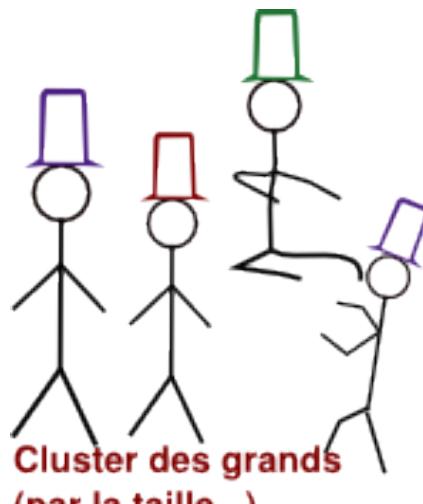
Cluster des rouges



Cluster des verts



Cluster des violets



Cluster des grands
(par la taille...)



Cluster des petits
et qui le vivent bien!

Choix de la distance

Propriétés d'une distance :

1. $d(x, y) \geq 0$
2. $d(x, y) = 0$ iff $x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Définir une distance sur chacun des champs

Champs numériques : $d(x, y) = |x - y|$, $d(x, y) = |x - y| / d_{\max}$ (distance normalisée).

Exemple : Age, taille, poids, ...

Distance – Données numériques

Combiner les distances : Soient $x=(x_1, \dots, x_n)$ et $y=(y_1, \dots, y_n)$

Exemples numériques :

Distance euclidienne :

$$d(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Distance de Manhattan :

$$d(x,y) = \sum_{i=1}^n |x_i - y_i|$$

Distance de Minkowski :

$$d(x,y) = \sqrt[q]{\sum_{i=1}^n |x_i - y_i|^q}$$

$q=1$: distance de Manhattan.

$q=2$: distance euclidienne

Choix de la distance

Champs discrets :

- Données binaires : $d(0,0)=d(1,1)=0$, $d(0,1)=d(1,0)=1$
- Donnée énumératives : distance nulle si les valeurs sont égales et 1 sinon.
- Donnée énumératives ordonnées : idem. On peut définir une distance utilisant la relation d'ordre.

Données de types complexes : textes, images, données génétiques, ...

Distance – Données binaires

**Table de contingence
(dissimilité)**

		Object <i>j</i>		<i>sum</i>
		1	0	
Object <i>i</i>	1	<i>a</i>	<i>b</i>	<i>a+b</i>
	0	<i>c</i>	<i>d</i>	<i>c+d</i>
		<i>sum</i>	<i>a+c</i>	<i>b+d</i>
				<i>p</i>

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

- Exemple $oi=(1,1,0,1,0)$ et $oj=(1,0,0,0,1)$
 $a=1$, $b=2$, $c=1$, $d=2$

Distance – Données binaires

**Table de contingence
(dissimilité)**

		Object <i>j</i>		<i>sum</i>
		1	0	
Object <i>i</i>	1	<i>a</i>	<i>b</i>	<i>a+b</i>
	0	<i>c</i>	<i>d</i>	<i>c+d</i>
<i>sum</i>	<i>a+c</i>	<i>b+d</i>	<i>p</i>	

- **Coefficient de correspondance simple** (similarité invariante, si la variable binaire est **symétrique**) :

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

- **Coefficient de Jaccard** (similarité non invariante, si la variable binaire est **asymétrique**):

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

Variables binaires (I)

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

Variable asymétrique: 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

Distance – Données binaires

Exemple : dissimilarité entre variables binaires

- Table de patients

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

- 8 attributs, avec
 - Sexe un attribut symétrique, et
 - Les attributs restants sont asymétriques
 - (test VIH, ...)

Distance – Données binaires

Les valeurs Y et P sont initialisées à 1, et la valeur N à 0.

Calculer la distance entre patients, basée sur le coefficient de Jaccard.

Jack	M	1	0	1	0	0	0
Mary	F	1	0	1	0	1	0
Jim	M	1	1	0	0	0	0

Jack/Mary	1	0
1	A=2	B=0
0	C=1	D=3

Jack/Jim	1	0
1	A=1	B=1
0	C=1	D=3

Jim, Mary	1	0
1	A=1	B=1
0	C=2	D=2

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

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

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

Distance – Données énumératives

Généralisation des variables binaires, avec plus de 2 états, e.g., rouge, jaune, bleu, vert

Méthode 1: correspondance simple

- m: # de correspondances, p: # total de variables

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

Variables Ordinales

Une variable ordinaire 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 $r_{if} \in \{1, \dots, M_f\}$
- 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

Besoin de standardiser les données

1. Le Z-score

1. Calculer l'écart moyen absolu

Où mf est la valeur moyenne de la série sur l'attribut.

2. Calcul de la mesure standardisée

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

$$m(\text{age})=60, S(\text{age})=5$$

$$M(\text{salaire})=11074, S(\text{salaire})=148$$

$$d(p1, p2) = 4,675$$

$$d(p1, p3) = 2,324$$

Conclusion: p1 ressemble plus à p3 qu'à p2 😊

En Présence de Variables de différents Types (données mixtes)

51

- 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

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

- f est binaire ou nominale:
 $d_{ij}^{(f)} = 0$ 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
 - ensuite traiter z_{if} comme une variable de type intervalle

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

Distance – Données mixtes

Exemple : (Age, Propriétaire résidence principale, montant des mensualités en cours)

$$x=(30,1,1000), y=(40,0,2200), z=(45,1,4000)$$

$$d(x,y)=\sqrt{ (10/15)^2 + 1^2 + (1200/3000)^2 } = 1.27$$

$$d(x,z)= \sqrt{ (15/15)^2 + 0^2 + (3000/3000)^2 } = 1.41$$

$$d(y,z)= \sqrt{ (5/15)^2 + 1^2 + (1800/3000)^2 } = 1.21$$

plus proche voisin de x = y

Distances normalisées.

Sommation : $d(x,y)=d_1(x_1,y_1) + \dots + d_n(x_n,y_n)$

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

Algorithmes de grille: basés sur un structure à multi-niveaux de granularité

Algorithmes à modèles: Un modèle est supposé pour chaque cluster ensuite vérifier chaque modèle sur chaque groupe pour choisir le meilleur

Méthodes de clustering - Caractéristiques

Extensibilité

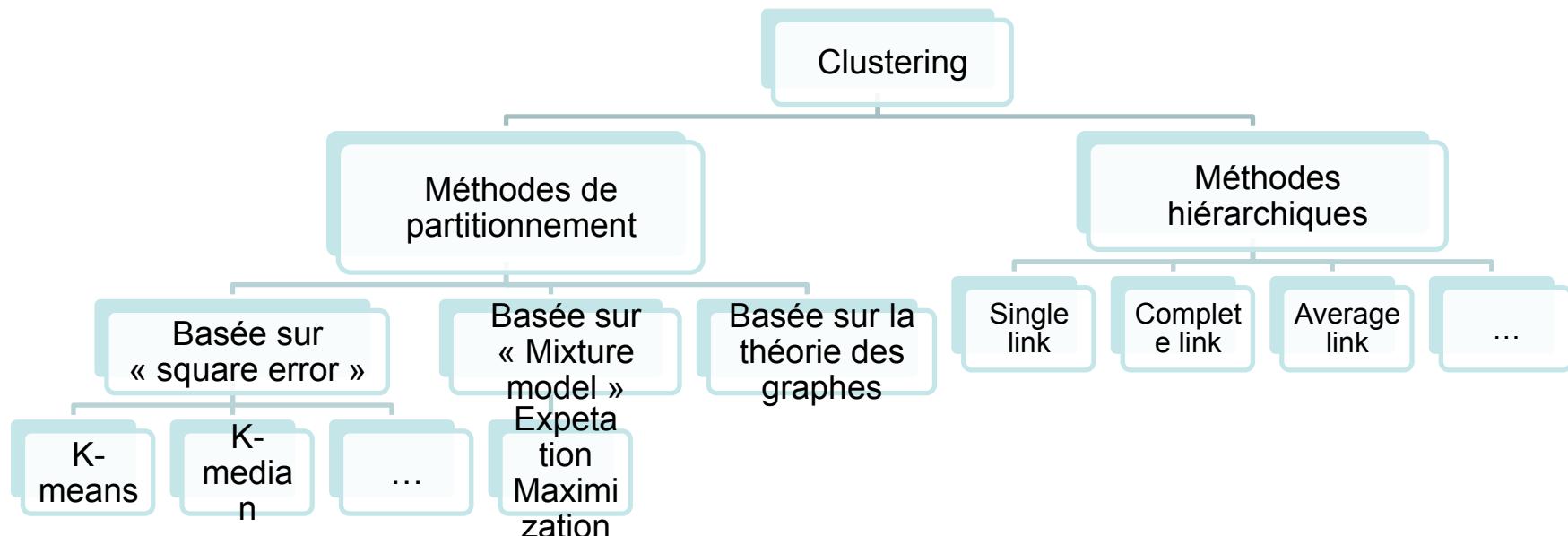
Habilité à traiter différents types de données

Découverte de clusters de différents formes

Connaissances requises (paramètres de l'algorithme)

Habilité à traiter les données bruitées et isolées.

Taxonomie



Algorithmes à partitionnement

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

Algorithme des k-moyennes (K-means)

Entrée : un échantillon de m enregistrements x_1, \dots, x_m

Paramètre : Fixer le nombre de cluster K

1. Choisir k centres initiaux c_1, \dots, c_k
2. Répartir chacun des m enregistrements dans le groupe i dont le centre ci est le plus proche.
3. Si aucun élément ne change de groupe alors arrêt et sortir les groupes
4. Calculer les nouveaux centres : pour tout i, c_i est la moyenne des éléments du groupe i (le barycentre).

Aller en 2.

Illustration (les données)

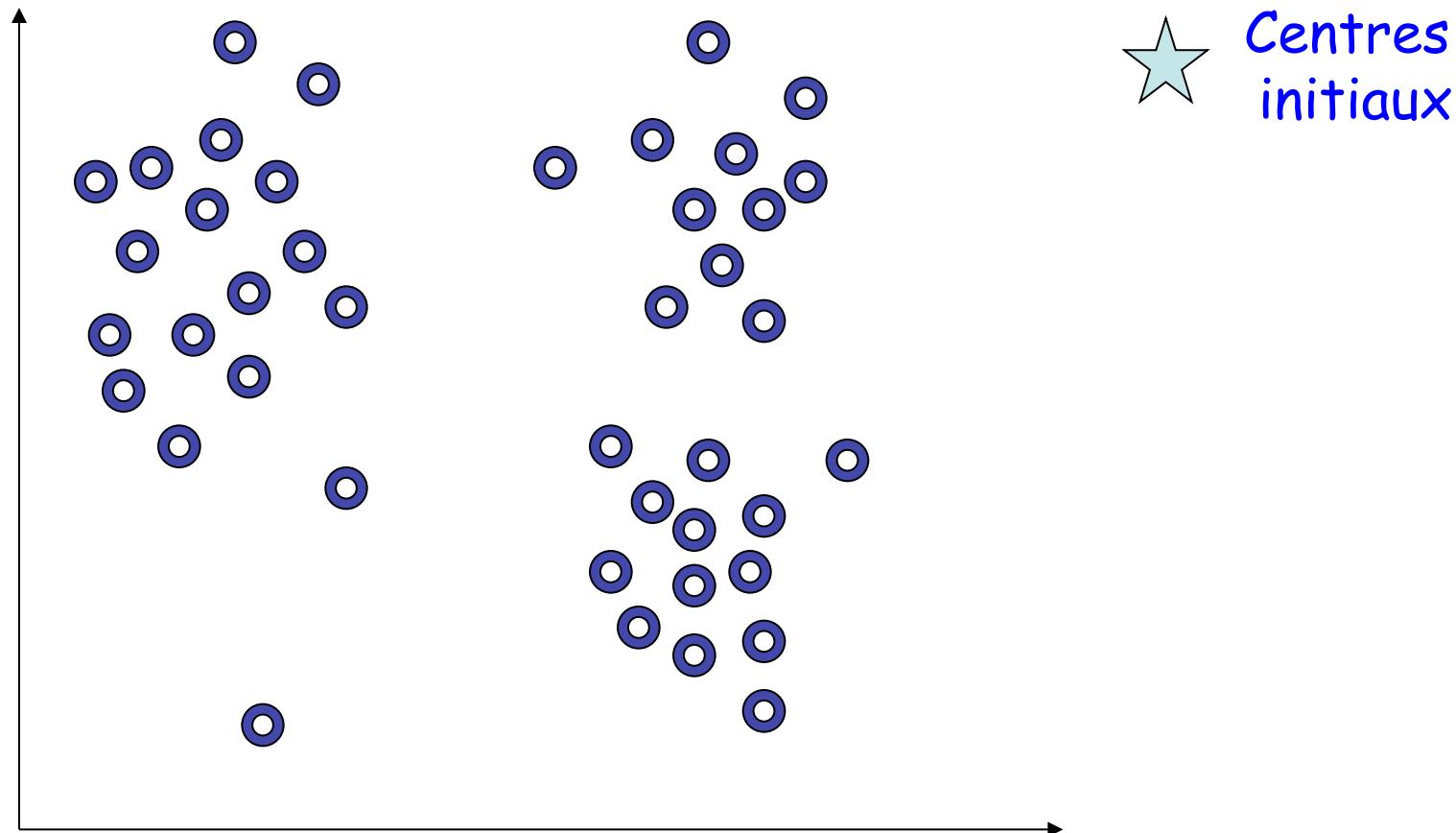


Illustration (Etape 1)

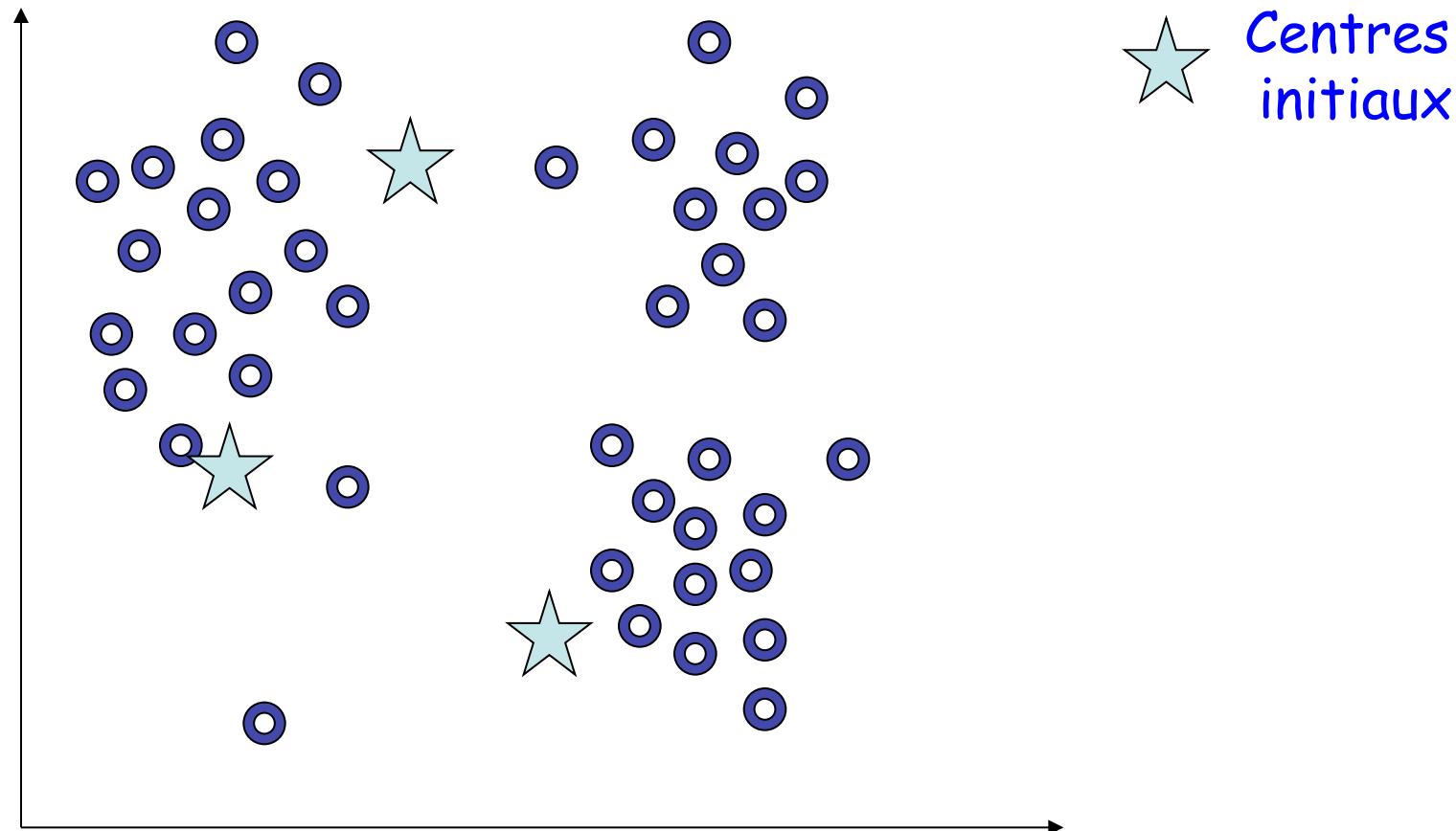


Illustration (Etape 2)

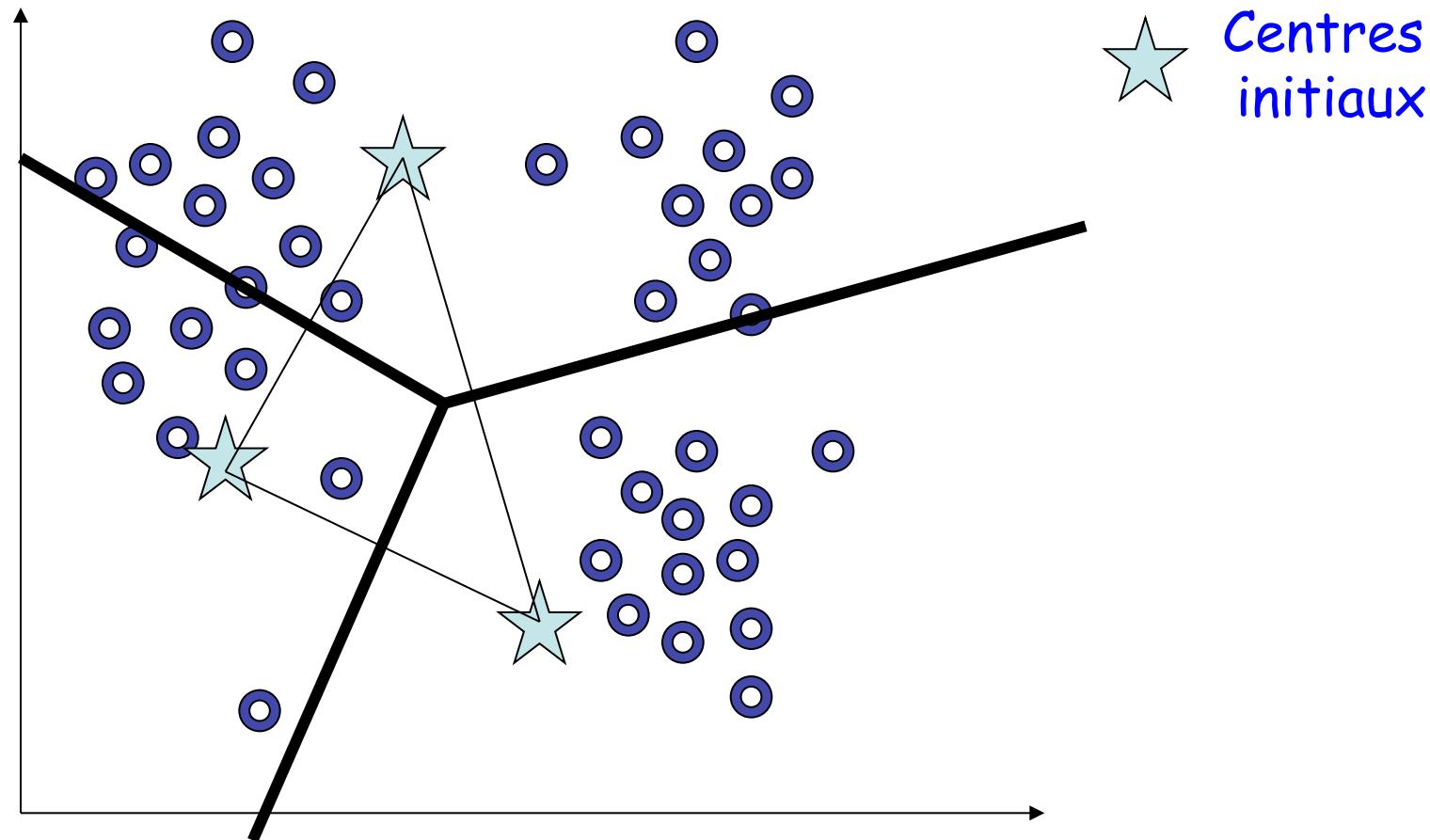


Illustration (Etape 3)

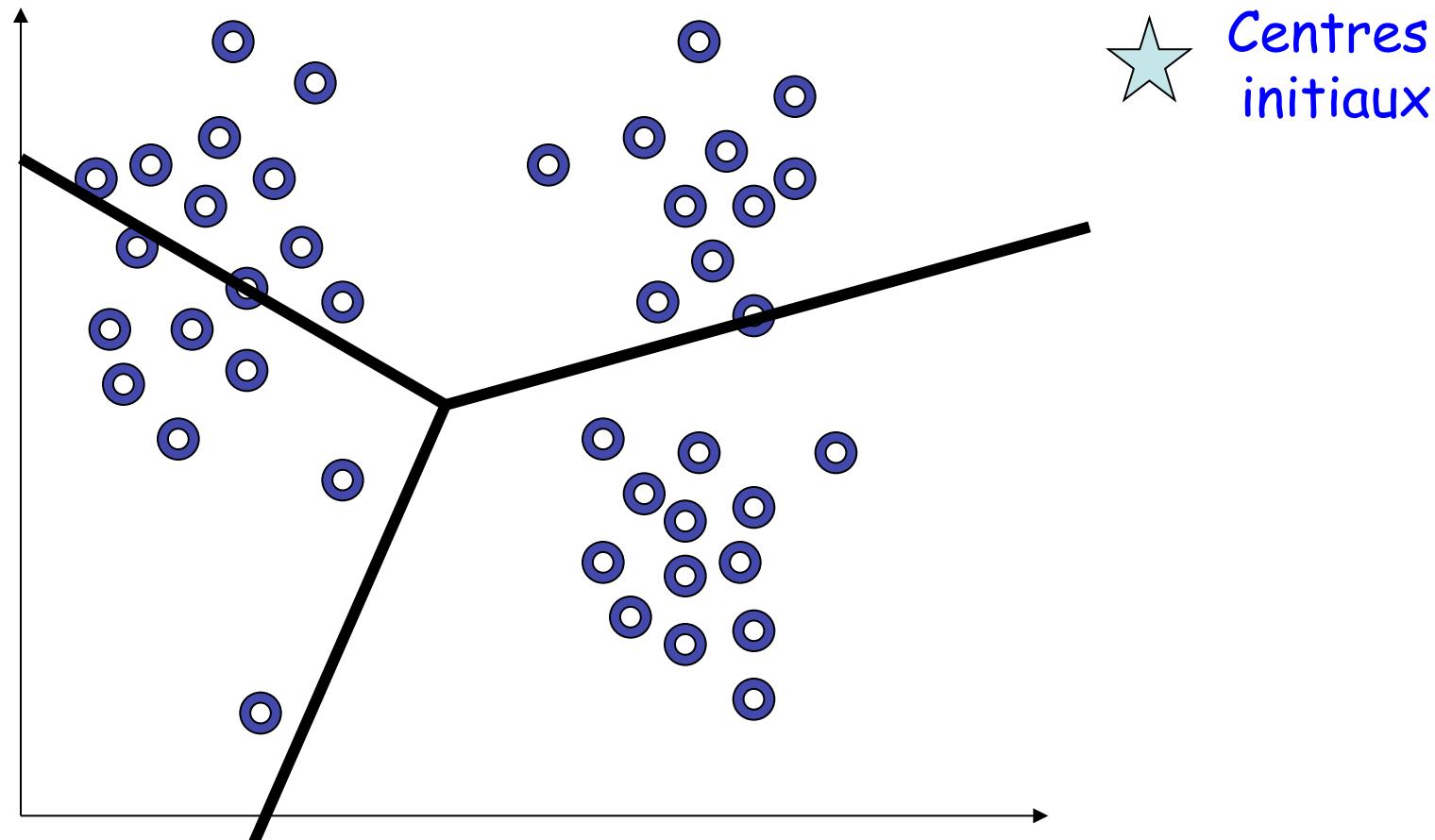


Illustration (Etape 4)

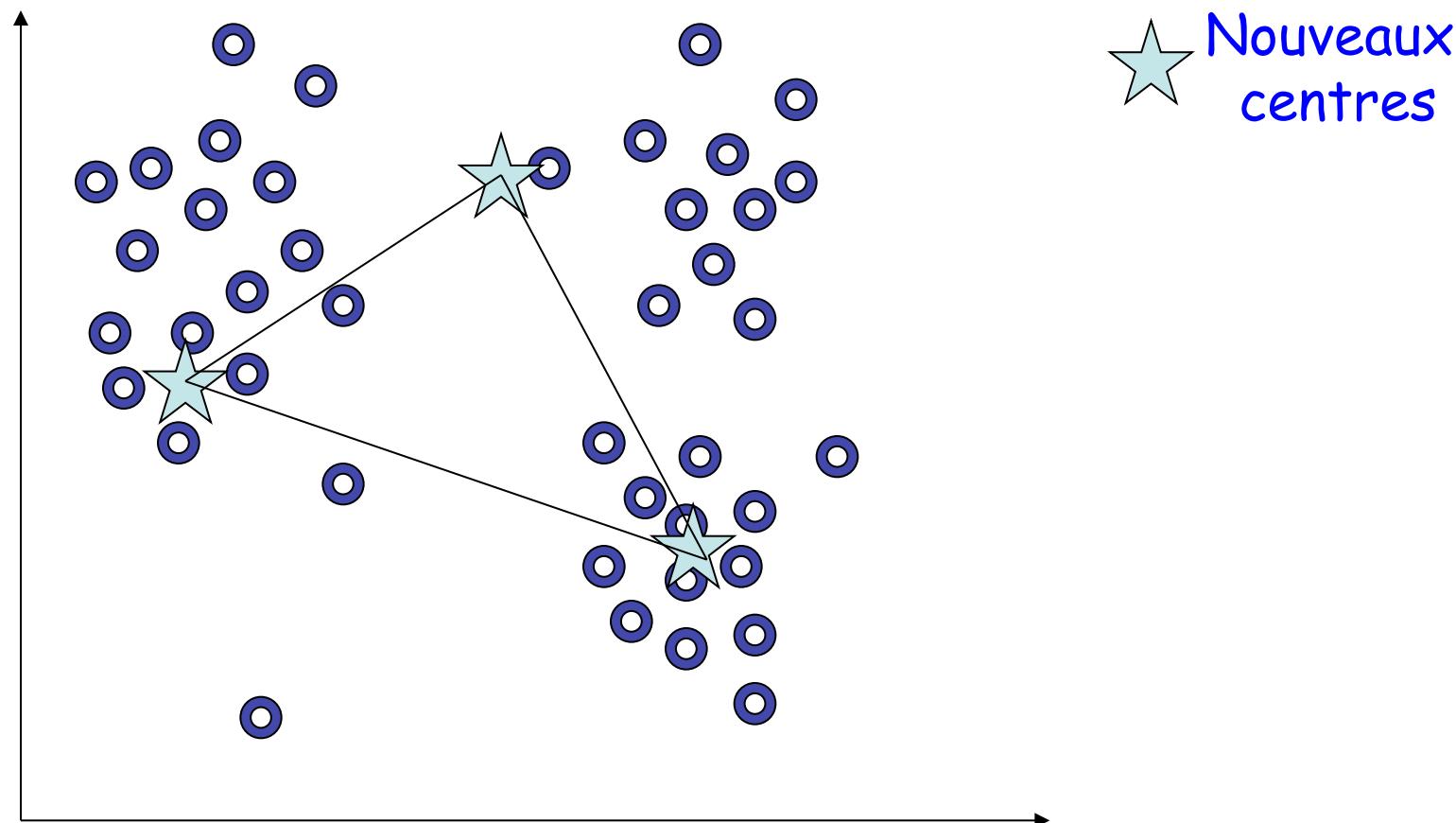


Illustration (Etape 1)

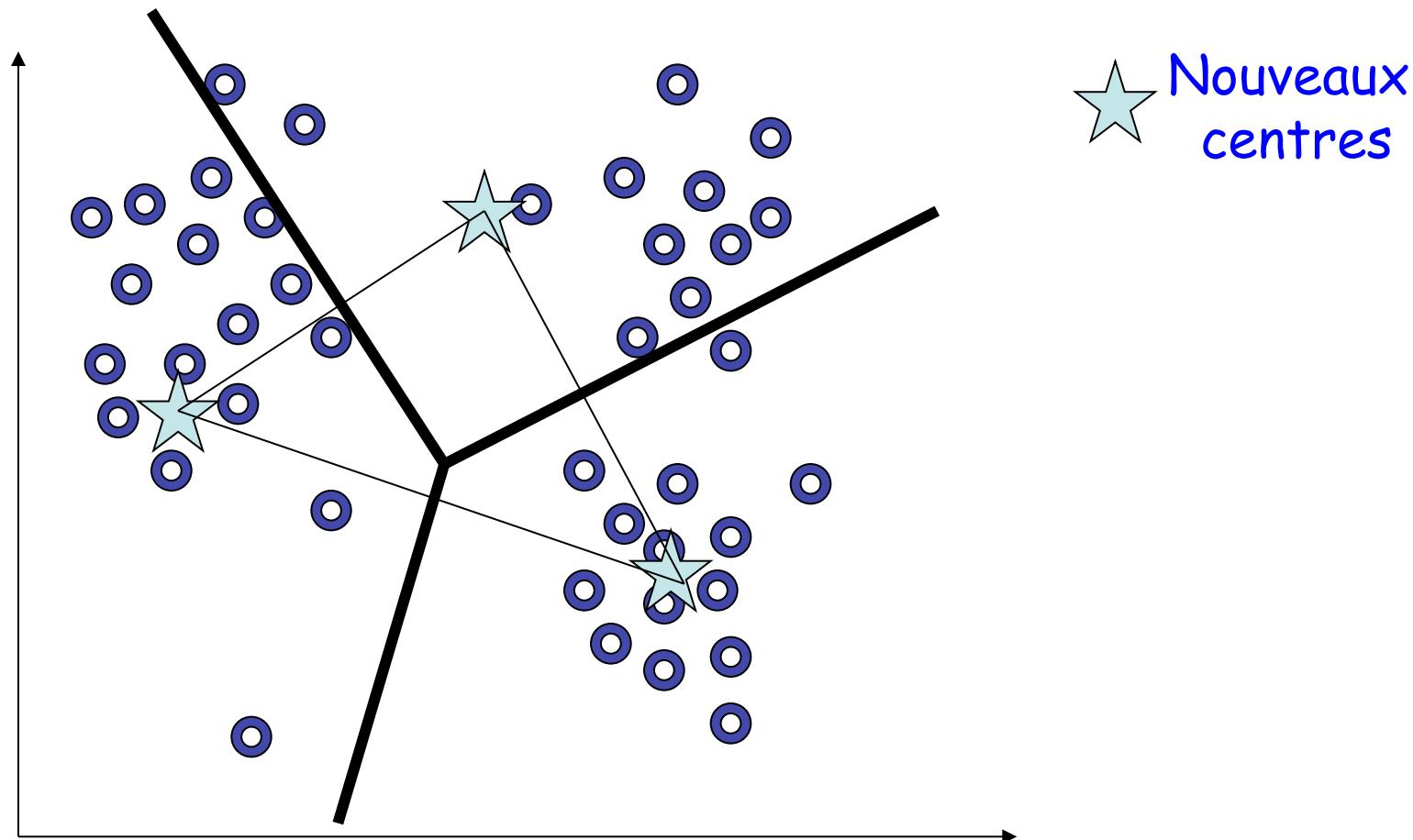
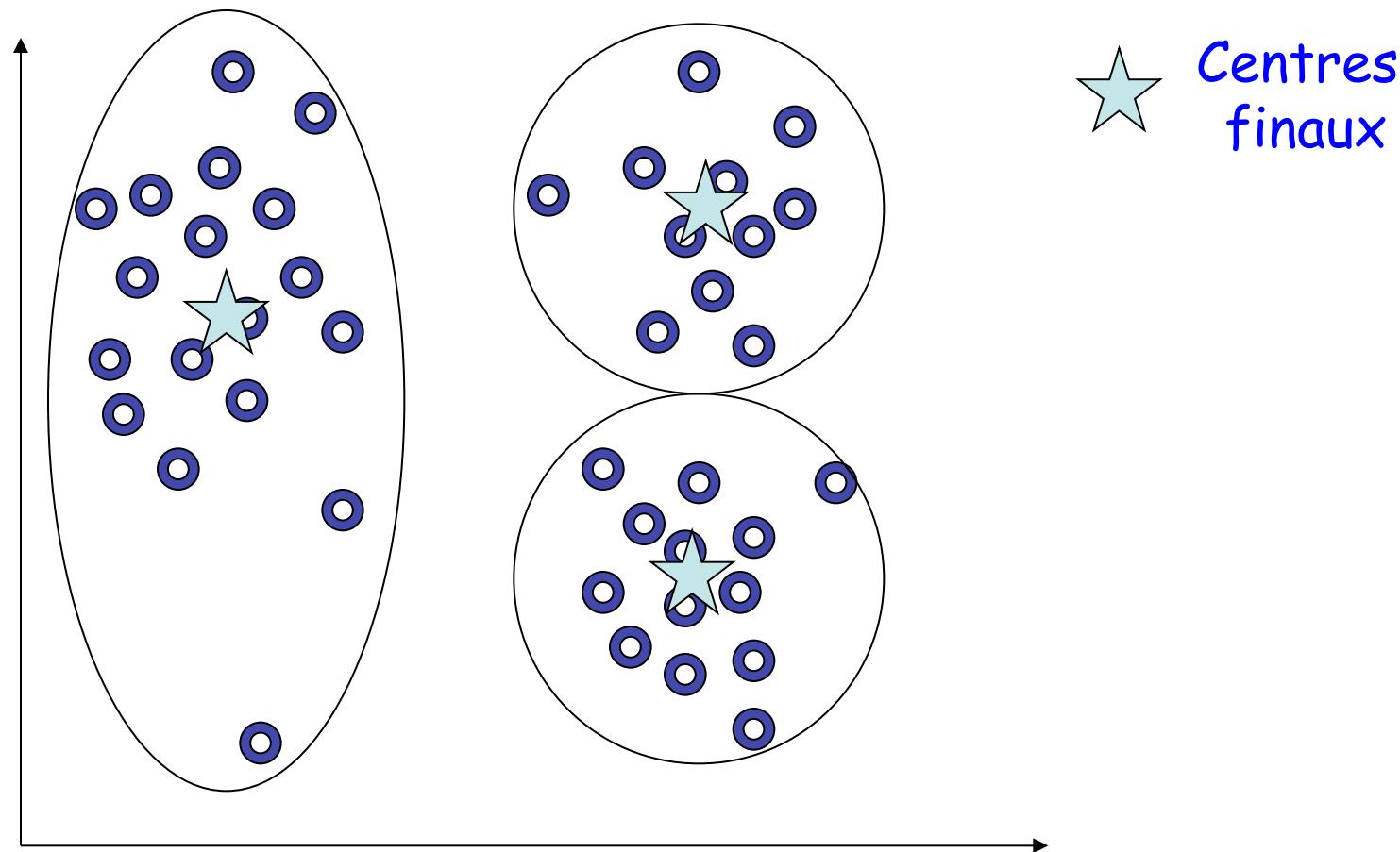


Illustration (3)



Algorithme des k-moyennes : Exemple

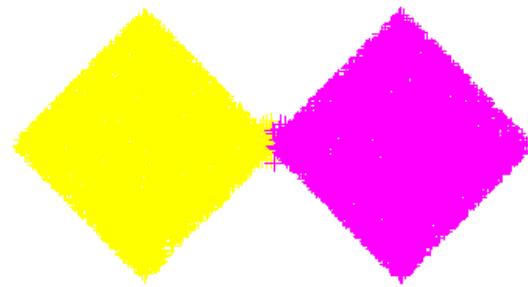
- 8 points A, \dots, H de l'espace euclidien 2D. $k=2$ (2 groupes)
- Tire aléatoirement 2 centres : B et D choisis.

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

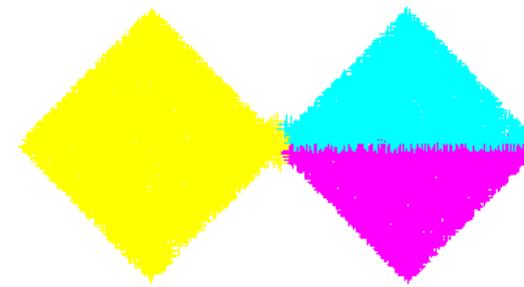
L.

Exemple

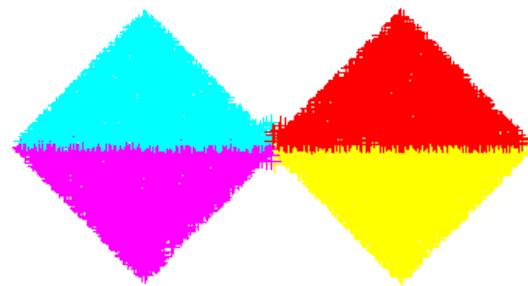
K = 2



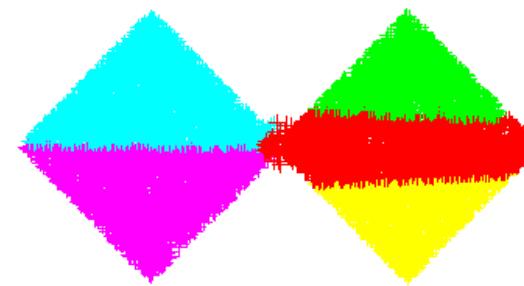
K = 3



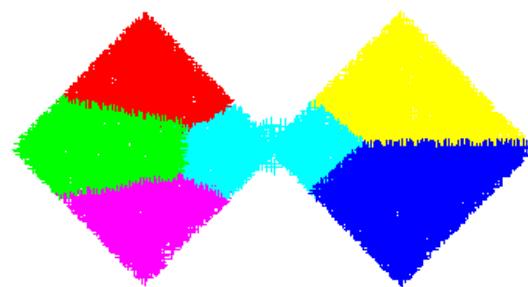
K = 4



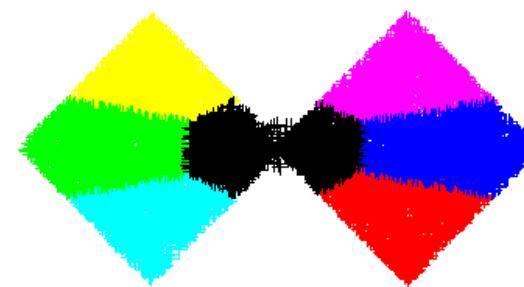
K = 5



K = 6



K = 7



Qualité

Mesurer la qualité du clustering

- Compacité des clusters.
- Séparation des clusters.
- Score de la partition.

Compacité

$$wc(C) = \sum_{k=1}^K wc(C_k) = \sum_{k=1}^K \sum_{x_i \in C_k} d(x_i, c_k)$$

Autre fonction possible :

$$wc(C_k) = \max_i \min_{x_j \in C_k} \{d(x_i, x_j) / x_i \in C_k, x_i \neq x_j\}$$

La plus grande distance minimale entre deux éléments d'un même cluster

Séparation

- Distance entre les centres des clusters :

$$bc = \sum_{1 \leq j < k \leq K} d(r_j, r_k)$$

- Distance entre ensembles :

- Distance minimale.
- Distance maximale.
- Distance moyenne.

Valeur de la partition

Valeur de la partition

Combiner wc (à minimiser) et bc (à maximiser).

Par exemple :

$$\frac{bc}{wc}$$

ou bien :

$$\frac{\alpha bc + \beta wc}{bc + wc}$$

K-moyennes : Avantages

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, $k, t \ll n$.

Produit généralement un optimum local ; un optimum global peut être obtenu en utilisant d'autres techniques telles que : algorithmes génétiques, ...

K-moyennes : Désavantages

Applicable seulement dans le cas où la moyenne des objets est définie

Besoin de spécifier k , le nombre de clusters, a priori

Incapable de traiter les données bruitées (noisy).

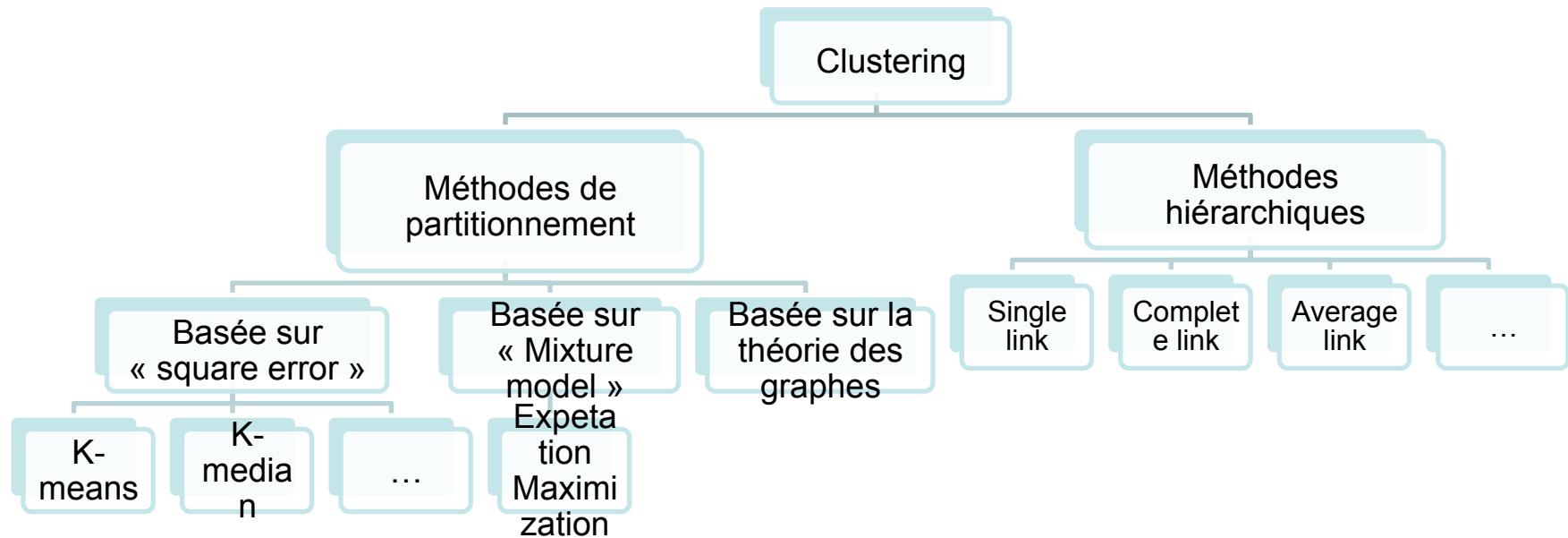
Non adapté pour découvrir des clusters avec structures non-convexes, et des clusters de tailles différentes

Les points isolés sont mal gérés (doivent-ils appartenir obligatoirement à un cluster ?) - probabiliste

K-moyennes : Variantes

- 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)

Taxonomie



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

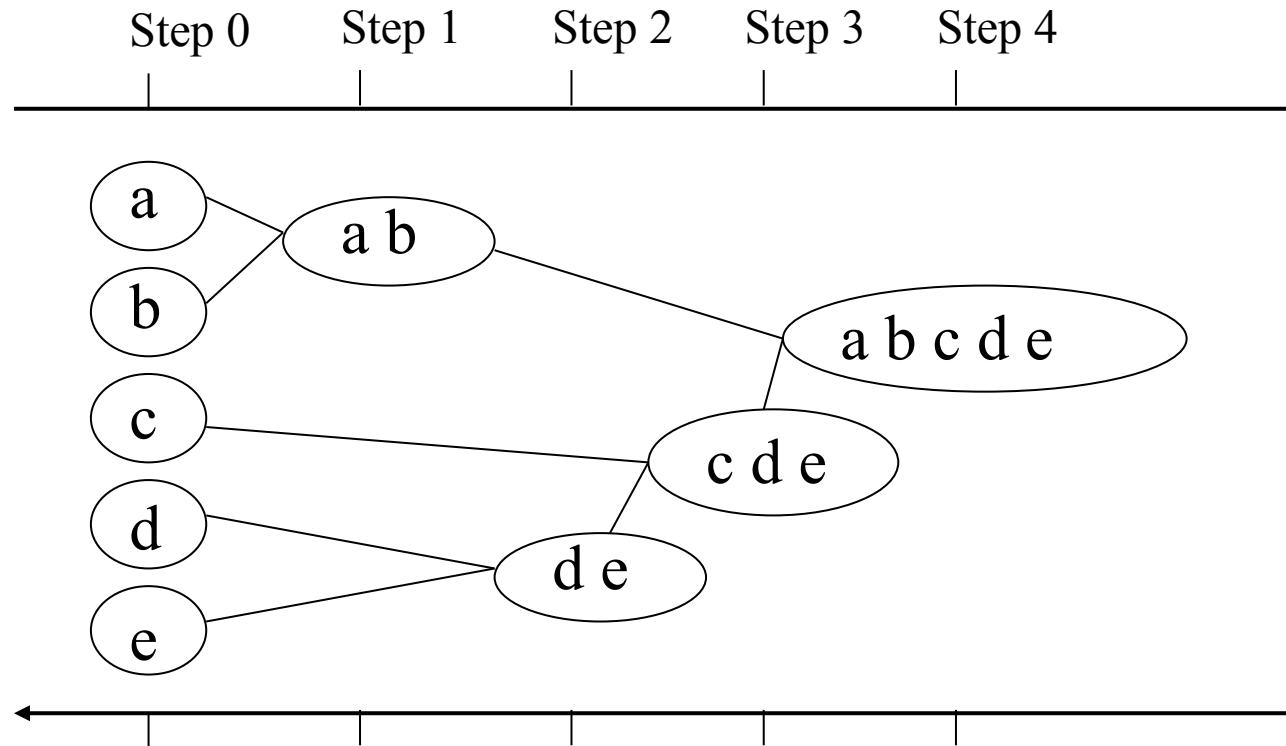
Une condition de terminaison peut être utilisée (ex. Nombre de clusters)

Méthodes hiérarchiques-agglomérative

Entrée : un échantillon de m enregistrements x_1, \dots, x_m

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 : Exemple

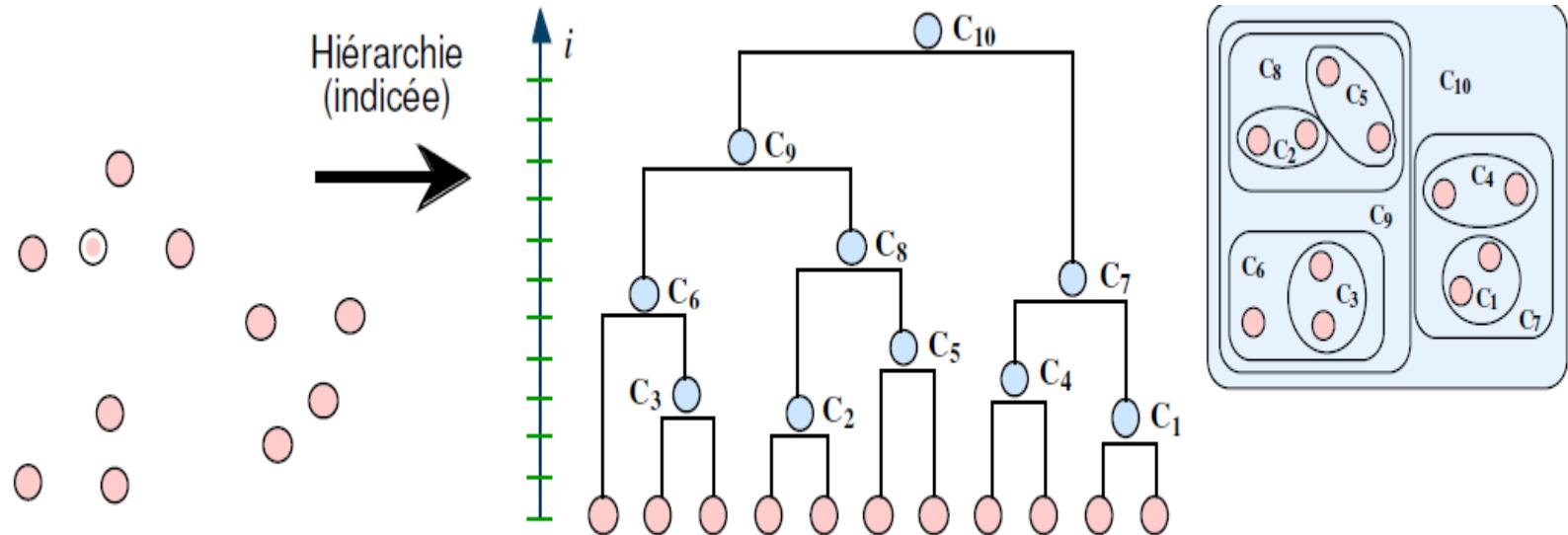


Arbre de clusters - Dendrogramme

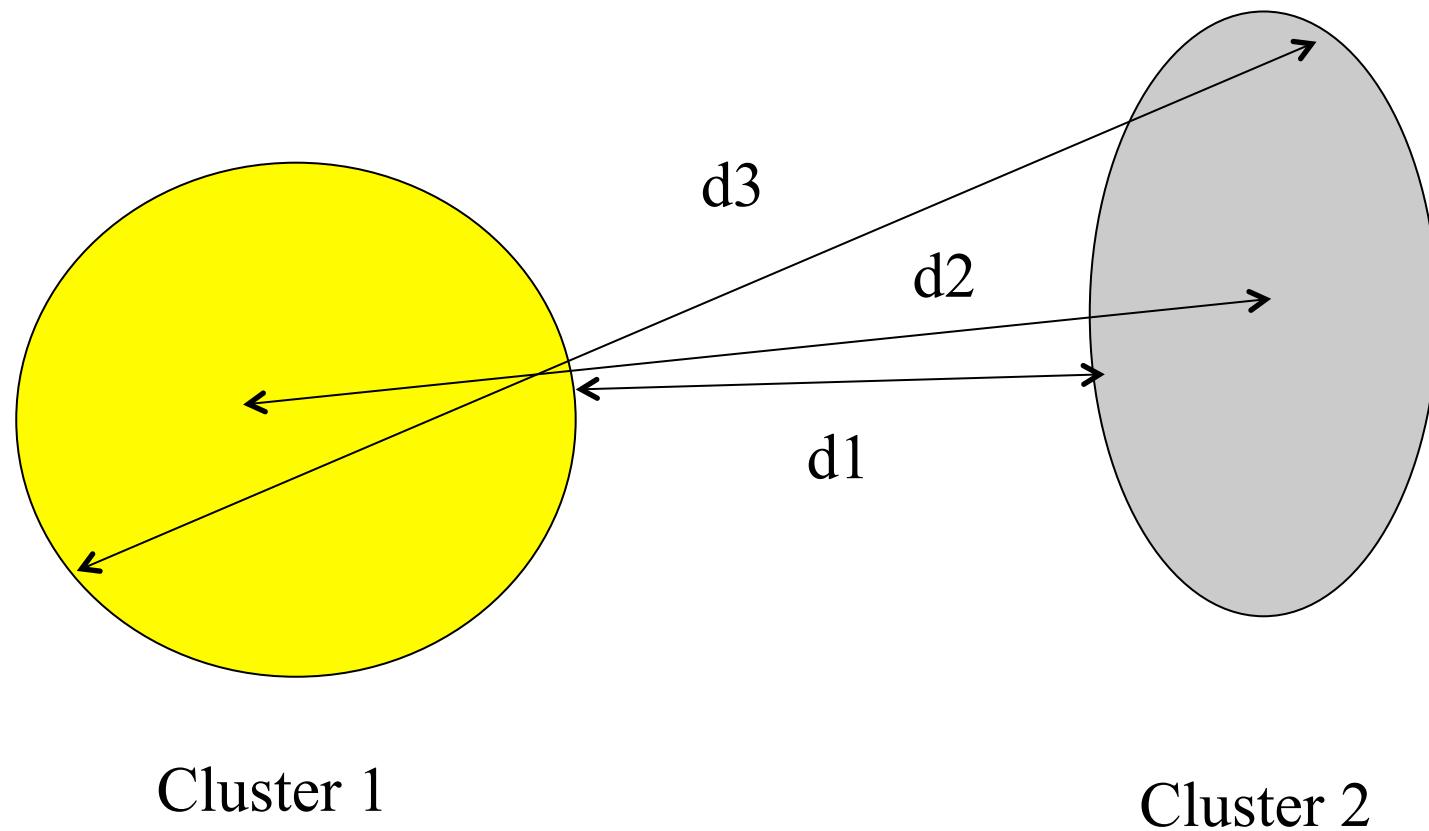
Résultat : Graphe hiérarchique qui peut être coupé à un niveau de dissimilarité pour former une partition.

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

- Les feuilles de l'arbre représentent les objets
- Les nœuds intermédiaires de l'arbre représentent les clusters



Mais quelle distance ?



Méthode agglomérative

Complexité : n^2

Dépend de la distance entre clusters :

- Distance minimale : clusters allongés.
- Distance maximale : clusters de même volume.
- Distance moyenne.
- Distance entre centres de clusters

Distance entre clusters

Distance entre les centres des clusters (Centroid Method)

- tendance à produire des classes de variance proche

Distance (saut) minimale entre toutes les paires de données des 2 clusters (Single Link Method)

$$d(i, j) = \min_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

- tendance à produire des classes générales (par effet de chaînage)
- sensibilité aux individus bruités.

Distance (saut) maximale entre toutes les paires de données des 2 clusters (Complete Link Method)

$$d(i, j) = \max_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

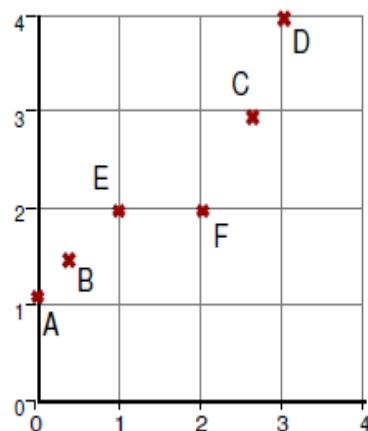
- tendance à produire des classes spécifiques (on ne regroupe que des classes très proches)
- sensibilité aux individus bruités.

Distance (saut) moyenne entre toutes la paires d'enregistrements (Average Linkage)

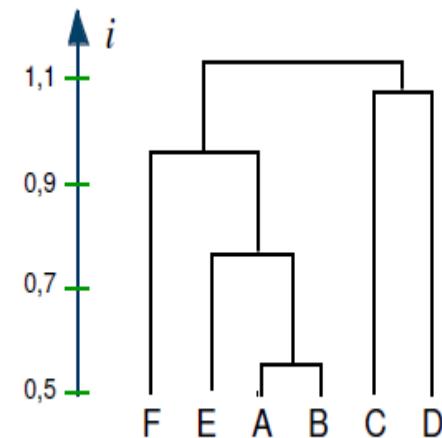
$$d(i, j) = \text{avg}_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

pas les mêmes résultats selon la métrique utilisée ...

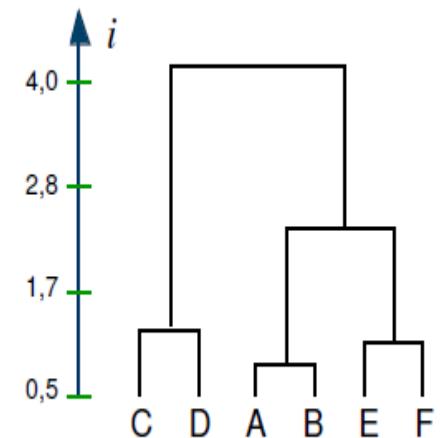
Données (métrique : dist. Eucl.)



Saut minimal



Saut maximal



Méthodes hiérarchiques : Avantages

Conceptuellement simple

Propriétés théoriques sont bien connues

Quand les clusters sont groupés, la décision est définitive => le nombre d'alternatives différentes à examiner est réduit

Méthodes hiérarchiques : Inconvénients

Groupement de clusters est définitif => décisions erronées sont impossibles à modifier ultérieurement (méthode gloutonne)

Méthodes non extensibles pour des ensembles de données de grandes tailles

Clustering : Validation

Solution optimale connue (table de contingence) :

$$\frac{n_{11} + n_{00}}{n_{11} + n_{10} + n_{01}}$$

Solution optimale inconnue :

- Homogénéité (intra-cluster)
- Séparation (inter-cluster)

Règles d'association

Sommaire

Exemple : Panier de la ménagère

Définitions

A-Priori

Algorithmes génétiques

Résumé

Exemple : Analyse du panier de la ménagère

- Découverte d'**associations** et de **corrélations** entre les articles achetés par les clients en analysant les achats effectués (panier)

Lait, Oeufs, Céréale, Lait



Client 2

Lait, Oeufs, Sucre, Pain



Client 1

Oeufs, Sucre



Client 3

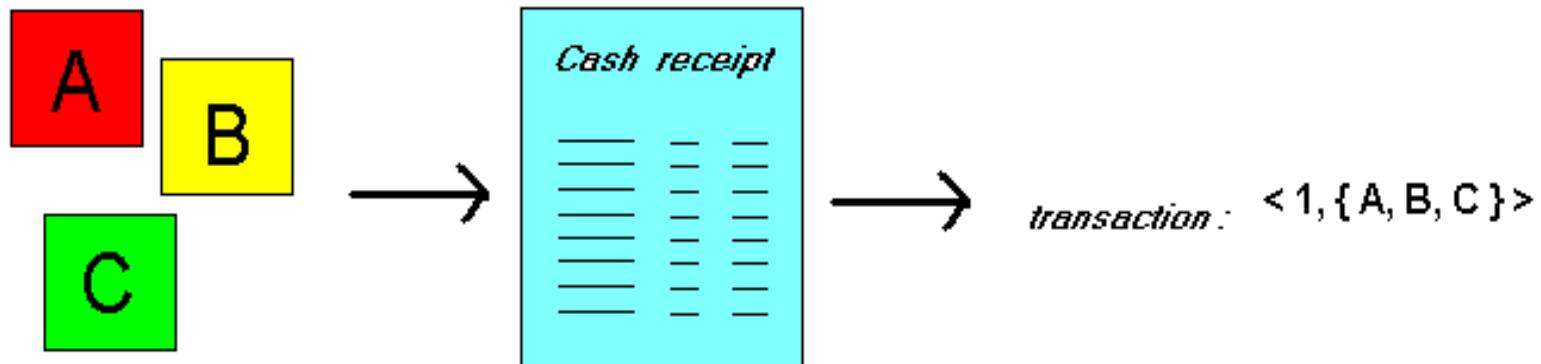
Exemple : Analyse du panier de la ménagère

Etant donnée :

- Une base de données de **transactions** de clients, où chaque transaction est représentée par un ensemble d'articles -**set of items**- (ex., produits)

Trouver :

- Groupes d'articles (itemset) achetés **fréquemment** (ensemble)



Exemple : Analyse du panier de la ménagère

Extraction d'informations sur le comportement de clients

- SI achat de riz + vin blanc ALORS achat de poisson (avec une grande probabilité)

Intérêt de l'information : peut suggérer ...

- Disposition des produits dans le magasin
- Quels produits mettre en promotion, gestion de stock, ...

Approche applicable dans d'autres domaines

- Cartes de crédit, e-commerce, ...
- Services des compagnies de télécommunication
- Services bancaires
- Traitements médicaux, ...

Règles d'associations

Recherche de règles d'association :

- Découvrir des patterns, corrélations, associations fréquentes, à partir d'ensembles d'items contenus dans des bases de données.

Compréhensibles : Facile à comprendre

Utiles : Aide à la décision

Efficaces : Algorithmes de recherche

Applications :

- Analyse des achats de clients, Marketing, Accès Web, Design de catalogue, Génomique, etc.

Règles d'associations

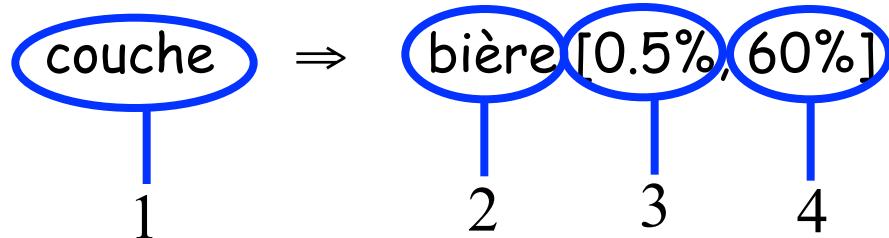
Formats de représentation des règles d'association :

- couches \Rightarrow bière [0.5%, 60%]
- achète:couches \Rightarrow achète:bière [0.5%, 60%]
- "SI achète couches ALORS achète bière dans 60% de cas. Les couches et la bière sont tous deux achetés dans 0.5% des transactions de la base de données."

Autres représentations (utilisée dans l'ouvrage de Han) :

- achète(x, "couches") \Rightarrow achète(x, "bière") [0.5%, 60%]

Règles d'associations



“SI achète couche,
ALORS achète bière,
dans 60% de cas,
dans 0.5% de la base”

Condition, partie gauche de la règle

Conséquence, partie droite de la règle

Support, fréquence (“partie gauche **et** droite sont présentes ensemble dans la base”)

Confiance (“si partie gauche de la règle est vérifiée, probabilité que la partie droite de la règle soit vérifiée”)

Règles d'associations

- **Support** : % d'instances de la base vérifiant la règle.

$$\text{support}(A \Rightarrow B [s, c]) = p(A \text{ et } B) = \underline{\text{support} (\{A,B\})}$$

- **Confiance** : % d'instances de la base vérifiant l'implication

$$\text{confiance}(A \Rightarrow B [s, c]) = p(B|A) = p(A \text{ et } B) / p(A) = \underline{\text{support}(\{A,B\})} / \underline{\text{support}(\{A\})}$$

Exemple

<i>TID</i>	<i>Items</i>
1	Pain, Lait
2	Bière, Couches, Pain, Oeufs
3	Bière, Coca, Couches, Lait
4	Bière, Pain, Couches, Lait
5	Coca, Pain, Couches, Lait

$\{Couches, Lait\} \Rightarrow_{s,\alpha} Bière$

Règle : $X \Rightarrow_{s,\alpha} y$

Support : $s = \frac{\sigma(X \cup y)}{|T|}$ ($s = P(X, y)$)

Confiance : $\alpha = \frac{\sigma(X \cup y)}{\sigma(X)}$ ($\alpha = P(y | X)$)

$$s = \frac{\sigma(Couches, Lait, Bière)}{\text{Nombre total d'instances}} = \frac{2}{5} = 0.4$$

$$\alpha = \frac{\sigma(Couches, Lait, Bière)}{\sigma(Couches, Lait)} = 0.66$$

Règles d'associations

Support minimum σ :

- Elevé \Rightarrow peu d'itemsets fréquents
- \Rightarrow peu de règles valides qui ont été souvent vérifiées
- Réduit \Rightarrow plusieurs règles valides qui ont été rarement vérifiées

Confiance minimum γ :

- Elevée \Rightarrow peu de règles, mais toutes “pratiquement” correctes
- Réduite \Rightarrow plusieurs règles, plusieurs d'entre elles sont “incertaines”

Valeurs utilisées : $\sigma = 2 - 10 \%$, $\gamma = 70 - 90 \%$

Recherche de règles d'association

Données d'entrée : liste d'achats

Achat = liste d'articles (longueur variable)

	Produit A	Produit B	Produit C	Produit D	Produit E
Achat 1	*			*	
Achat 2	*	*	*		
Achat 3	*				*
Achat 4	*			*	*
Achat 5		*		*	

Recherche de règles d'association

Tableau de co-occurrence : combien de fois deux produits ont été achetés ensemble ?

	Produit A	Produit B	Produit C	Produit D	Produit E
Produit A	4	1	1	2	2
Produit B	1	2	1	1	0
Produit C	1	1	1	0	0
Produit D	2	1	0	3	1
Produit E	2	0	0	1	2

Illustration / Exemple

- Règle d'association :
 - Si A alors B (règle 1)
 - Si A alors D (règle 2)
 - Si D alors A (règle 3)
- Supports :
 - $\text{Support}(1)=20\%$; $\text{Support}(2)=\text{Support}(3)=40\%$
- Confiances :
 - $\text{Confiance}(2) = 50\%$; $\text{Confiance}(3) = 67\%$
- On préfère la règle 3 à la règle 2.

Description de la méthode

- Support et confiance ne sont pas toujours suffisants
- Ex : Soient les 3 articles A, B et C

article	A	B	C	A et B	A et C	B et C	A, B et C
support	45%	42,5%	40%	25%	20%	15%	5%

- Règles à 3 articles : même support 5%
- Confiance
 - Règle : Si A et B alors C = 0.20
 - Règle : Si A et C alors B = 0.25
 - Règle : Si B et C alors A = 0.33

Recherche de règles

- Soient une liste de n articles et de m achats.
- 1. Calculer le nombre d'occurrences de chaque article.
- 2. Calculer le tableau des co-occurrences pour les paires d 'articles.
- 3. Déterminer les règles de niveau 2 en utilisant les valeurs de support, confiance et amélioration.
- 4. Calculer le tableau des co-occurrences pour les triplets d 'articles.
- 5. Déterminer les règles de niveau 3 en utilisant les valeurs de support, confiance et amélioration
- ...

Complexité

- Soient :
 - n : nombre de transactions dans la BD
 - m : Nombre d'attributs (items) différents
- Complexité
 - Nombre de règles d'association : $O(m \cdot 2^{m-1})$
 - Complexité de calcul : $O(n \cdot m \cdot 2^m)$

Réduction de la complexité

- n de l'ordre du million (parcours de la liste nécessaire)
- Taille des tableaux en fonction de m et du nombre d 'articles présents dans la règle

	2	3	4
n	$n(n-1)/2$	$n(n-1)(n-2)/6$	$n(n-1)(n-2)(n-3)/24$
100	4950	161 700	3 921 225
10000	$5 \cdot 10^7$	$1.7 \cdot 10^{11}$	$4.2 \cdot 10^{14}$

- Conclusion de la **règle restreinte** à un sous-ensemble de l 'ensemble des articles vendus.
 - **Exemple** : articles nouvellement vendues.
- Crédation de **groupes** d 'articles (différents niveaux d'abstraction).
- **Elagage** par support minimum.

Illustration sur une BD commerciale

Attribut	Compteur
Pain	4
Coca	2
Lait	4
Bière	3
Couches	4
Oeufs	1

Attributs (1-itemsets)



Itemset	Compteur
{Pain,Lait}	3
{Pain,Bière}	2
{Pain,Couches}	3
{Lait,Bière}	2
{Lait,Couches}	3
{Bière,Couches}	3

paires (2-itemsets)



Triplets (3-itemsets)

Support Minimum = 3

Si tout sous-ensemble est considéré,

$$C_1^6 + C_2^6 + C_3^6 = 41$$

En considérant un seuil support min,

$$6 + 6 + 2 = 14$$

Itemset	Compteur
{Pain,Lait,Couches}	3
{Lait,Couches,Bière}	2



L'algorithme Apriori [Agrawal93]

- **Deux étapes**
 - Recherche des k-itemsets fréquents ($\text{support} \geq \text{MINSUP}$)
 - (Pain, Fromage, Vin) = 3-itemset
 - **Principe** : Les sous-itemsets d'un k-itemset fréquent sont obligatoirement fréquents
 - Construction des règles à partir des k-itemsets trouvés
 - Une règle fréquente est retenue si et seulement si sa confiance $c \geq \text{MINCONF}$
 - **Exemple** : ABCD fréquent
 - $AB \rightarrow CD$ est retenue si sa confiance $\geq \text{MINCONF}$

Recherche des k-itemsets fréquents (1)

Exemple

- $I = \{A, B, C, D, E, F\}$
- $T = \{AB, ABCD, ABD, ABDF, ACDE, BCDF\}$
- $\text{MINSUP} = 1/2$

Calcul de L1 (ensemble des 1-itemsets)

- $C_1 = I = \{A, B, C, D, E, F\}$ // C_1 : ensemble de 1-itemsets candidats
- $s(A) = s(B) = 5/6, s(C) = 3/6, s(D) = 5/6, s(E) = 1/6, s(F) = 2/6$
- $L_1 = \{A, B, C, D\}$

Calcul de L2 (ensemble des 2-itemsets)

- $C_2 = L_1 \times L_1 = \{AB, AC, AD, BC, BD, CD\}$
- $s(AB) = 4/6, s(AC) = 2/6, s(AD) = 4/6, s(BC) = 2/6, s(BD) = 4/6, s(CD) = 3/6$
- $L_2 = \{AB, AD, BD, CD\}$

Recherche des k-itemsets fréquents (2)

- Calcul de L_3 (ensemble des 3-itemsets)
 - $C_3 = \{ABD\}$ ($ABC \notin C_3$ car $AC \notin L_2$)
 - $s(ABD) = 3/6$
 - $L_3 = \{ABD\}$
- Calcul de L_4 (ensemble des 4-itemsets)
 - $C_4 = \emptyset$
 - $L_4 = \emptyset$
- Calcul de L (ensembles des itemsets fréquents)
 - $L = \bigcup L_i = \{A, B, C, D, AB, AD, BD, CD, ABD\}$

L'algorithme Apriori

```
L1 = {1-itemsets fréquents};  
for (k=2; Lk-1 ≠ φ; k++) do  
    Ck = apriori_gen(Lk-1);  
    forall instances t ∈ T do  
        Ct = subset(Ck, t);  
        forall candidats c ∈ Ct do  
            c.count++;  
    Lk = { c ∈ Ck / c.count ≥ MINSUP }  
L = ∪iLi;
```

La procédure Apriori_gen

{ Jointure $L_{k-1} * L_{k-1}$; k-2 éléments communs}

insert into C_k ;

select p.item₁, p.item₂, ..., p.item_{k-1}, q.item_{k-1}

from L_{k-1p}, L_{k-1q}

where p.item₁=q.item₁, ..., p.item_{k-2}=q.item_{k-2}
, p.item_{k-1}< q.item_{k-1}

forall itemsets $c \in C_k$ **do**

forall (k-1)-itemsets $s \subset c$ **do**

if $s \notin L_{k-1}$ **then**

delete c **from** C_k ;

Apriori - Exemple

Base de données D

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

Scan D

C_1

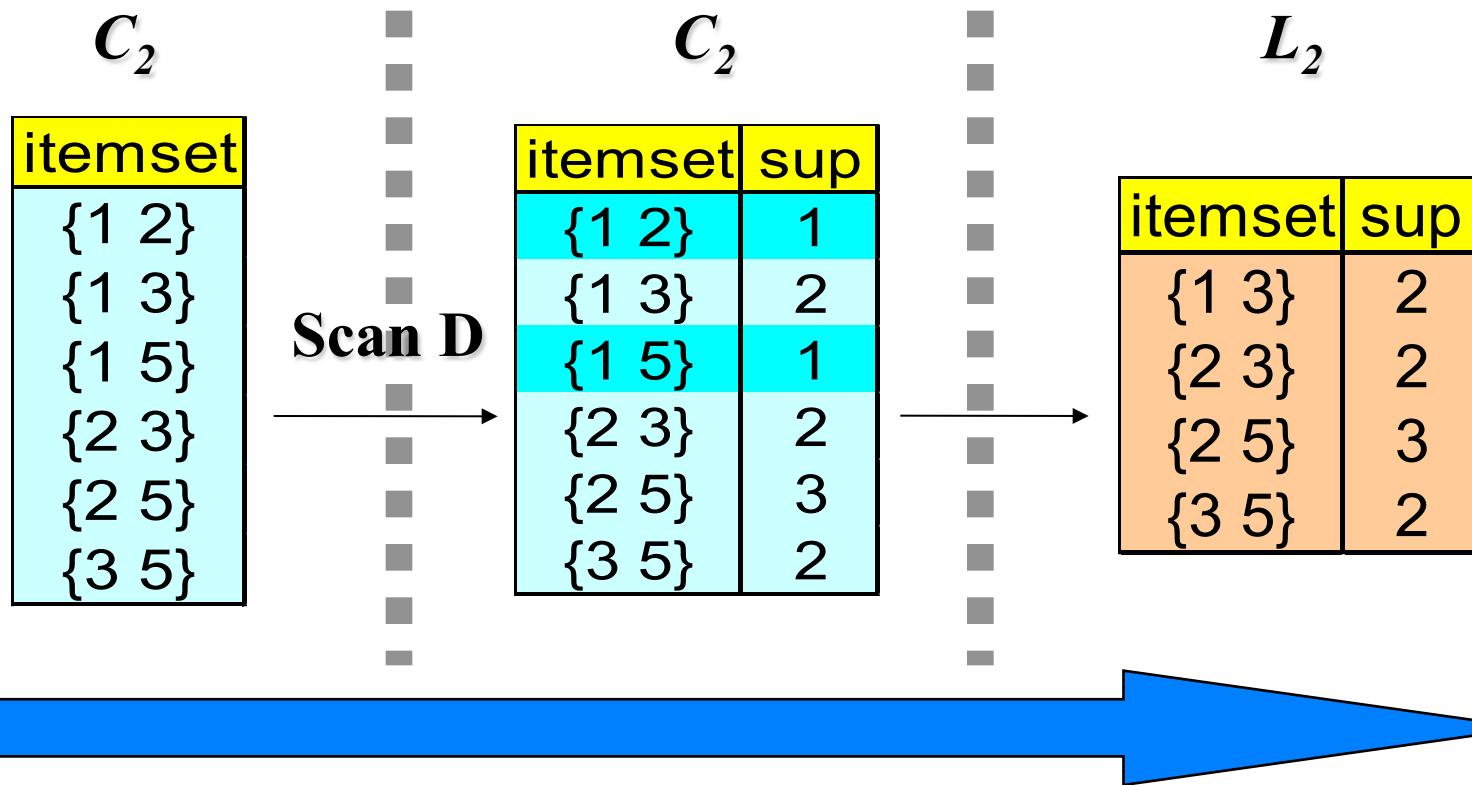
itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

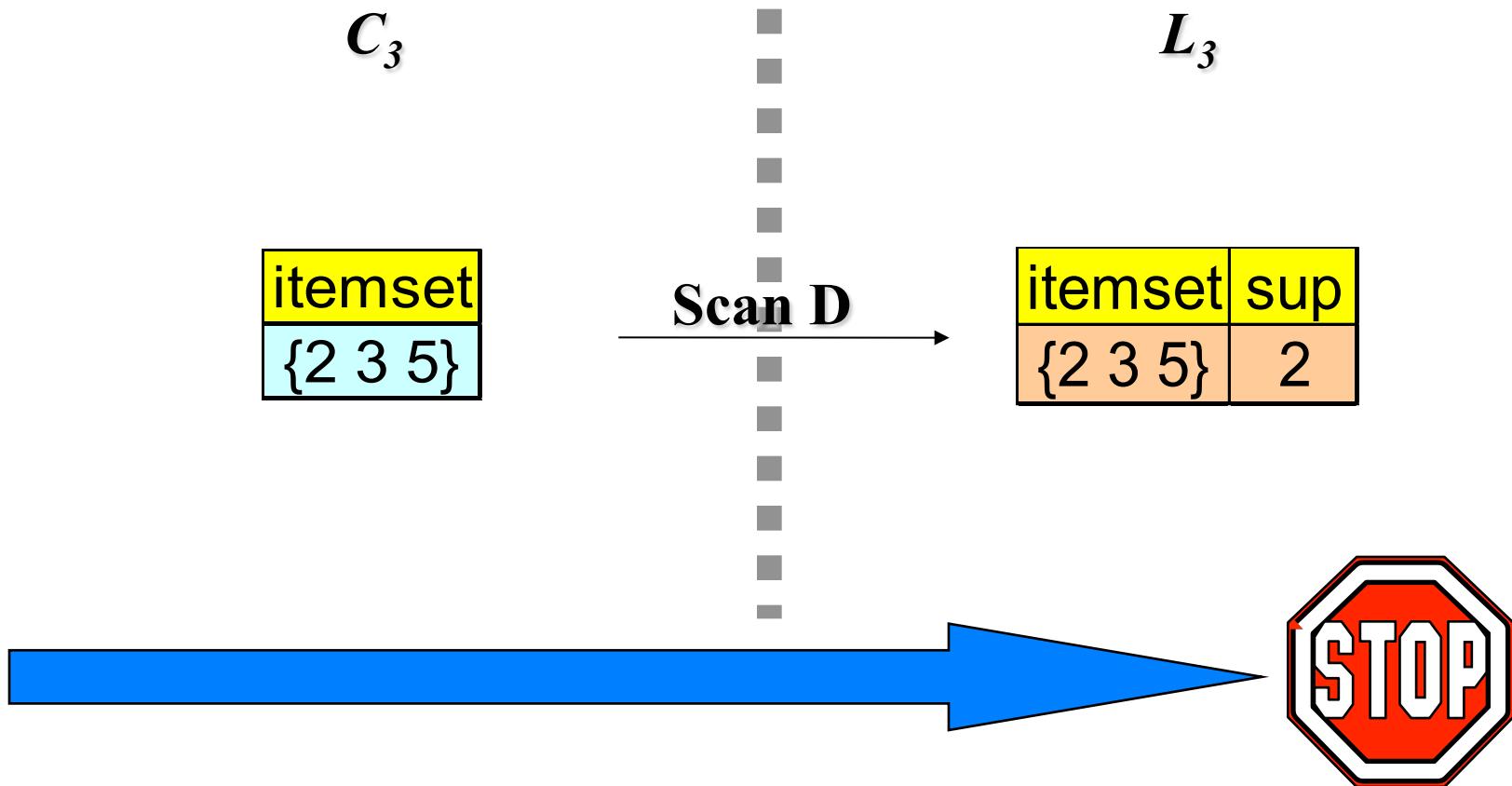
itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3



Apriori - Exemple

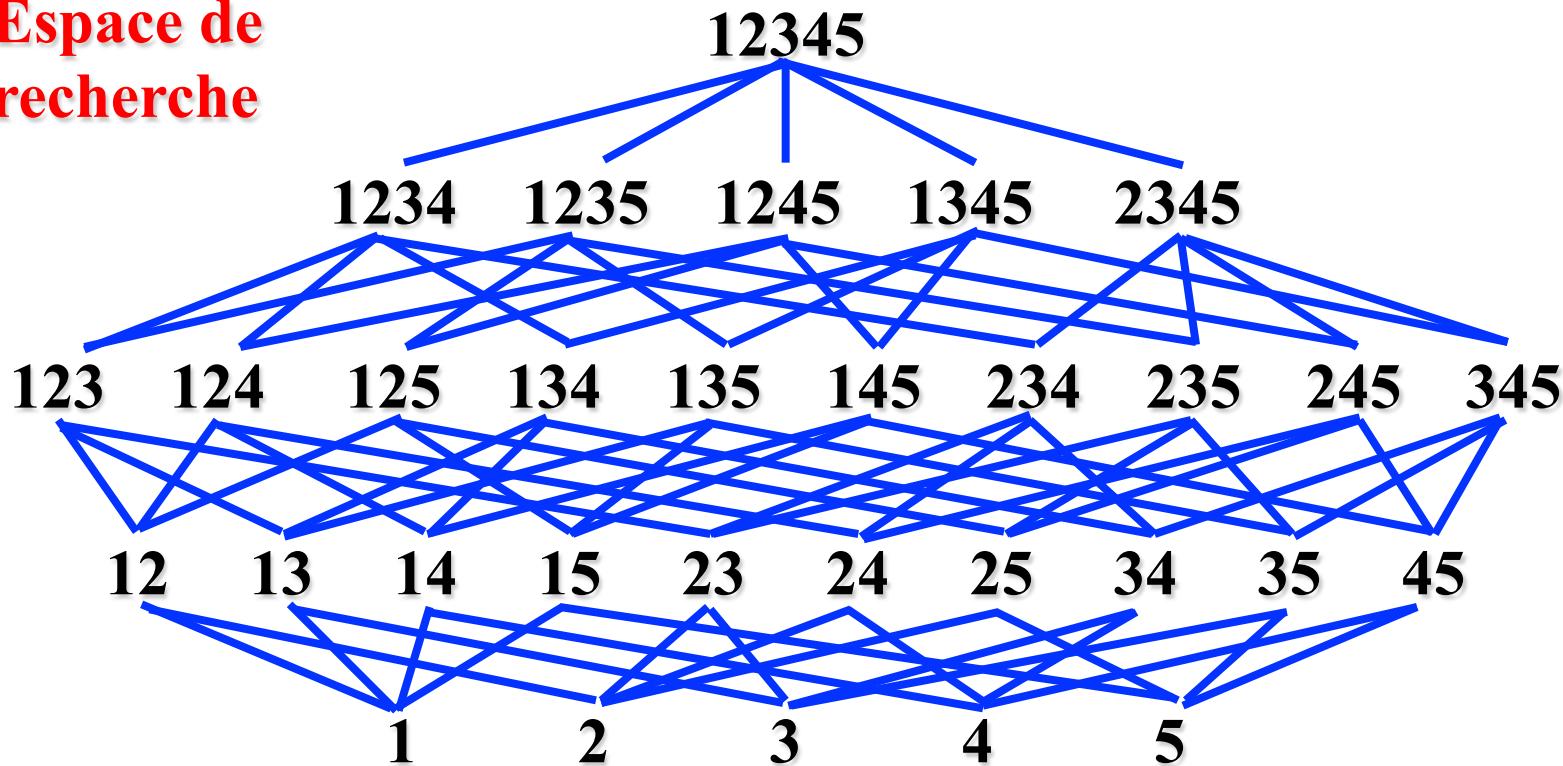


Apriori - Exemple



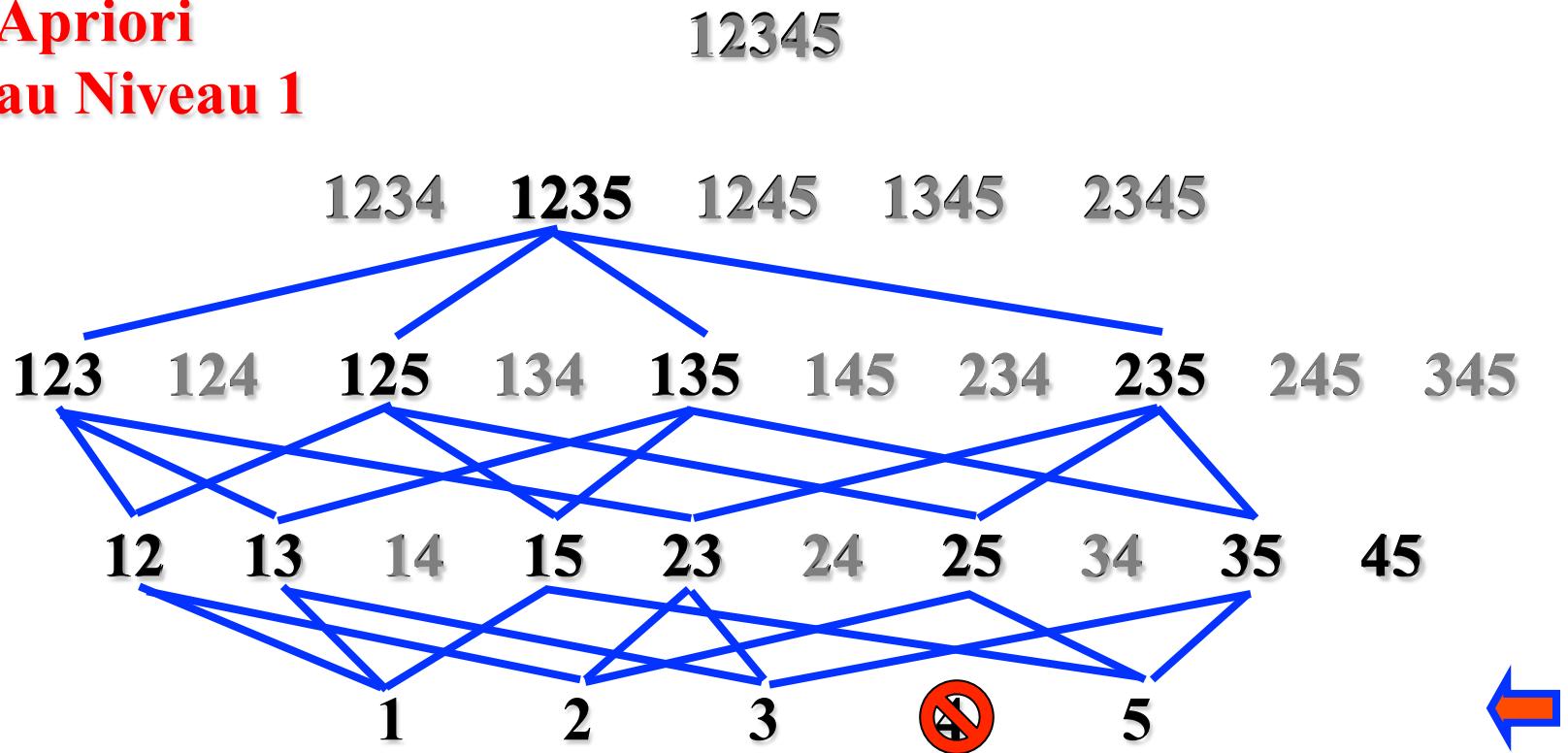
Apriori - Exemple

Espace de recherche



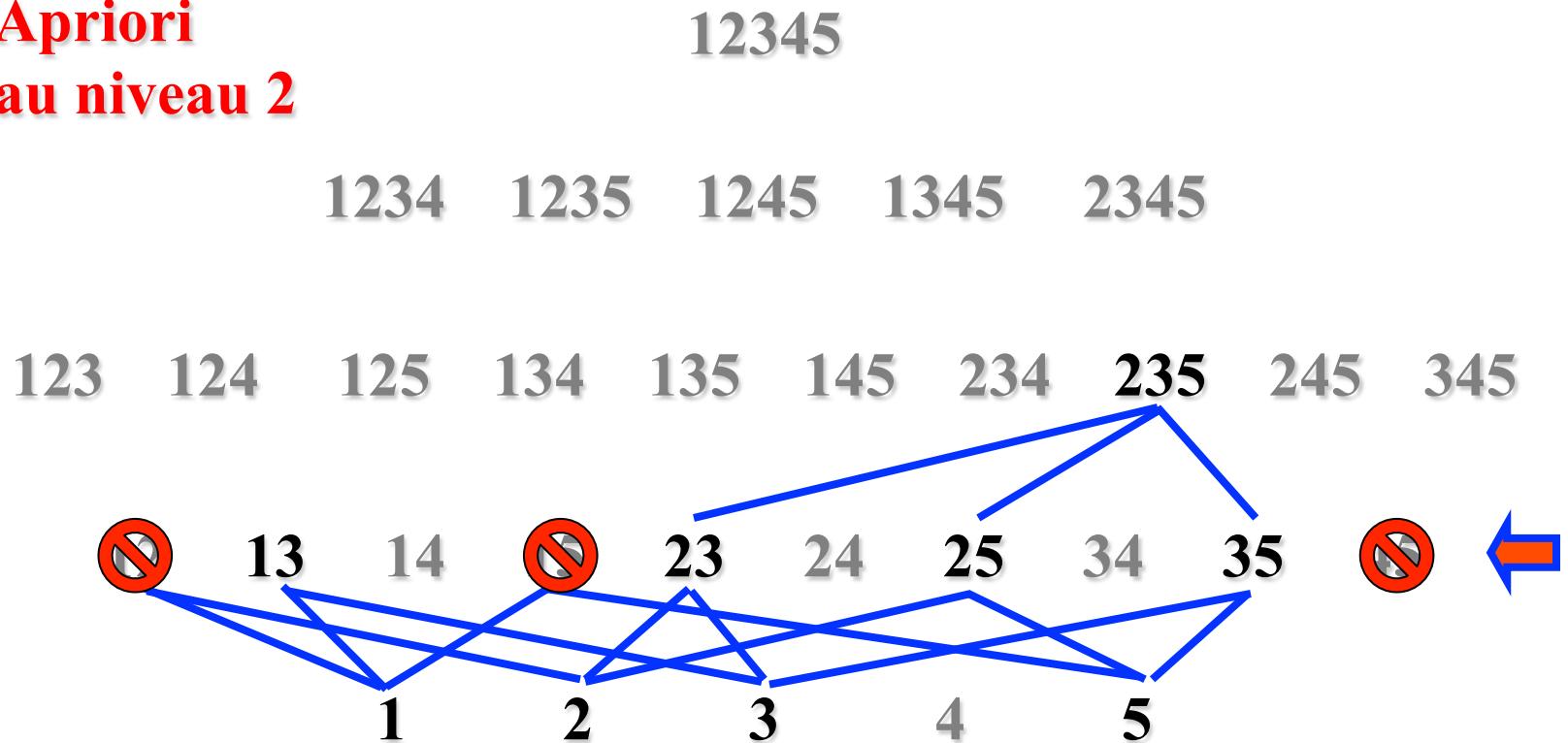
Apriori - Exemple

Apriori
au Niveau 1



Apriori - Exemple

Apriori
au niveau 2



Génération des règles à partir des itemsets

Pseudo-code :

- **pour** chaque itemset fréquent I
 - générer tous les sous-itemsets non vides s de I
 - **pour** chaque sous-itemset non vide s de I
 - produire la règle " $s \Rightarrow (I-s)$ " si $support(I)/support(s) \geq min_conf$ ", où min_conf est la confiance minimale
- **Exemple** : itemset fréquent $I = \{abc\}$,
 - Sous-itemsets $s = \{a, b, c, ab, ac, bc\}$
 - $a \Rightarrow bc$, $b \Rightarrow ac$, $c \Rightarrow ab$
 - $ab \Rightarrow c$, $ac \Rightarrow b$, $bc \Rightarrow a$

Génération des règles à partir des itemsets

Règle 1 à mémoriser :

- La génération des itemsets fréquents est une opération **coûteuse**
- La génération des règles d'association à partir des itemsets fréquents est **rapide**

Règle 2 à mémoriser :

- Pour la génération des itemsets, le **seuil support** est utilisé.
- Pour la génération des règles d'association, le **seuil confiance** est utilisé.

Complexité en pratique ?

- A partir d'un exemple réel (petite taille) ...
- Expériences réalisées sur un serveur Alpha Citum 4/275 avec 512 MB de RAM & Red Hat Linux release 5.0 (kernel 2.0.30)

Apriori - Complexité

Phase coûteuse : Génération des candidats

- Ensemble des candidats de grande taille :
 - 10^4 1-itemset fréquents génèrent 10^7 candidats pour les 2-itemsets
 - Pour trouver un itemset de taille 100, e.x., $\{a_1, a_2, \dots, a_{100}\}$, on doit générer $2^{100} \approx 10^{30}$ candidats.
- Multiple scans de la base de données :
 - Besoin de $(n + 1)$ scans, n est la longueur de l'itemset le plus long

Apriori - Complexité

En pratique :

- Pour l'algorithme Apriori basique, le nombre d'attributs est généralement plus critique que le nombre de transactions
- Par exemple :
 - 50 attributs chacun possédant 1-3 valeurs, 100.000 transactions (not very bad)
 - 50 attributs chacun possédant 10-100 valeurs, 100.000 transactions (quite bad)
 - 10.000 attributs chacun possédant 5-10 valeurs, 100 transactions (very bad...)
- Notons :
 - Un attribut peut avoir plusieurs valeurs différentes
 - Les algorithmes traitent chaque paire attribut-valeur comme un attribut (2 attributs avec 5 valeurs → “10 attributs”)

Quelques pistes pour résoudre le problème ...

Apriori – Réduction de la complexité

Suppression de transactions :

- Une transaction qui ne contient pas de k-itemsets fréquents est inutile à traiter dans les parcours (scan) suivants.

Partitionnement :

- Tout itemset qui est potentiellement fréquent dans une BD doit être potentiellement fréquent dans au moins une des partitions de la BD.

Echantillonage :

- Extraction à partir d'un sous-ensemble de données, décroître le seuil support

Apriori - Avantages

- Résultats clairs : règles faciles à interpréter.
- Simplicité de la méthode
- Aucune hypothèse préalable (Apprentissage non supervisé)
- Introduction du temps : méthode facile à adapter aux séries temporelles. Ex : Un client ayant acheté le produit A est susceptible d'acheter le produit B dans deux ans.

Apriori - Inconvénients

- **Coût de la méthode** : méthode coûteuse en temps
- **Qualité des règles** : production d'un nombre important de règles triviales ou inutiles.
- **Articles rares** : méthode non efficace pour les articles rares.
- **Adapté aux règles binaires**
- Apriori amélioré
 - Variantes de Apriori : DHP, DIC, etc.
 - Partition [Savasere et al. 1995]
 - Eclat et Clique [Zaki et al. 1997]
 - ...

Typologie des règles

- Règles d'association binaires
 - Forme : *if C then P.* C,P : ensembles d'objets
- Règles d'association quantitatives
 - Forme : *if C then P*
 - C = terme1 & terme2 & ... & termen
 - P = termen+1
 - termei = <attributj, op, valeur> ou <attributj, op, valeur_de, valeur_a>
 - Classes : valeurs de P
 - Exemple : if ((Age>30) & (situation=marié)) then prêt=prioritaire
- Règles de classification généralisée
 - Forme : *if C then P, P=p1, p2, ..., pm* P: attribut but
- etc.

Règles d'association – Résumé

- Probablement la contribution la plus significative de la communauté KDD
- Méthodes de recherche de règles :
 - A-priori
 - Algorithmes génétiques
- Plusieurs travaux ont été publiés dans ce domaine

Règles d'association – Résumé

Plusieurs issues ont été explorées : intérêt d'une règle, optimisation des algorithmes, parallélisme et distribution, ...

Directions de recherche :

- Règles d'associations pour d'autres types de données : données spatiales, multimedia, séries temporelles, ...

Critères pour les règles

Mesure	Formule	Effet
Support S	$\frac{C \text{ et } P}{N}$	% transactions qui contiennent C et P
Confiance C	$\frac{C \text{ et } P}{C}$	Probabilité conditionnelle
Intérêt I	$\frac{C \text{ et } P}{C \times P}$	Privilégie les motifs rares (ayant un support faible)
Conviction V	$\frac{C \times \bar{P}}{C \text{ et } \bar{P}}$	Mesure la faiblesse de (C, not P) $V >> \therefore P$ se passe avec C
Piatetsky-Shapiro's	$C \text{ et } P - C \times P$	Mesure la dépendance
Surprise R	$\frac{(C \text{ et } P - C \text{ et } \bar{P})}{P}$	Cherche des règles étonnantes Mesure l'infirimation(C, NOT P)

Classification

Sommaire

Définition

Validation d'une classification (accuracy)

K-NN (plus proches voisins)

Arbres de décision

Réseaux de neurones

Autres méthodes de classification

Etude de cas réel : Protéomique

Résumé

Classification

- Elle permet de **prédir** si un élément est membre d'un groupe ou d'une catégorie donné.
- **Classes**
 - Identification de groupes avec des profils particuliers
 - Possibilité de décider de l'appartenance d'une entité à une classe
- Caractéristiques
 - **Apprentissage supervisé** : classes connues à l'avance
 - Pb : qualité de la classification (taux d'erreur)
 - Ex : établir un diagnostic (si erreur !!!)

Classification - Applications

Comprendre les critères prépondérants pour l'achat d'un produit ou d'un service

Isoler les critères explicatifs d'un comportement d'achat

Analyse de risque: détecter les facteurs prédisant un comportement de non paiement

Détecter les causes de réclamation

Processus à deux étapes



Etape 1 :

Construction du modèle à partir de l'ensemble d'apprentissage (training set)

Etape 2 :

Utilisation du modèle : tester la précision du modèle et l'utiliser dans la classification de nouvelles données

Construction du modèle



Chaque **instance** est supposée appartenir à une classe prédefinie

La classe d'une instance est déterminée par l'attribut "**classe**"

L'ensemble des instances d'apprentissage est utilisé dans la construction du modèle

Le **modèle** est représenté par des règles de classification, arbres de décision, formules mathématiques, ...

Utilisation du modèle

Classification de nouvelles instances ou instances inconnues



Estimer le taux d'erreur du modèle

- la classe connue d'une instance test est comparée avec le résultat du modèle
- Taux d'erreur = pourcentage de tests incorrectement classés par le modèle

Validation de la Classification (accuracy)

Estimation des taux d'erreurs :

Partitionnement : apprentissage et test (ensemble de données important)

- Utiliser 2 ensembles indépendents, e.g., ensemble d'apprentissage (2/3), ensemble test (1/3)

Apprentissage D_t

Validation $D \setminus D_t$

Validation de la Classification (accuracy)

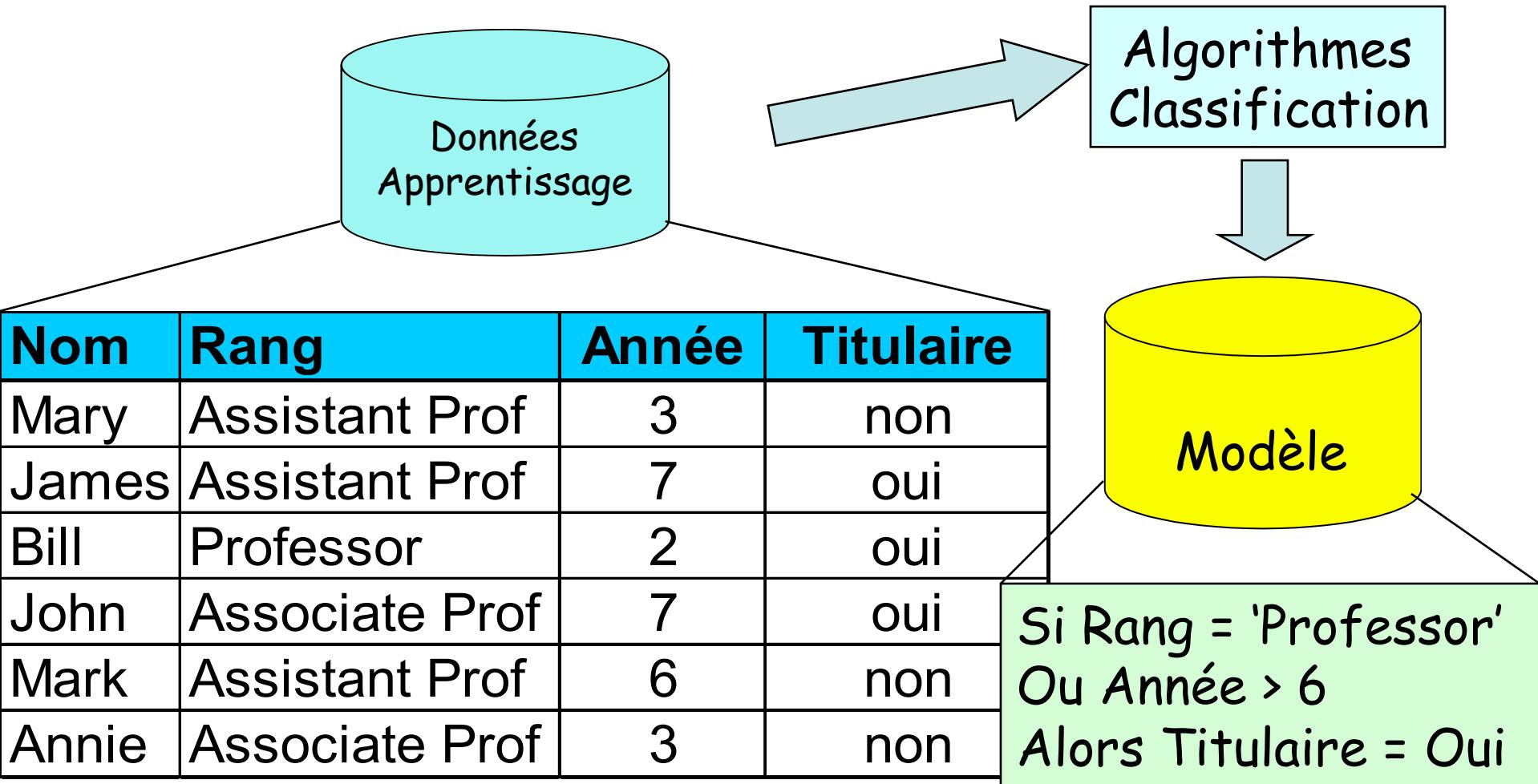
Validation croisée (ensemble de données modéré)

- Diviser les données en k sous-ensembles
- Utiliser $k-1$ sous-ensembles comme données d'apprentissage et un sous-ensemble comme données test

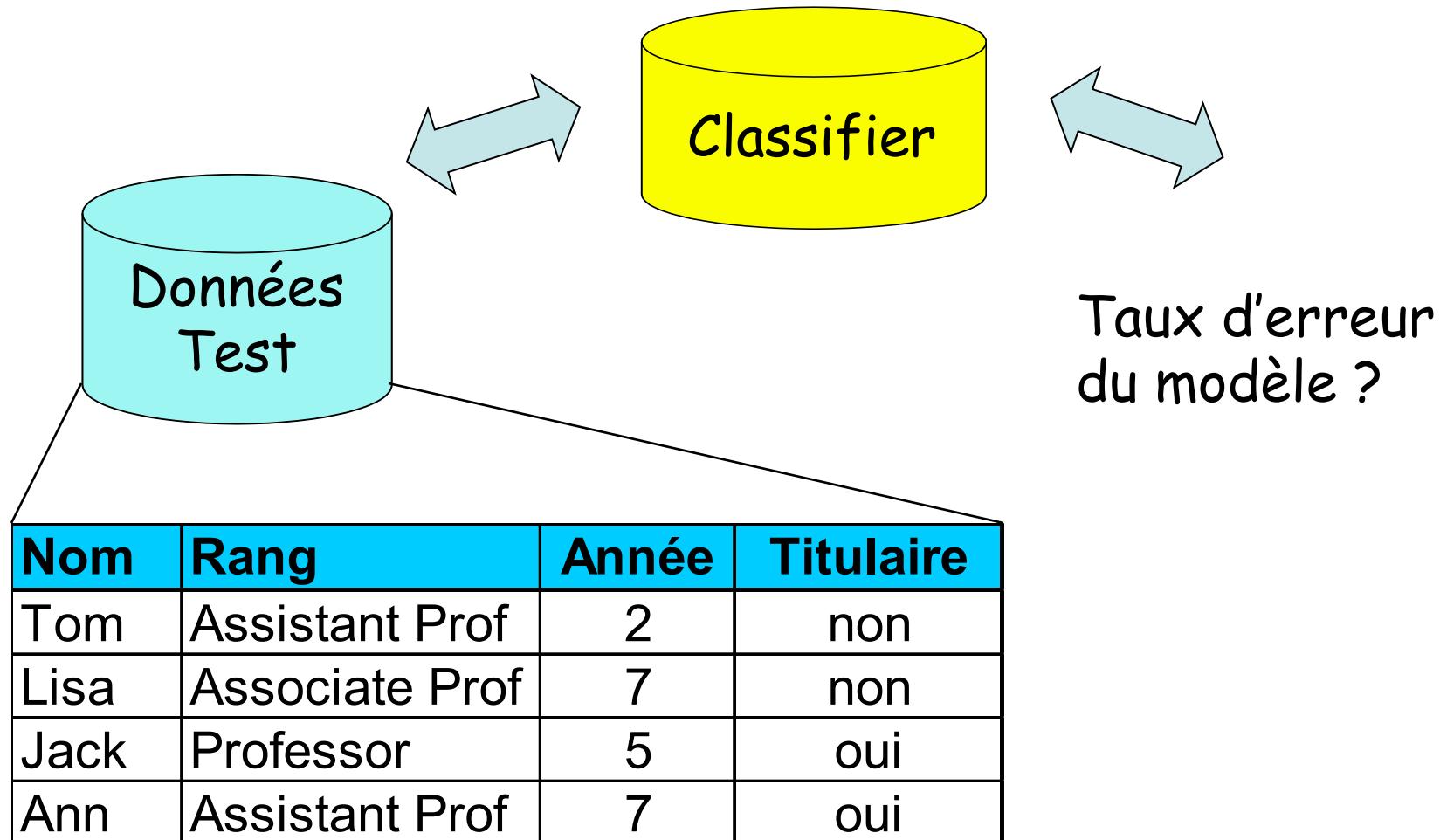


Bootstrapping : n instances test aléatoires (ensemble de données réduit)

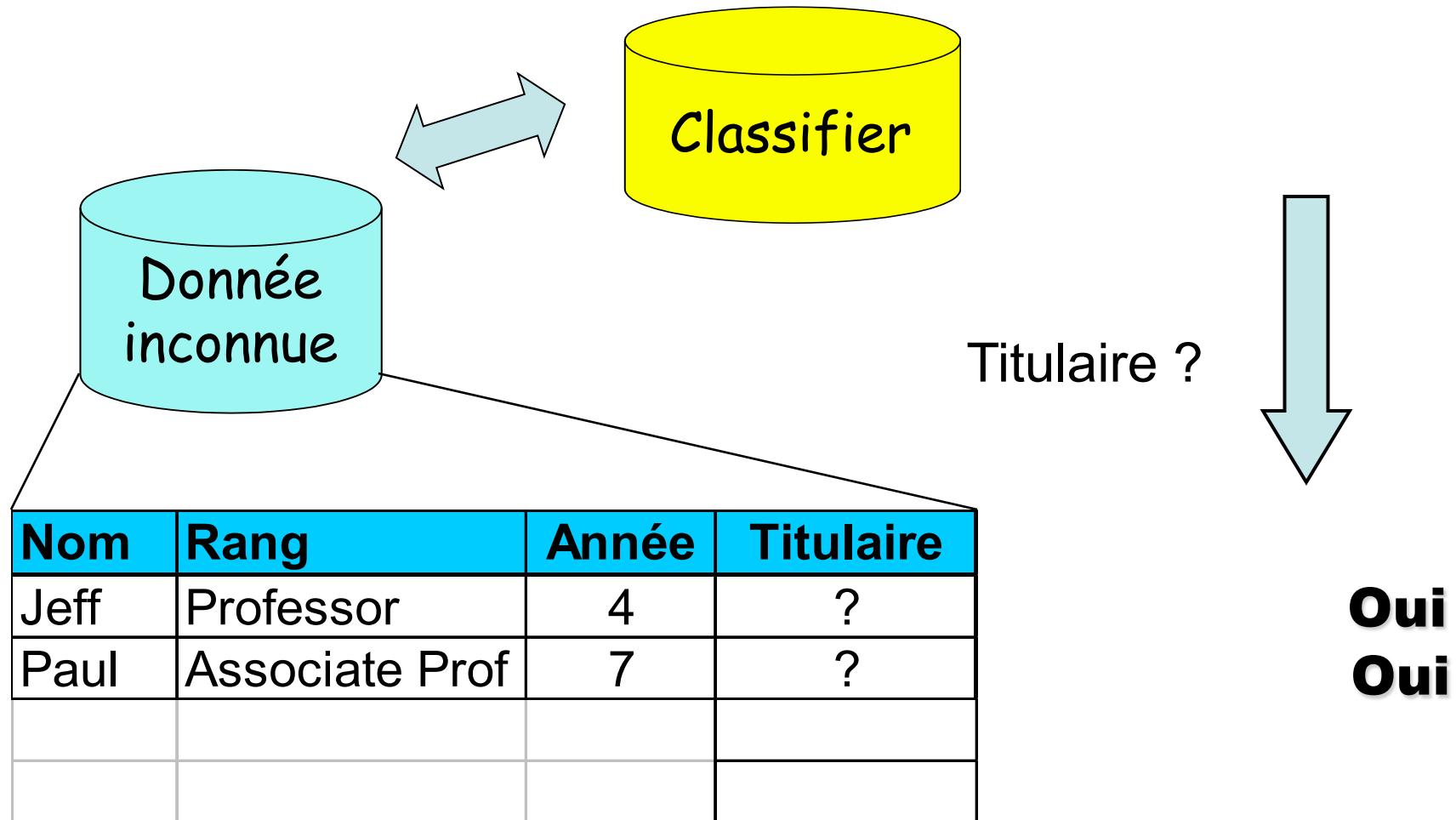
Exemple : Construction du modèle



Exemple : Utilisation du modèle



Exemple : Utilisation du modèle



Evaluation des méthodes de classification

Taux d'erreur (Accuracy)

Temps d'exécution (construction, utilisation)

Robustesse (bruit, données manquantes,...)

Extensibilité

Interprétabilité

Simplicité



Méthodes de Classification



- Méthode K-NN (plus proche voisin)
- Arbres de décision
- Réseaux de neurones
- Classification bayésienne
- Caractéristiques
 - Apprentissage supervisé (classes connues)

Méthode des plus proches voisins

Méthode dédiée à la classification (k-NN : nearest Neighbors).

Méthode de raisonnement à partir de cas : prendre des décisions en recherchant un ou des cas similaires déjà résolus.

Pas d'étape d 'apprentissage : construction d 'un modèle à partir d'un échantillon d 'apprentissage (réseaux de neurones, arbres de décision, ...).

Modèle = échantillon d'apprentissage + fonction de distance + fonction de choix de la classe en fonction des classes des voisins les plus proches.

Algorithme kNN (K-nearest neighbors)

Objectif : affecter une classe à une nouvelle instance

donnée : un échantillon de m enregistrements classés ($x, c(x)$)

entrée : un enregistrement y

- 1. Déterminer les k plus proches enregistrements de y
- 2. combiner les classes de ces k exemples en une classe c

sortie : la classe de y est $c(y)=c$

Algorithme kNN : sélection de la classe

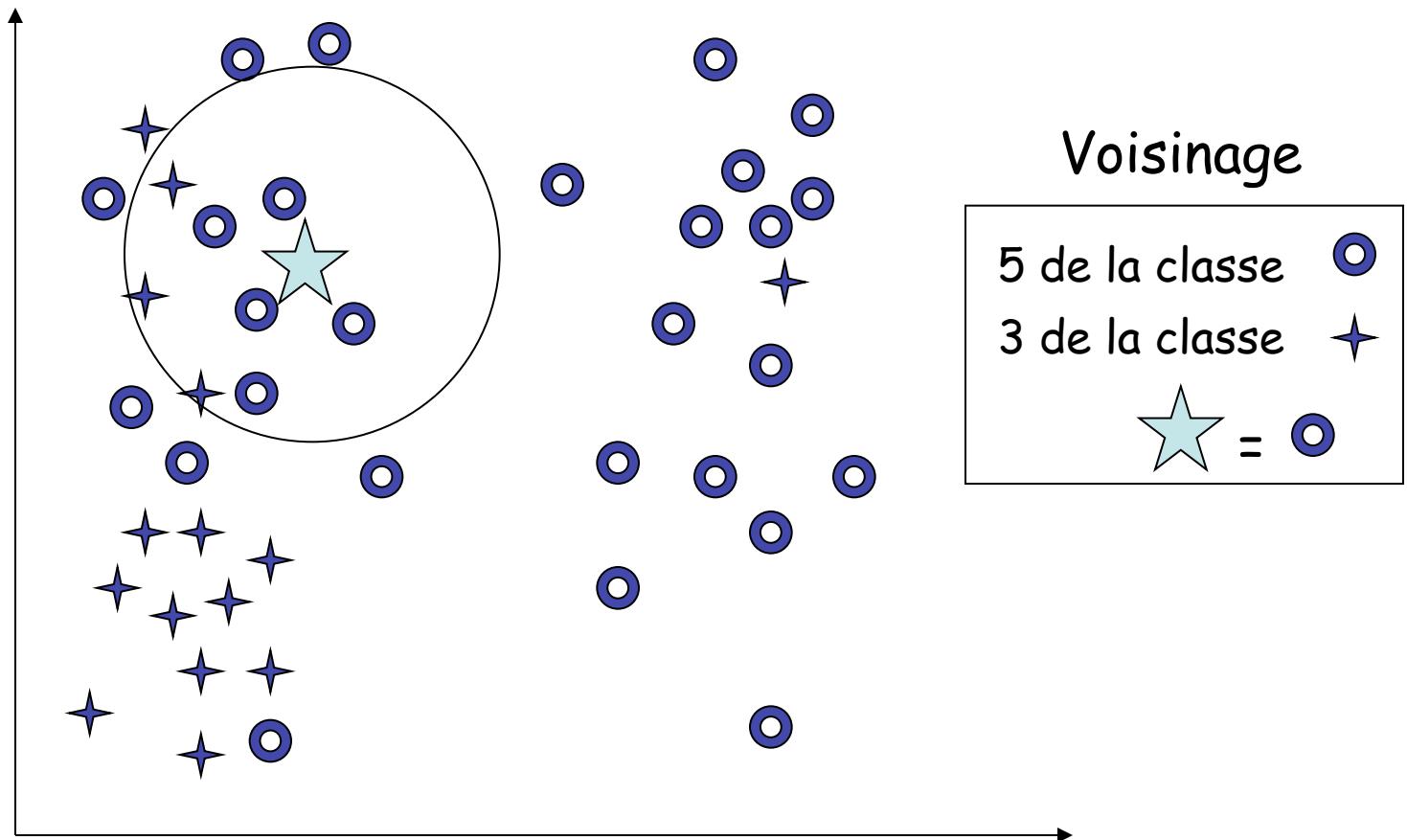
Solution simple : rechercher le cas le plus proche et prendre la même décision (Méthode 1-NN).

Combinaison des k classes :

- Heuristique : $k = \text{nombre d'attributs} + 1$
- Vote majoritaire : prendre la classe majoritaire.
- Vote majoritaire pondéré : chaque classe est pondérée. Le poids de $c(x_i)$ est inversement proportionnel à la distance $d(y, x_i)$.

Confiance : Définir une confiance dans la classe attribuée = rapport entre les votes gagnants et le total des votes.

Illustration



Retour sur KNN : Exemple (1)

Customer	Age	Income	No. credit cards	Loyal
John 	35	35K	3	No
Rachel 	22	50K	2	Yes
Hannah 	63	200K	1	No
Tom 	59	170K	1	No
Nellie 	25	40K	4	Yes
David 	37	50K	2	?

Retour sur KNN : Exemple (2)

K = 3

Customer	Age	Income	No. credit cards	Loyal	Distance from David
John 	35	35K	3	No	$\sqrt{[(35-37)^2 + (35-50)^2 + (3-2)^2]} = 15.16$
Rachel 	22	50K	2	Yes	$\sqrt{[(22-37)^2 + (50-50)^2 + (2-2)^2]} = 15$
Hannah 	63	200K	1	No	$\sqrt{[(63-37)^2 + (200-50)^2 + (1-2)^2]} = 152.23$
Tom 	59	170K	1	No	$\sqrt{[(59-37)^2 + (170-50)^2 + (1-2)^2]} = 122$
Nellie 	25	40K	4	Yes	$\sqrt{[(25-37)^2 + (40-50)^2 + (4-2)^2]} = 15.74$
David 	37	50K	2	Yes	

Algorithme kNN : critique

Pas d'apprentissage : introduction de nouvelles données ne nécessite pas la reconstruction du modèle.

Clarté des résultats

Tout type de données

Nombre d'attributs

Temps de classification : -

Stocker le modèle : -

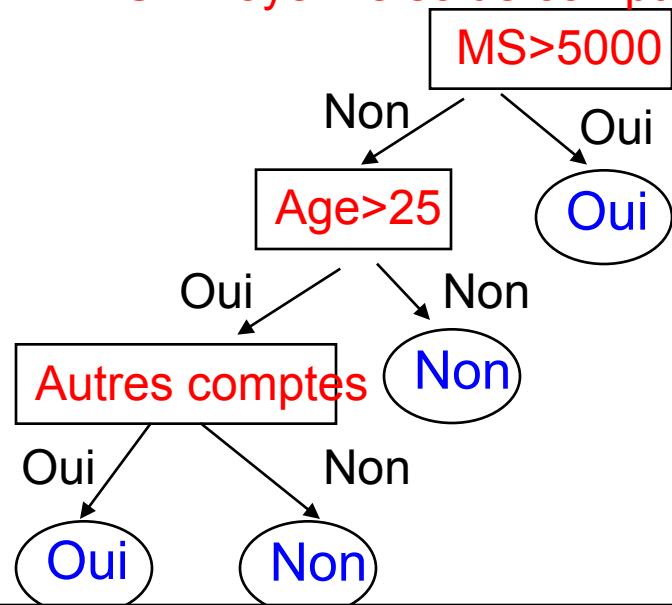
Distance et nombre de voisins : dépend de la distance, du nombre de voisins et du mode de combinaison.

Arbres de décision

- **Génération d'arbres de décision à partir des données**
- **Arbre = Représentation graphique d'une procédure de classification**

Accord d'un prêt bancaire

MS : moyenne solde compte courant



Un arbre de décision est un arbre où :

Noeud interne = un attribut

Branche d'un noeud = un test sur un attribut

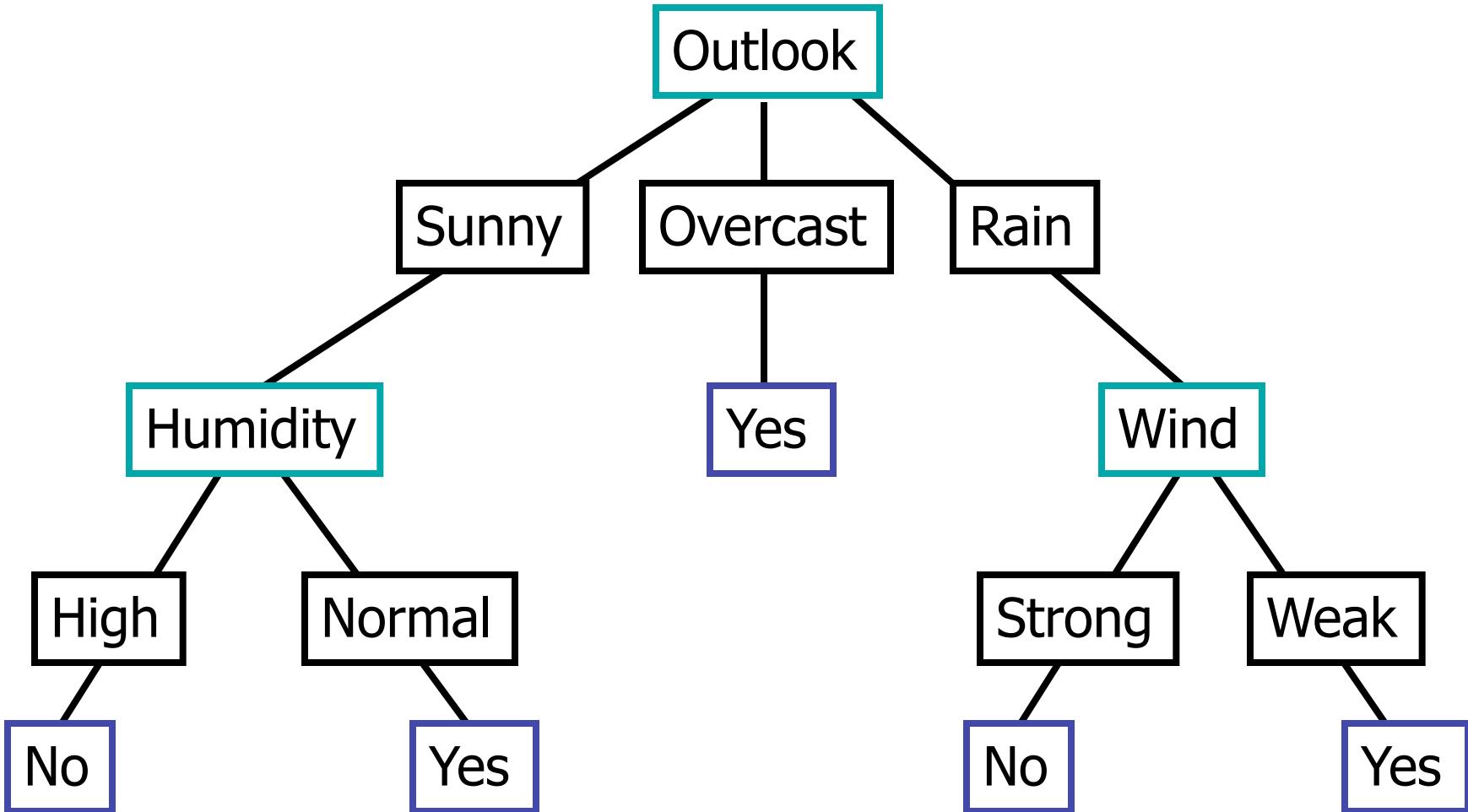
Feuilles = classe donnée

Arbre de décision - Exemple

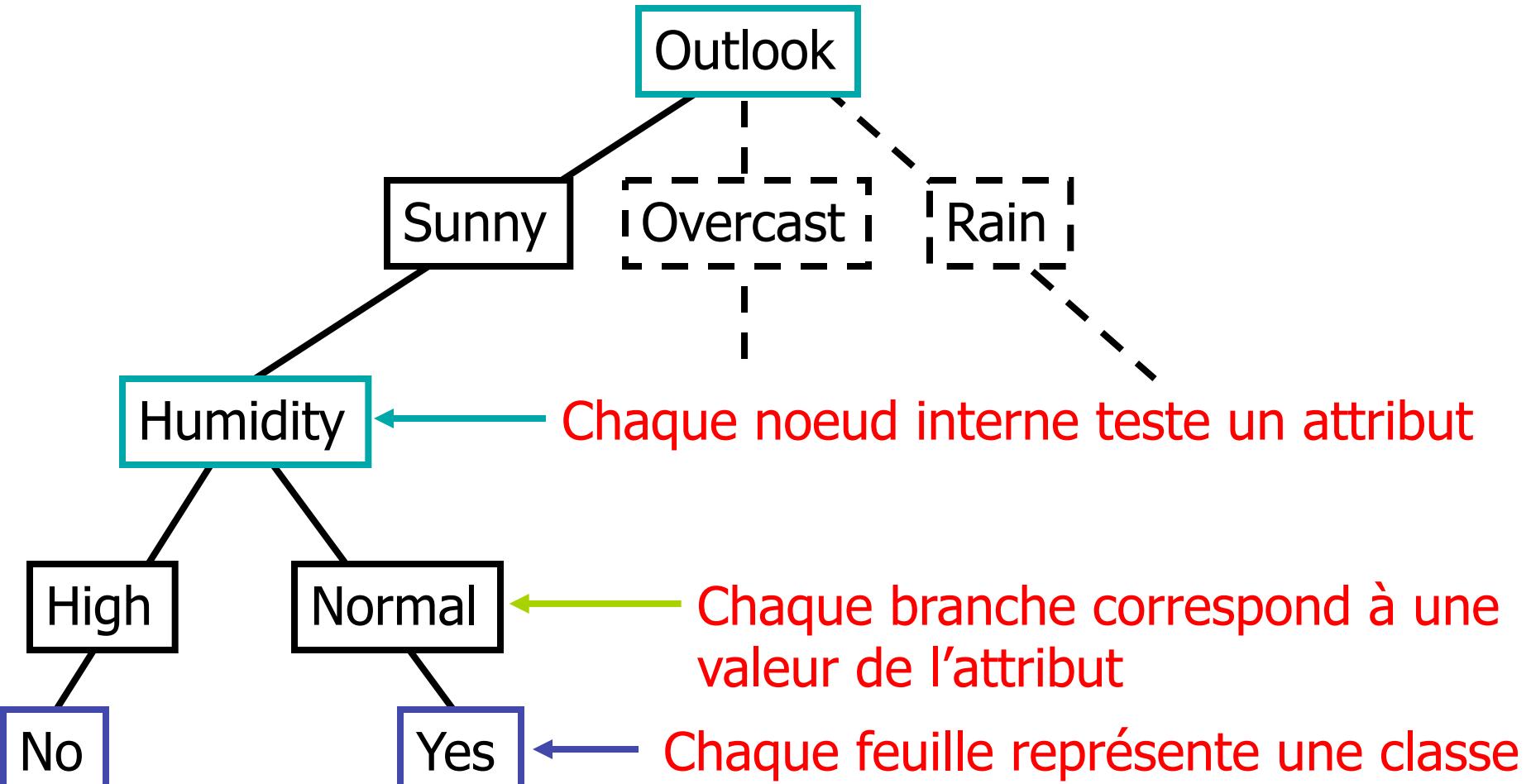
Ensemble
d'apprentissage

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Arbre de décision - Exemple



Exemple – Jouer au tennis ?



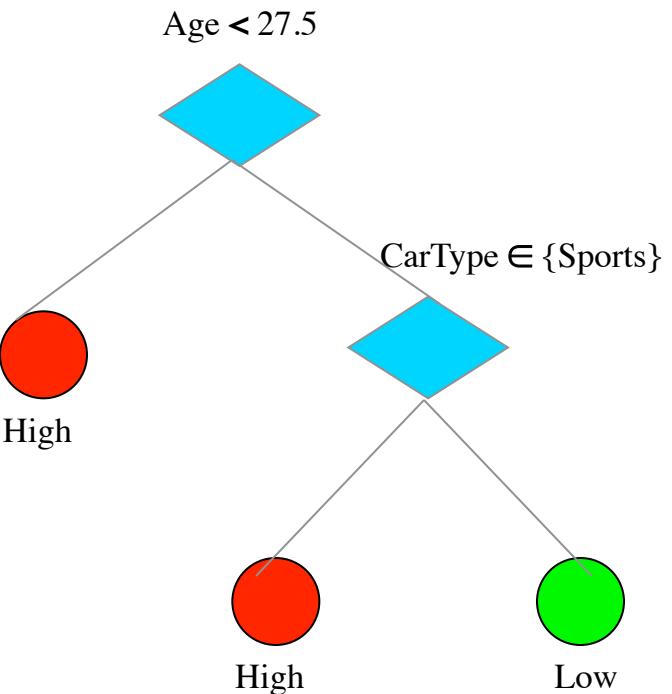
Arbres de décision – Exemple

Risque - Assurances

Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

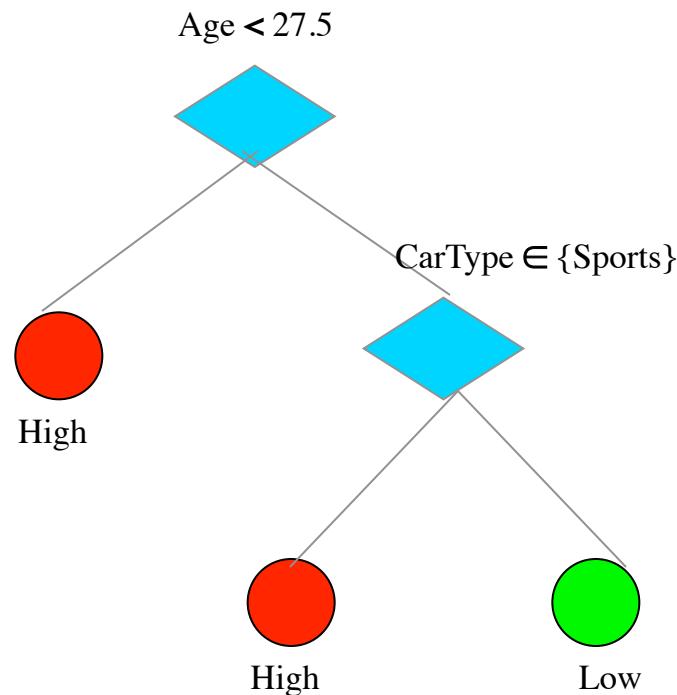
Numérique

Enumératif



$\text{Age}=40, \text{CarType}=Family \Rightarrow \text{Class}=Low$

Des arbres de décision aux règles



- 1) $\text{Age} < 27.5 \Rightarrow \text{High}$
- 2) $\text{Age} \geq 27.5 \text{ and } \text{CarType} = \text{Sports} \Rightarrow \text{High}$
- 3) $\text{Age} \geq 27.5 \text{ and } \text{CarType} \neq \text{Sports} \Rightarrow \text{Low}$

Arbres de décision – Exemple

Détection de fraudes fiscales

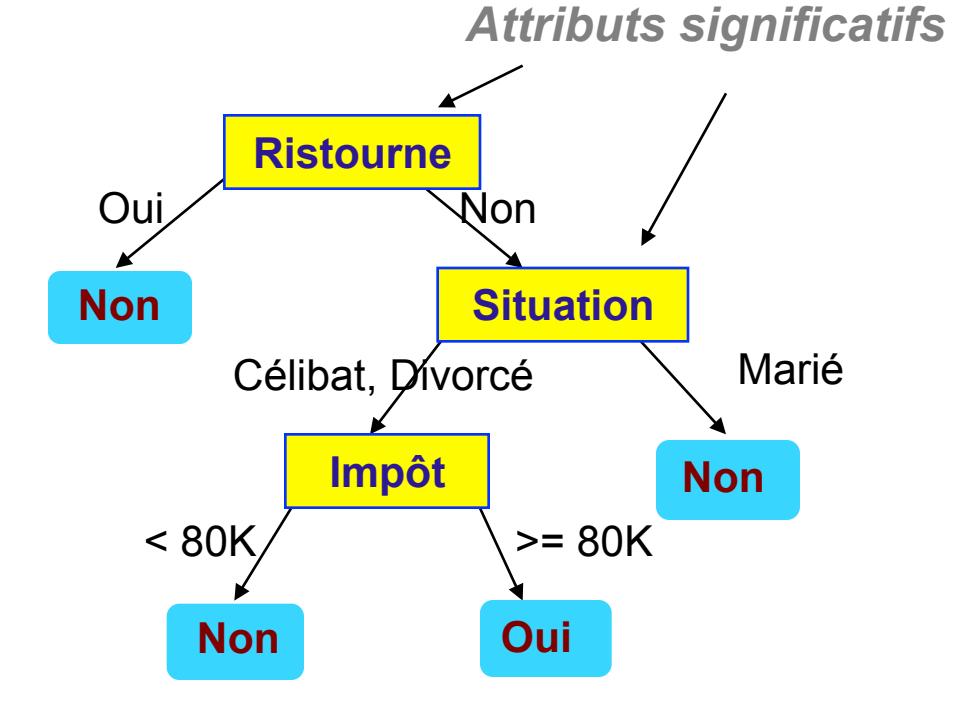
<i>Id</i>	Ristourne	Situation famille	Impôt revenu	Fraude
1	Oui	Célibat.	125K	Non
2	Non	Marié	100K	Non
3	Non	Célibat.	70K	Non
4	Oui	Marié	120K	Non
5	Non	Divorcé	95K	Oui
6	Non	Marié	60K	Non
7	Oui	Divorcé	220K	Non
8	Non	Célibat.	85K	Oui
9	Non	Marié	75K	Non
10	Non	Célibat.	90K	Oui

énumératif

énumératif

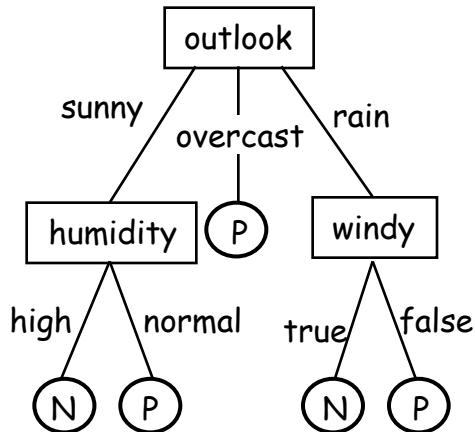
numérique

classe



- L'attribut significatif à un noeud est déterminé en se basant sur l'indice Gini.
- Pour classer une instance : descendre dans l'arbre selon les réponses aux différents tests. Ex = (Ristourne=Non, Situation=Divorcé, Impôt=100K) → Oui

De l'arbre de décision aux règles de classification



Une règle est générée pour chaque chemin de l'arbre (de la racine à une feuille)

Les paires attribut-valeur d'un chemin forment une conjonction

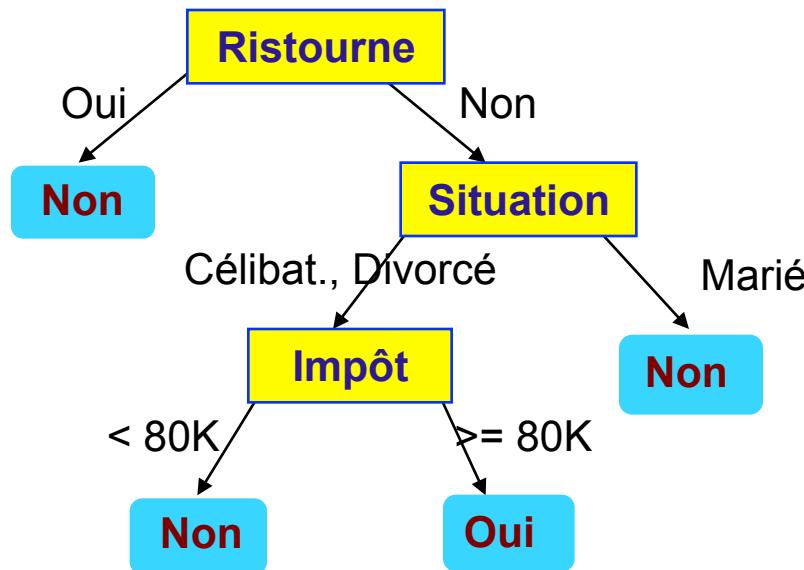
Le nœud terminal représente la classe prédite

Les règles sont généralement plus faciles à comprendre que les arbres

**Si outlook=sunny
Et humidity=normal
Alors play tennis**

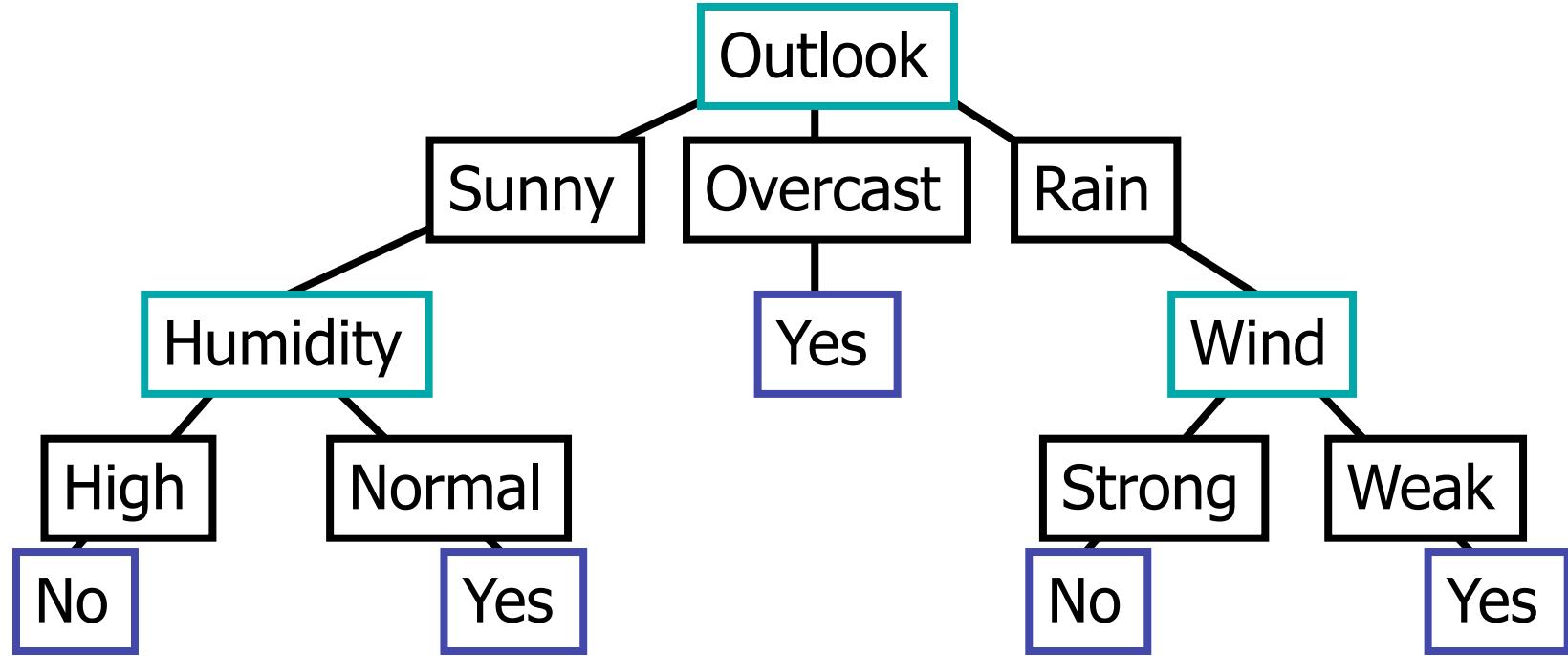
Des arbres de décision aux règles

Arbre de décision = Système de règles exhaustives et mutuellement exclusives



- 1) Ristourne = Oui \Rightarrow Non
- 2) Ristourne = Non et Situation in {Célibat., Divorcé} et Impôt $<$ 80K \Rightarrow Non
- 3) Ristourne = Non et Situation in {Célibat., Divorcé} et Impôt \geq 80K \Rightarrow Oui
- 4) Ristourne = Non et Situation in {Marié} \Rightarrow Non

Des arbres de décision aux règles



- R₁: If (Outlook=Sunny) \wedge (Humidity=High) Then PlayTennis>No
- R₂: If (Outlook=Sunny) \wedge (Humidity=Normal) Then PlayTennis>Yes
- R₃: If (Outlook=Overcast) Then PlayTennis>Yes
- R₄: If (Outlook=Rain) \wedge (Wind=Strong) Then PlayTennis>No
- R₅: If (Outlook=Rain) \wedge (Wind=Weak) Then PlayTennis>Yes

Génération de l'arbre de décision

Deux phases dans la génération de l'arbre :

1. Construction de l'arbre
 - Arbre peut atteindre une taille élevée
2. Elaguer l'arbre (Pruning)
 - Identifier et supprimer les branches qui représentent du “bruit” → Améliorer le taux d'erreur

Algorithmes de classification

Construction de l'arbre

- Au départ, toutes les instances d'apprentissage sont à la **racine** de l'arbre
- **Sélectionner** un attribut et choisir un test de séparation (**split**) sur l'attribut, qui sépare le “mieux” les instances.
- La sélection des attributs est basée sur une heuristique ou une mesure statistique.
- **Partitionner** les instances entre les noeuds fils suivant la satisfaction des tests logiques

Algorithmes de classification

- Traiter chaque nœud fils de façon récursive
- Répéter jusqu'à ce que tous les nœuds soient des **terminaux**. Un nœud courant est terminal si :
 - Il n'y a plus d'attributs disponibles
 - Le nœud est "**pur**", i.e. toutes les instances appartiennent à une seule classe,
 - Le nœud est "**presque pur**", i.e. la majorité des instances appartiennent à une seule classe (Ex : 95%)
 - Nombre minimum d'instances par branche (Ex : algorithme C5 évite la croissance de l'arbre, k=2 par défaut)
- Etiqueter le nœud terminal par la **classe majoritaire**

Algorithmes de classification

Elaguer l'arbre obtenu (pruning)

- Supprimer les sous-arbres qui n'améliorent pas l'erreur de la classification (accuracy) → arbre ayant un meilleur pouvoir de **généralisation**, même si on augmente l'erreur sur l'ensemble d'apprentissage
- Eviter le problème de **sur-spécialisation (over-fitting)**, i.e., on a appris “par cœur” l'ensemble d'apprentissage, mais on n'est pas capable de généraliser

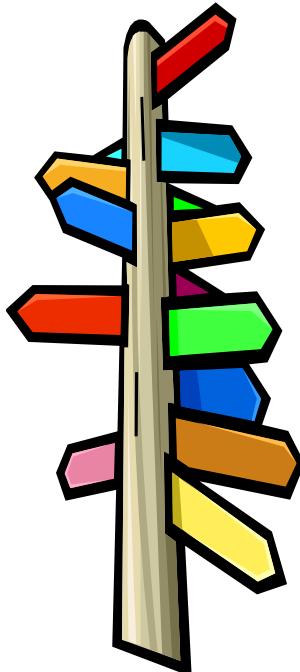
Sur-spécialisation - arbre de décision

L'arbre généré peut sur-spécialiser l'ensemble d'apprentissage

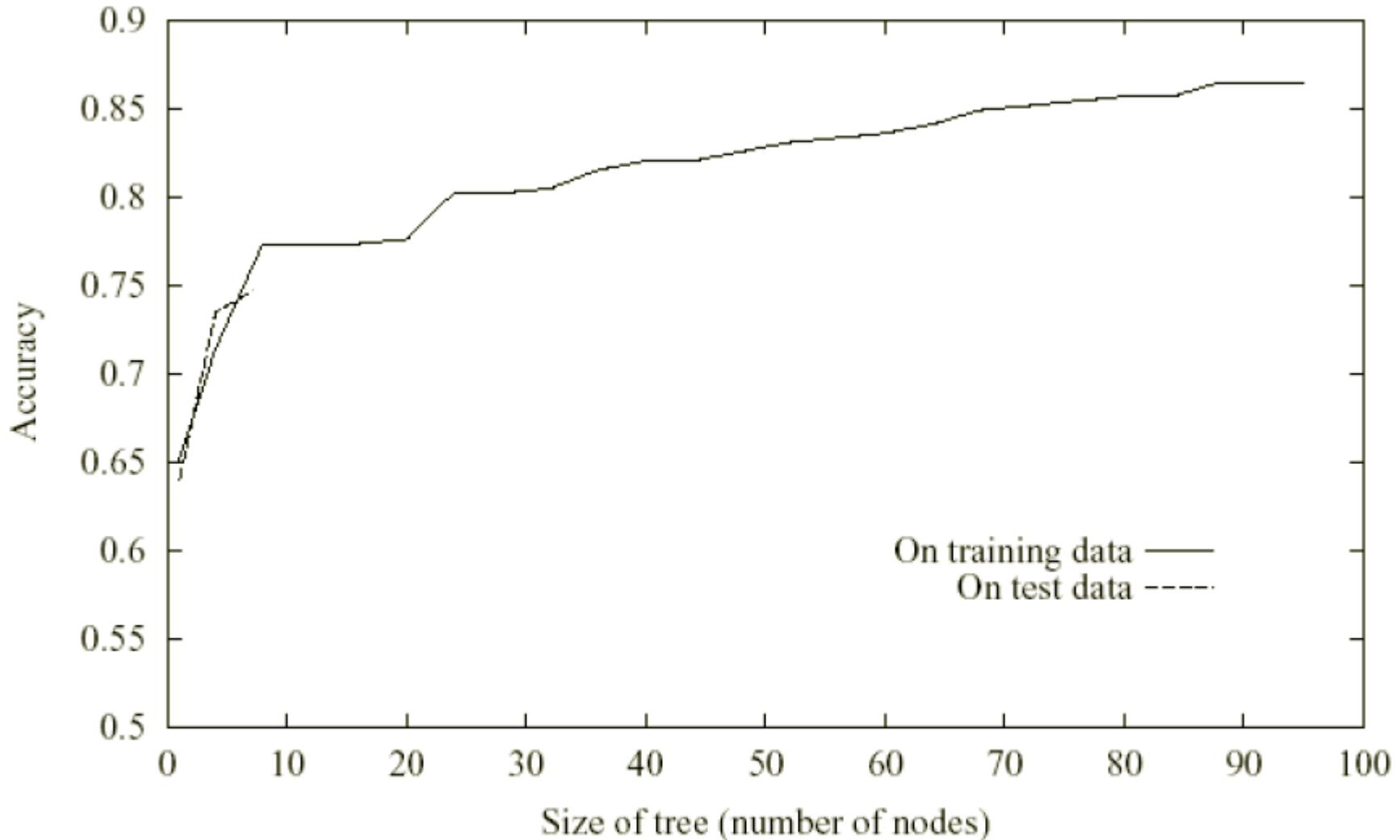
- Plusieurs branches
- Taux d'erreur important pour les instances inconnues

Raisons de la sur-spécialisation

- bruits et exceptions
- Peu de donnée d'apprentissage
- Maxima locaux dans la recherche gloutonne

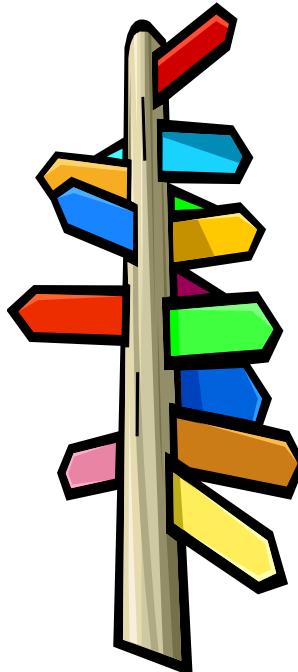


Overfitting dans les arbres de décision



Comment éviter l'overfitting ?

Deux approches :



Pré-élagage : Arrêter de façon prématuée la construction de l'arbre

Post-élagage : Supprimer des branches de l'arbre complet ("fully grown")

Convertir l'arbre en règles ; élaguer les règles de façon indépendante (C4.5)

Construction de l'arbre - Synthèse

Evaluation des différents branchements pour tous les attributs

Sélection du “meilleur” branchement “et de l’attribut “gagnant”

Partitionner les données entre les fils

Construction en largeur (C4.5) ou en profondeur (SPLIT)

Questions critiques :

- Formulation des tests de branchement
- Mesure de sélection des attributs

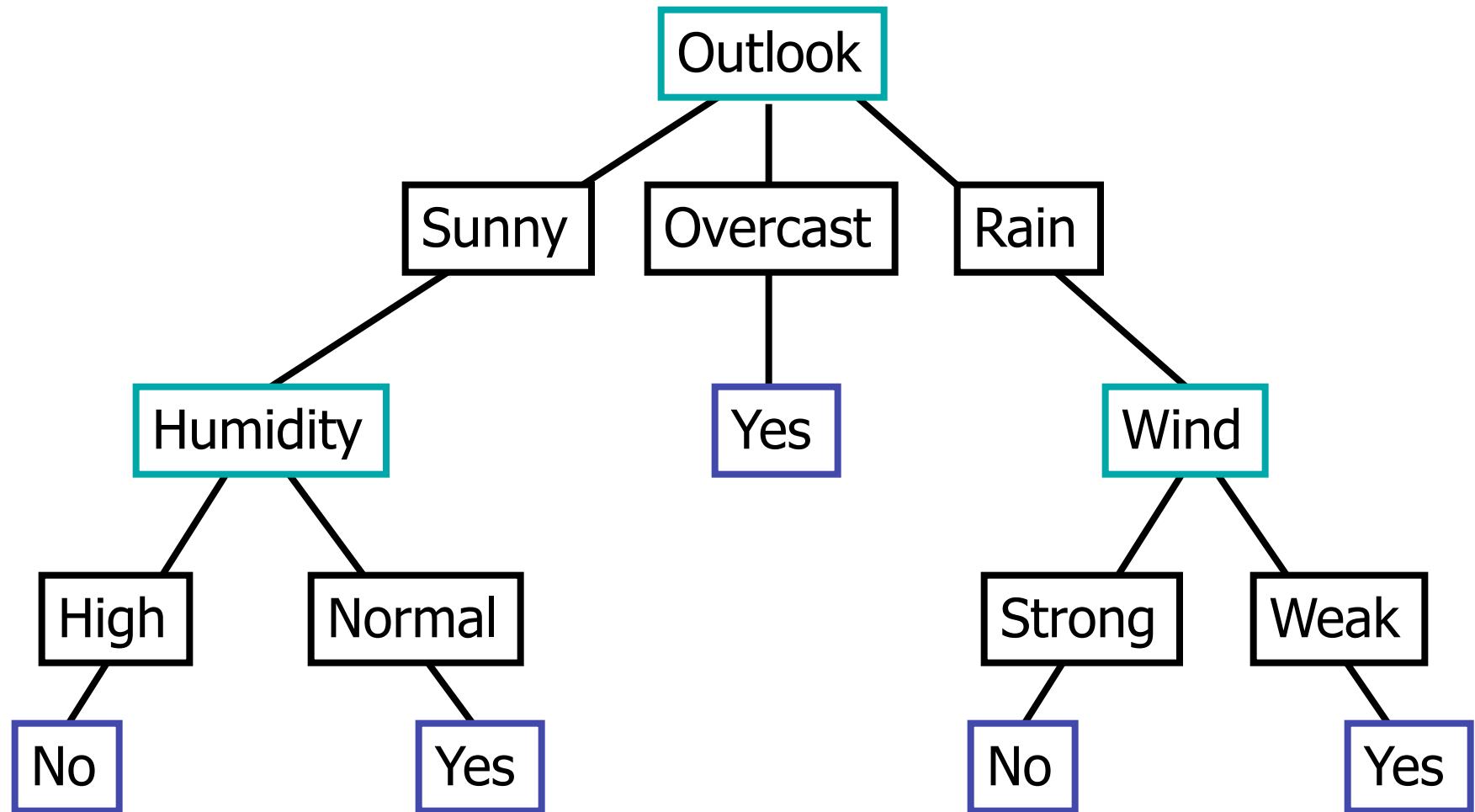
Exemple : Jouer au tennis ?

Ensemble
d'apprentissage

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

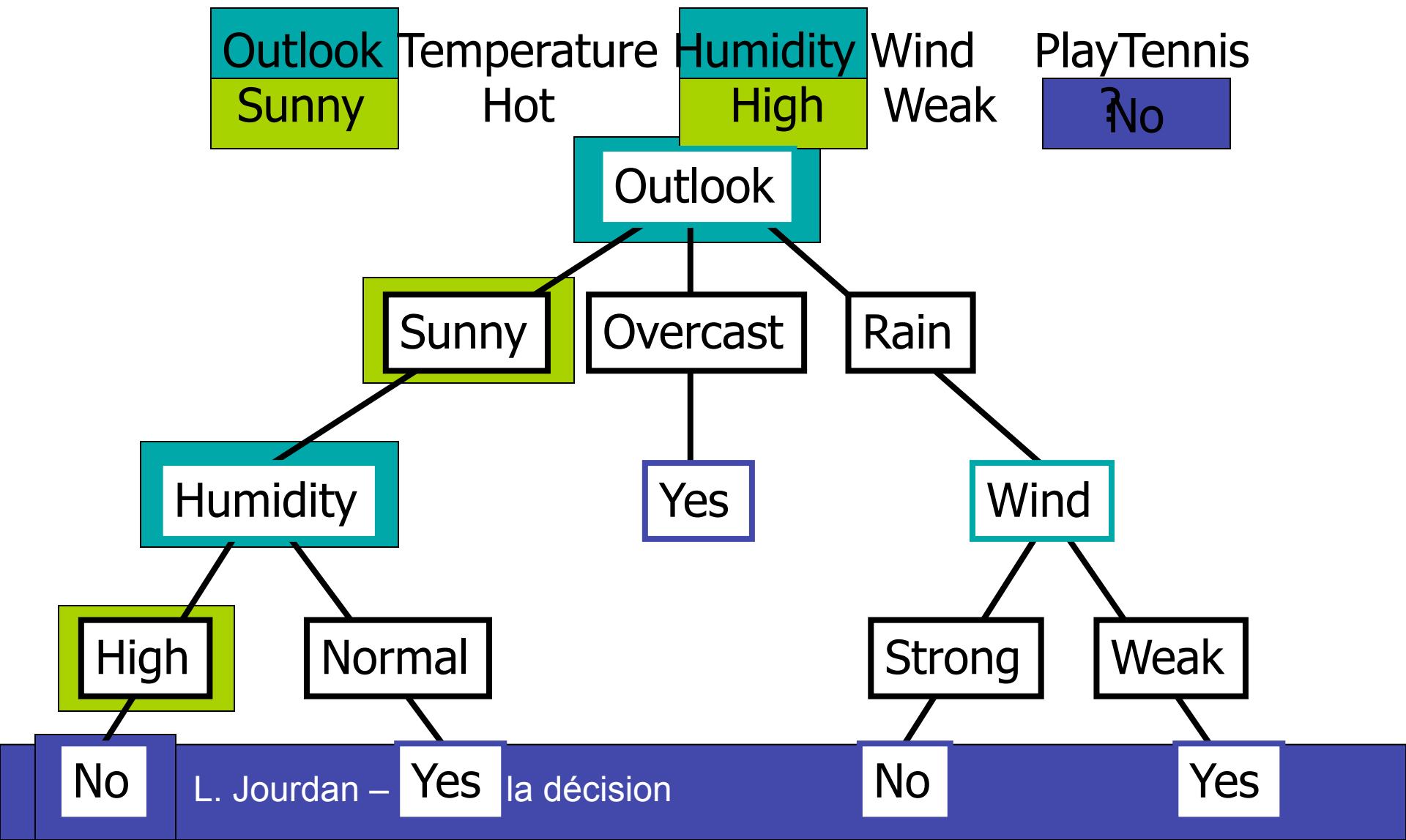
Arbre de décision obtenu avec ID3 (Quinlan 86)

16
7



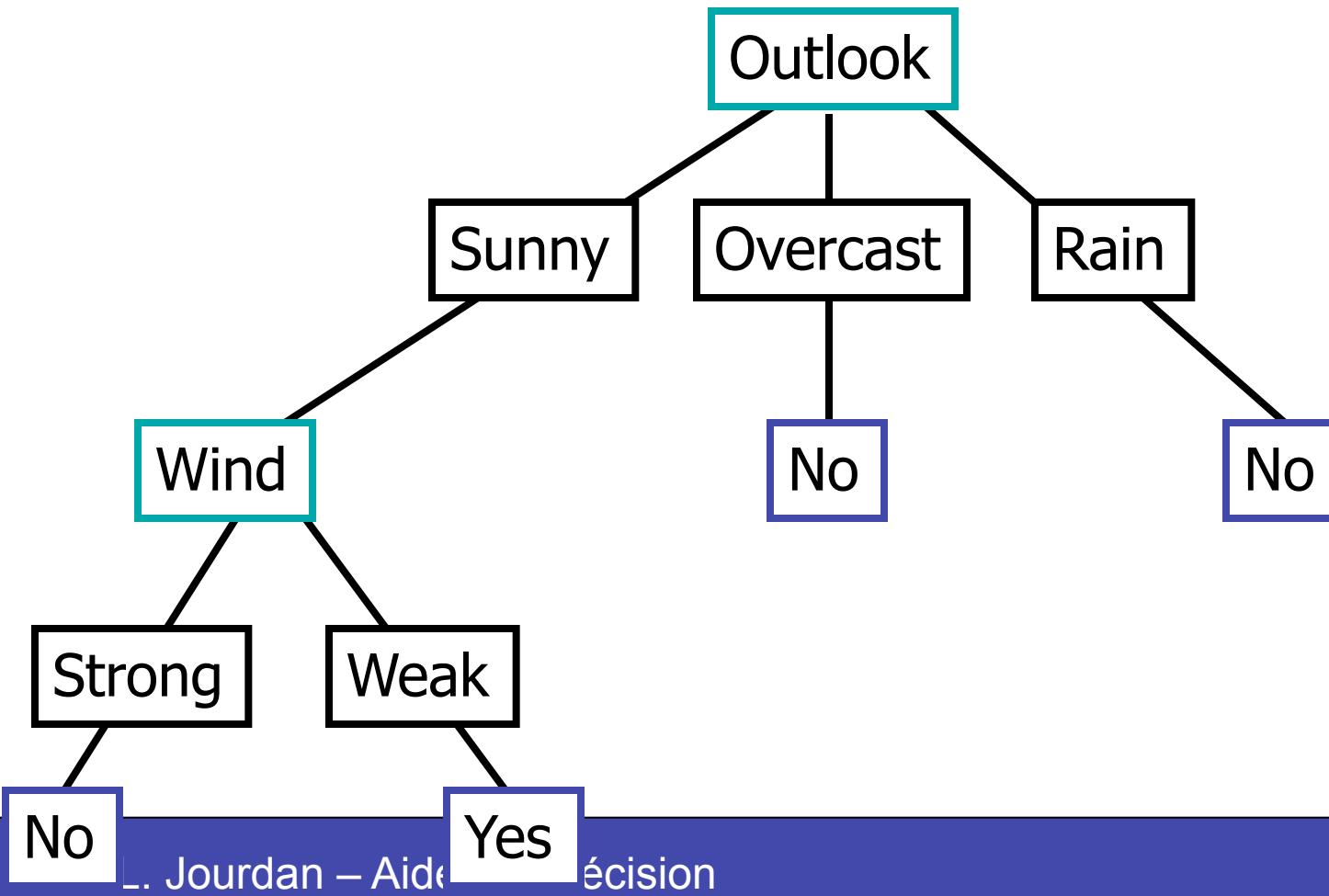
Arbre de décision obtenu avec ID3 (Quinlan 86)

16
8



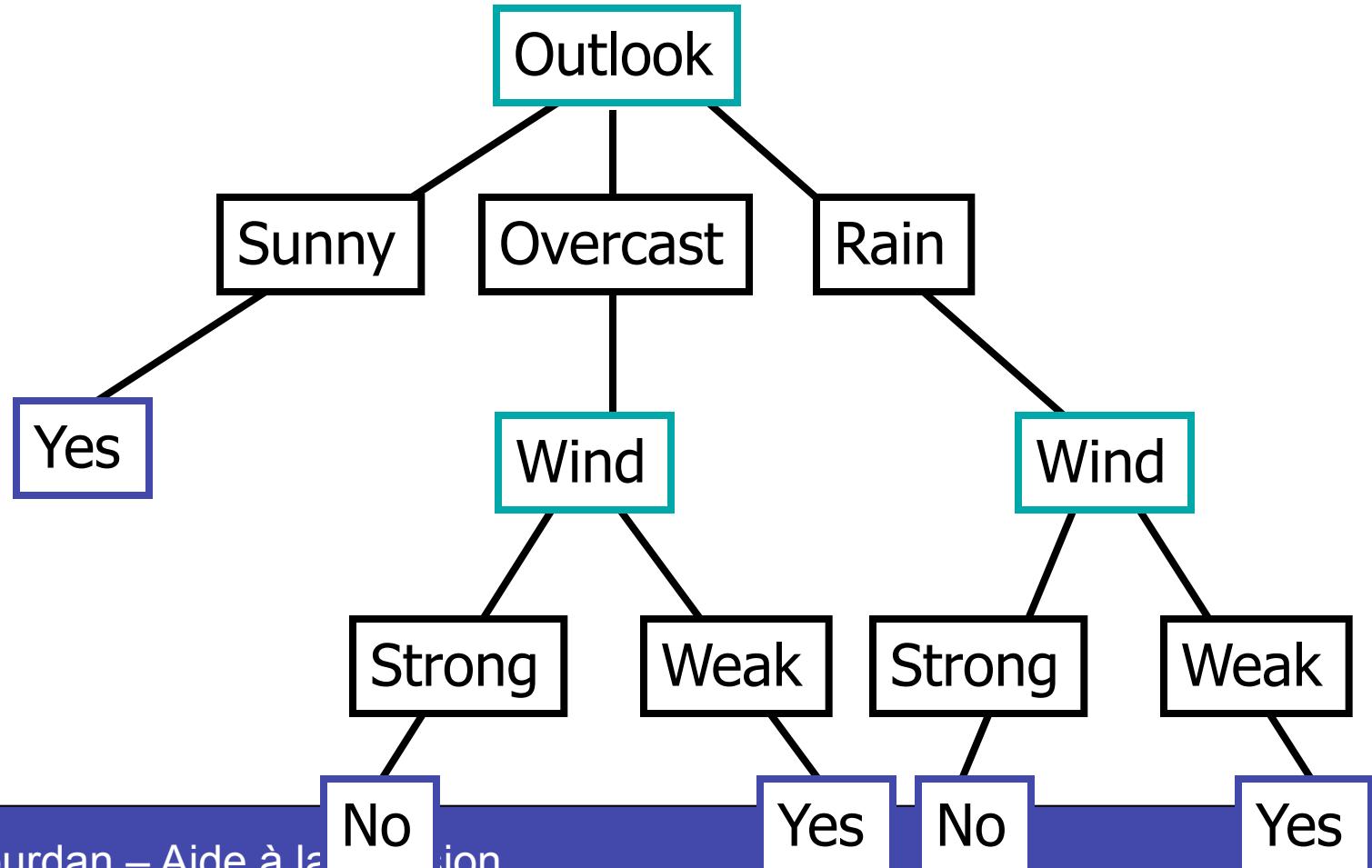
Arbre de décision et conjonction

Outlook=Sunny \wedge Wind=Weak



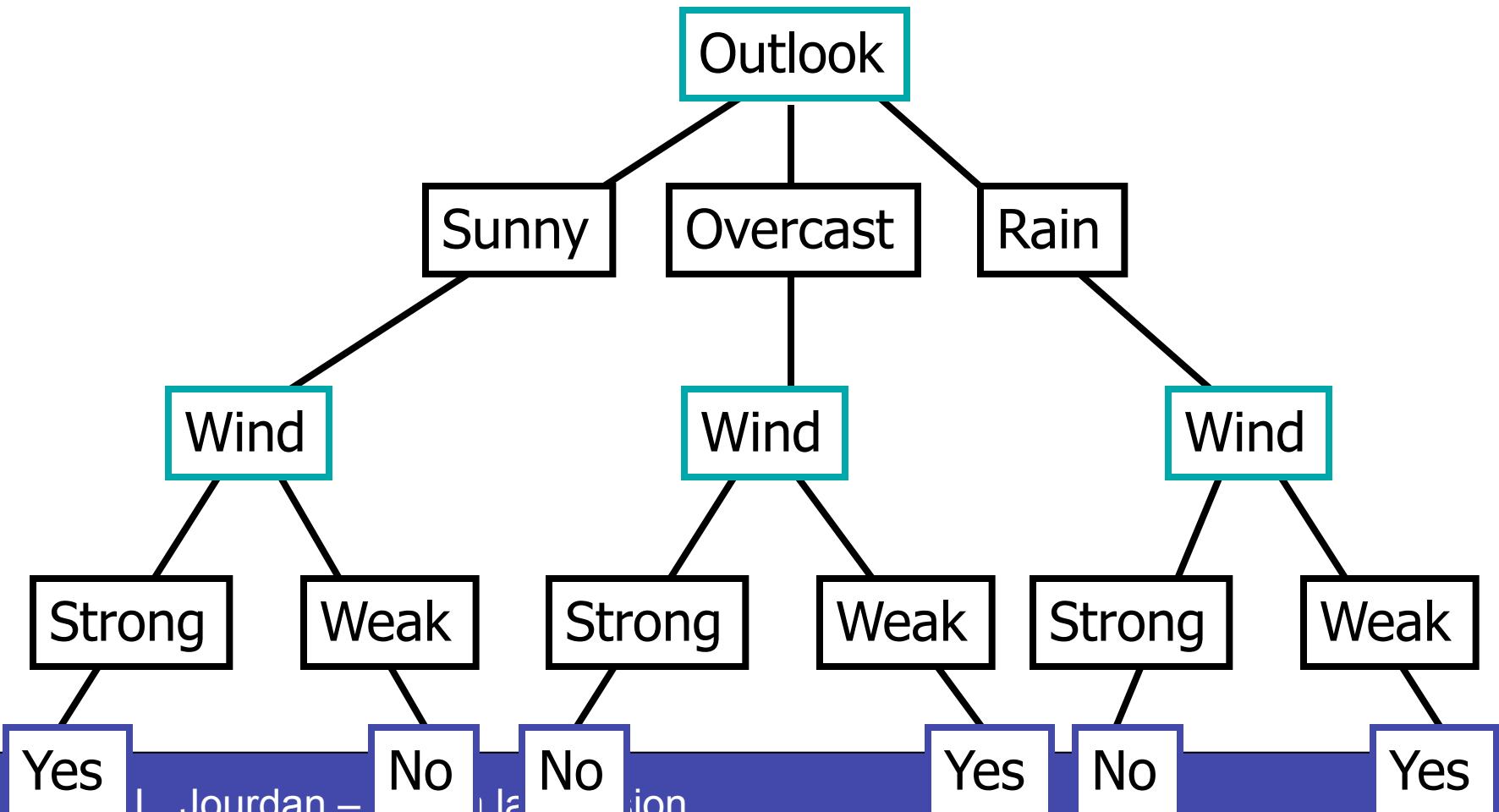
Arbre de décision et disjonction

$\text{Outlook} = \text{Sunny} \vee \text{Wind} = \text{Weak}$



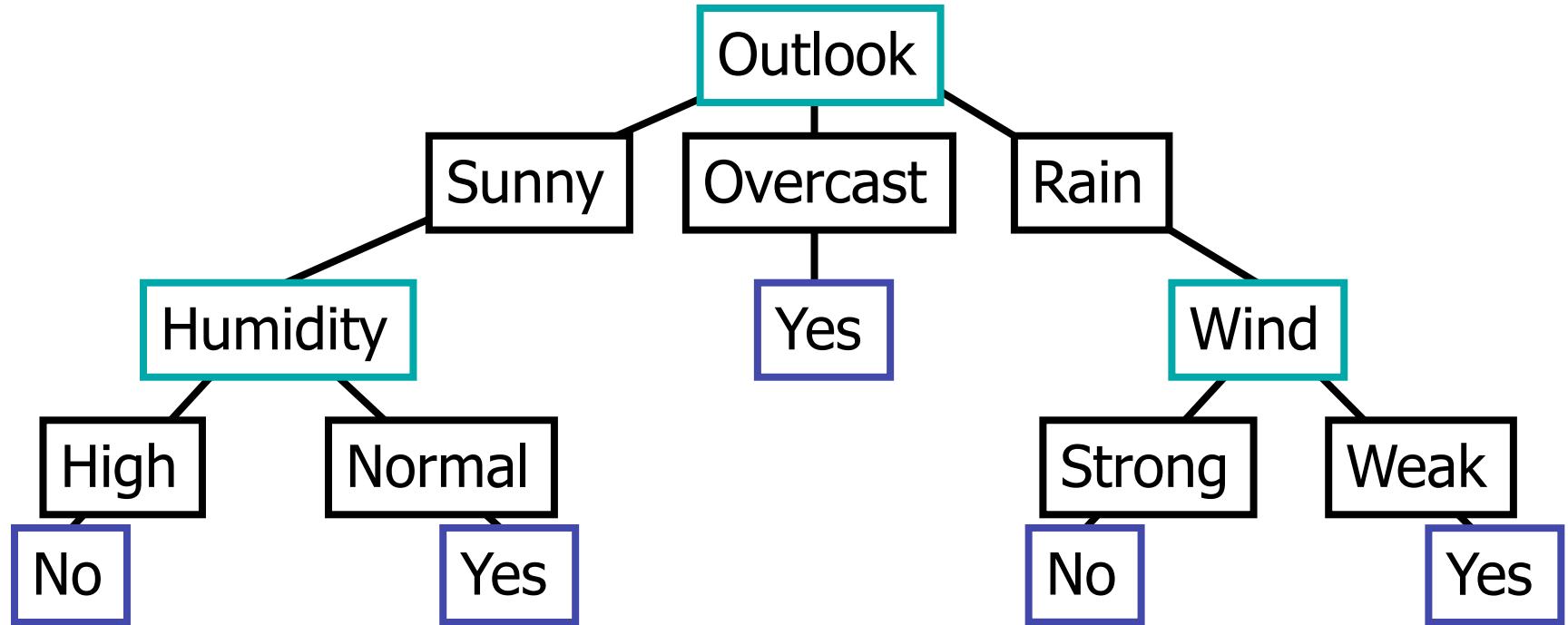
Arbre de décision et XOR

Outlook=Sunny XOR Wind=Weak



Arbre de décision et conjonction

- arbre de décision représente des disjonctions de conjonctions



$(\text{Outlook}=\text{Sunny} \wedge \text{Humidity}=\text{Normal})$
v
 $(\text{Outlook}=\text{Overcast})$

v
 $(\text{Outlook}=\text{Rain} \wedge \text{Wind}=\text{Weak})$

Algorithmes pour les arbres de décision

Algorithme de base

- Construction récursive d'un arbre de manière "diviser-pour-régner" descendante
- Attributs considérés énumératifs
- Glouton (piégé par les optima locaux)

Plusieurs variantes : ID3, C4.5, CART, CHAID

- Différence principale : mesure de sélection d'un attribut – critère de branchement (split)
- Ex : CART : 2 partitions par nœuds

Mesures de sélection d'attributs

Gain d'Information (ID3, C4.5, C5)

Indice Gini (CART)

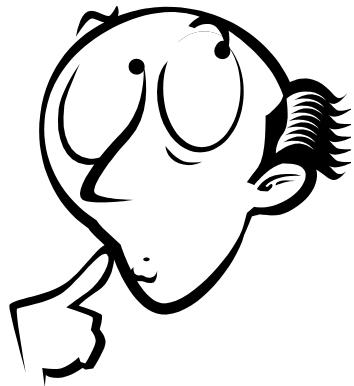
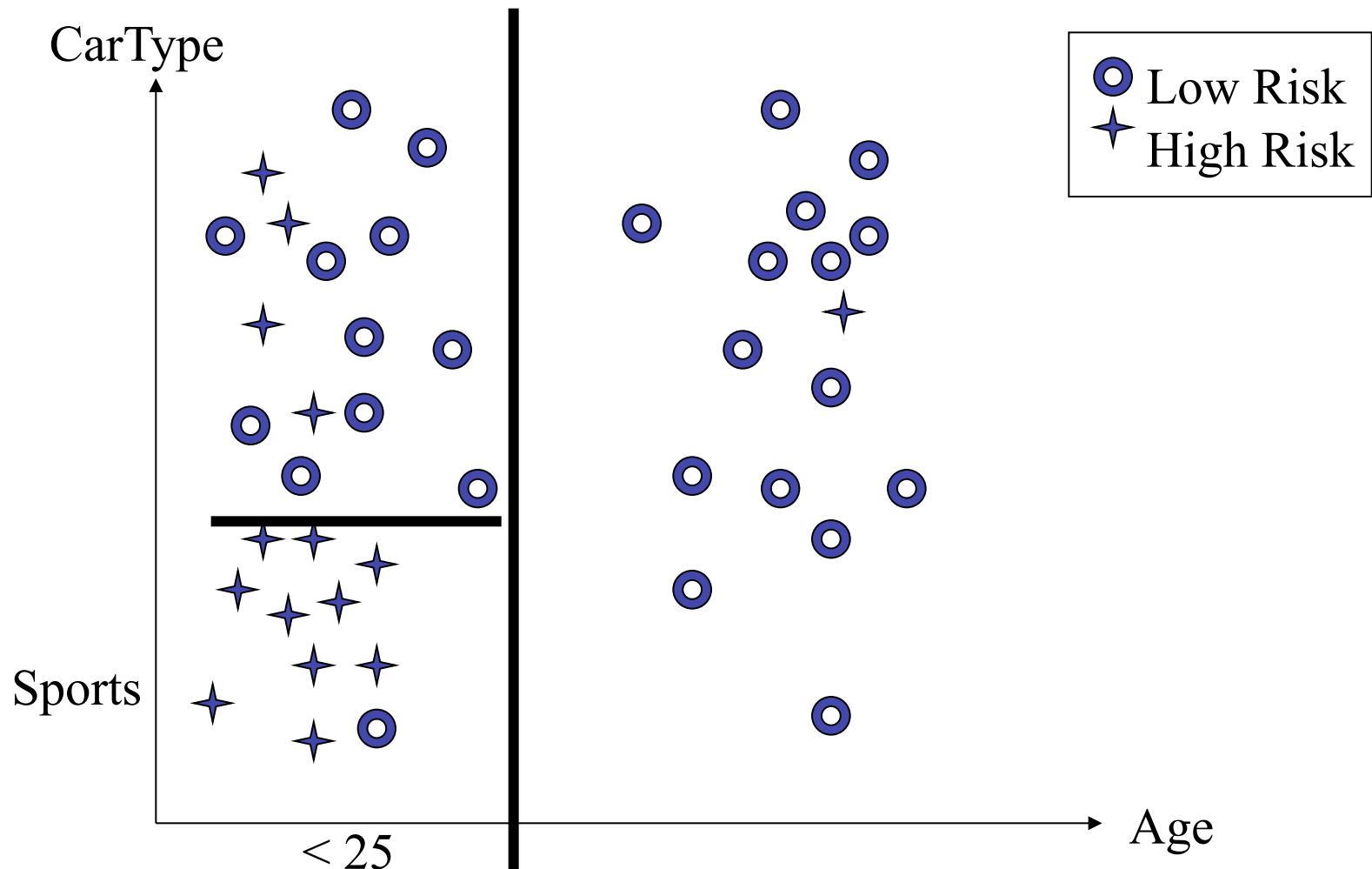


Table de contingence statistique χ^2 (CHAID)

G-statistic

Bonne sélection et branchement ?



Gain d'information

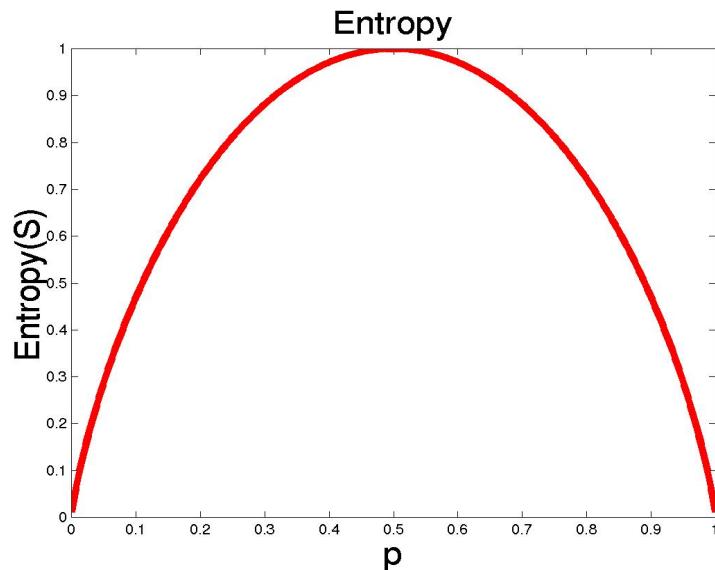
Sélectionner l'attribut avec le plus grand gain d'information

Soient P et N deux classes et S un ensemble d'instances avec p éléments de P et n éléments de N

L'information nécessaire pour déterminer si une instance prise au hasard fait partie de P ou N est (entropie) :

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

Entropie



Ex : var. booléenne $X=1$
Avec probabilité p

S est l'ensemble d'apprentissage

p_+ est la proportion d'exemples positifs (P)

p_- est la proportion d'exemples négatifs (N)

Entropie mesure l'impureté de S

- $\text{Entropie}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$

Gain d'information

Soient les ensembles $\{S_1, S_2, \dots, S_v\}$ formant une partition de l'ensemble S , en utilisant l'attribut A

Toute partition S_i contient p_i instances de P et n_i instances de N

L'entropie, ou l'information nécessaire pour classifier les instances dans les sous-arbres S_i est (entropie conditionnelle classe/attribut A):

$$E(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

Le gain d'information par rapport au branchement sur A est

$$Gain(A) = I(p, n) - E(A)$$

Choisir l'attribut qui maximise le gain \rightarrow besoin d'information minimal (recherche “greedy” – gloutonne)

Gain d'information - Exemple

Hypothèses :

Classe P : jouer_tennis = “oui”

Classe N : jouer_tennis = “non”

Information nécessaire pour classer un exemple donné est :

$$I(p,n) = I(9,5) = 0.940$$

Gain d'information - Exemple

Calculer l'entropie pour l'attribut outlook :

outlook	p_i	n_i	$I(p_i, n_i)$
sunny	2	3	0,971
overcast	4	0	0
rain	3	2	0,971

On a $E(outlook) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$

Alors $Gain(outlook) = I(9,5) - E(outlook) = 0.246$

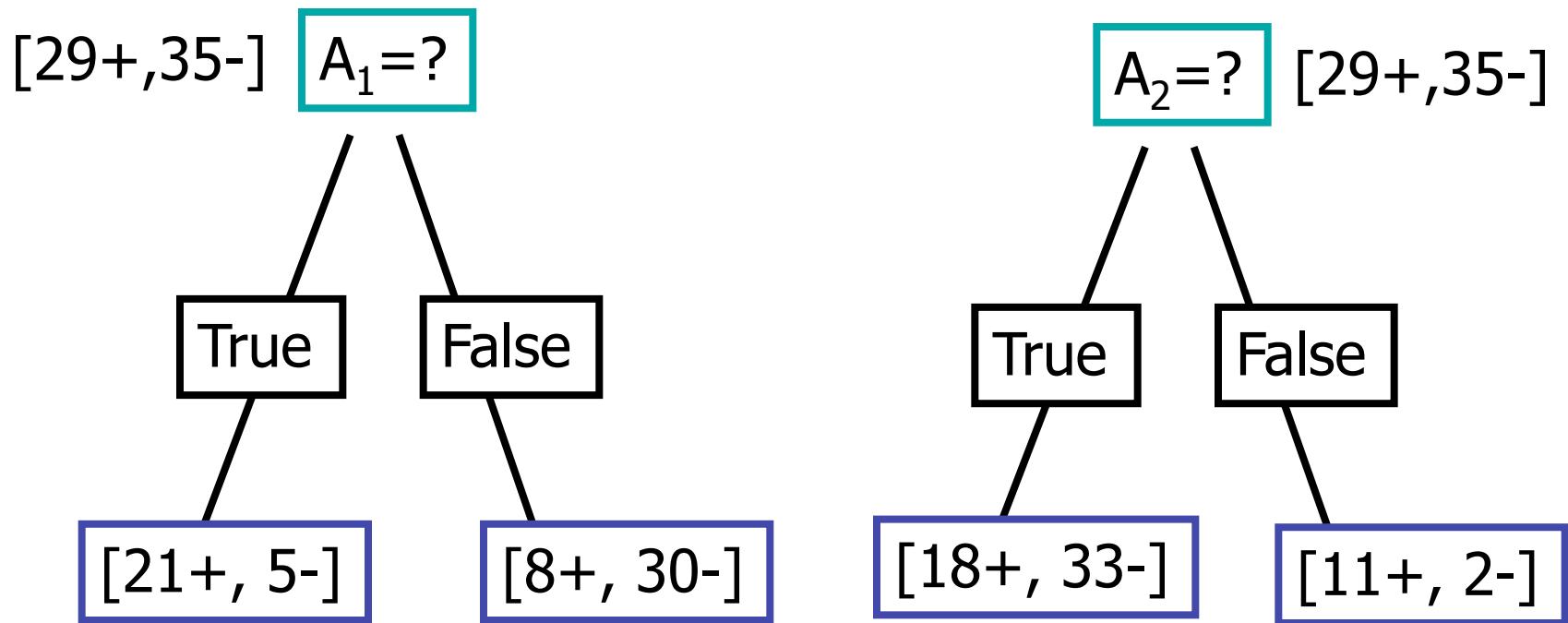
De manière similaire

$$Gain(temperature) = 0.029$$

$$Gain(humidity) = 0.151$$

$$Gain(windy) = 0.048$$

Quel Attribut est "meilleur" ?

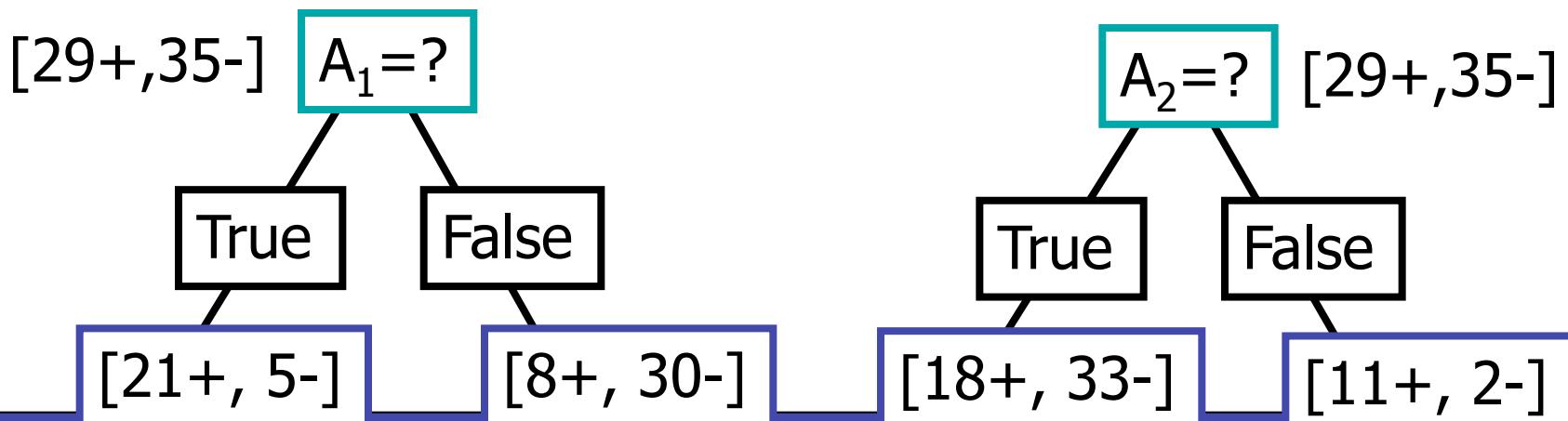


Gain d'information - Exemple

$\text{Gain}(S,A)$: réduction attendue de l'entropie dûe au branchement de S sur l'attribut A

$$\text{Gain}(S,A) = \text{Entropie}(S) - \sum_{v \in \text{values}(A)} |S_v|/|S| \text{ Entropie}(S_v)$$

$$\begin{aligned} \text{Entropie}([29+, 35-]) &= -29/64 \log_2 29/64 - 35/64 \log_2 35/64 \\ &= 0.99 \end{aligned}$$



Gain d'information - Exemple

$$\text{Entropie}([21+, 5-]) = 0.71$$

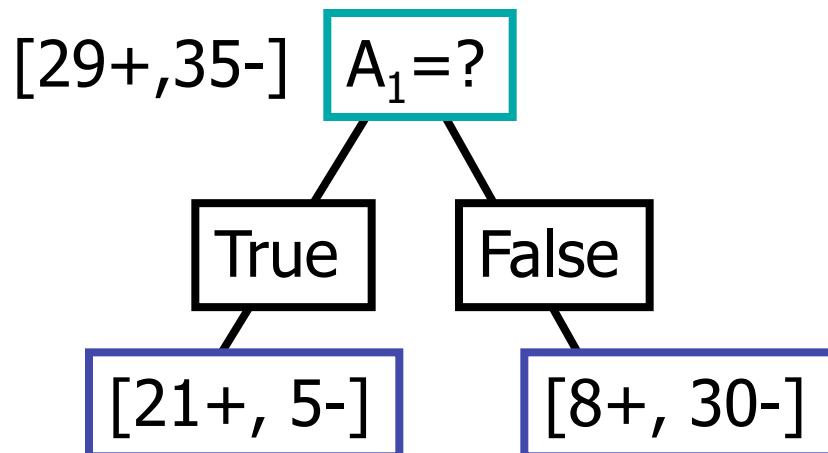
$$\text{Entropie}([8+, 30-]) = 0.74$$

$$\text{Gain}(S, A_1) = \text{Entropie}(S)$$

$$-26/64 * \text{Entropie}([21+, 5-])$$

$$-38/64 * \text{Entropie}([8+, 30-])$$

$$= 0.27$$



$$\text{Entropie}([18+, 33-]) = 0.94$$

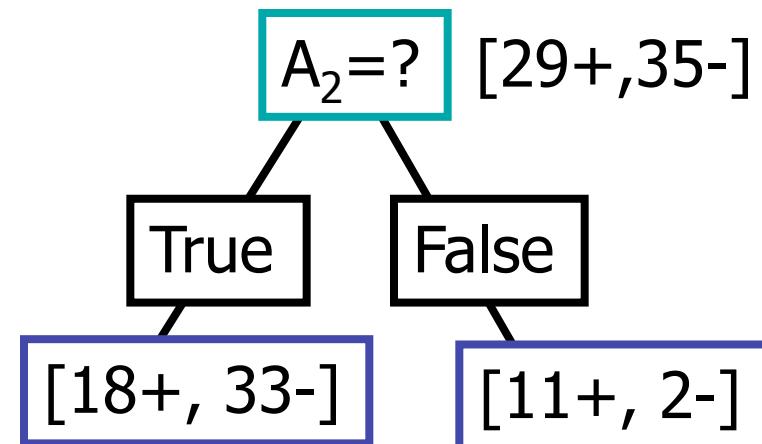
$$\text{Entropie}([11+, 2-]) = 0.62$$

$$\text{Gain}(S, A_2) = \text{Entropie}(S)$$

$$-51/64 * \text{Entropie}([18+, 33-])$$

$$-13/64 * \text{Entropie}([11+, 2-])$$

$$= 0.12$$



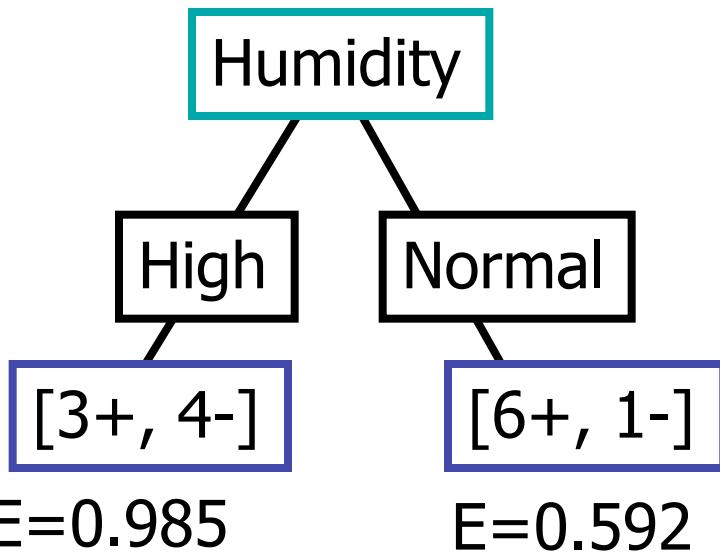
Exemple d'apprentissage

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Sélection de l'attribut suivant

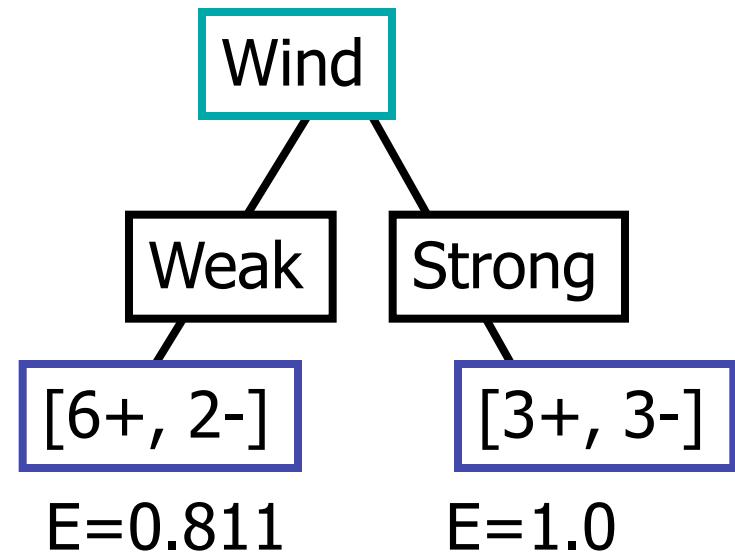
$$S=[9+, 5-]$$

$$E=0.940$$



$$S=[9+, 5-]$$

$$E=0.940$$



$$\text{Gain}(S, \text{Humidity})$$

$$=0.940 - (7/14) * 0.985$$

$$- (7/14) * 0.592$$

$$=0.151$$

$$\text{Gain}(S, \text{Wind})$$

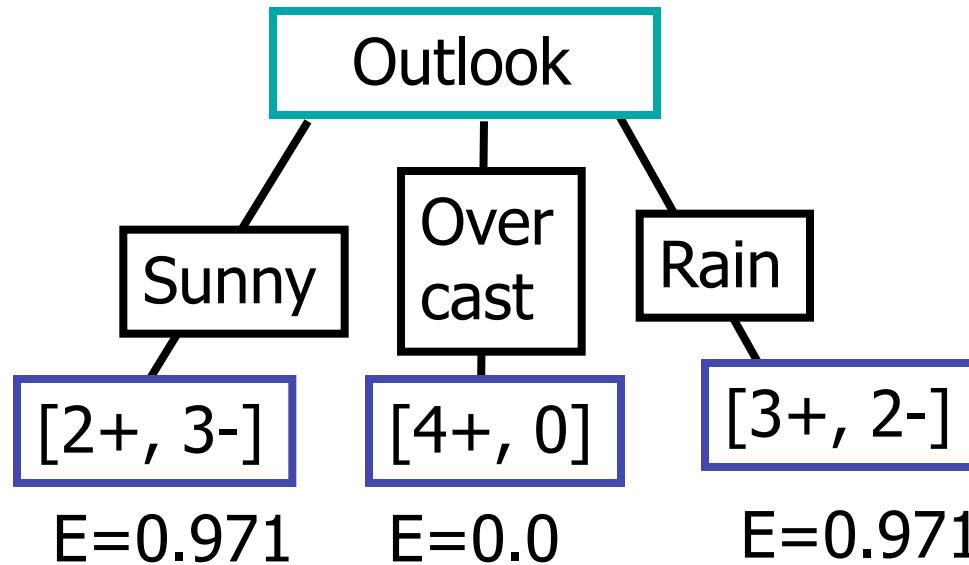
$$=0.940 - (8/14) * 0.811$$

$$- (6/14) * 1.0$$

$$=0.048$$

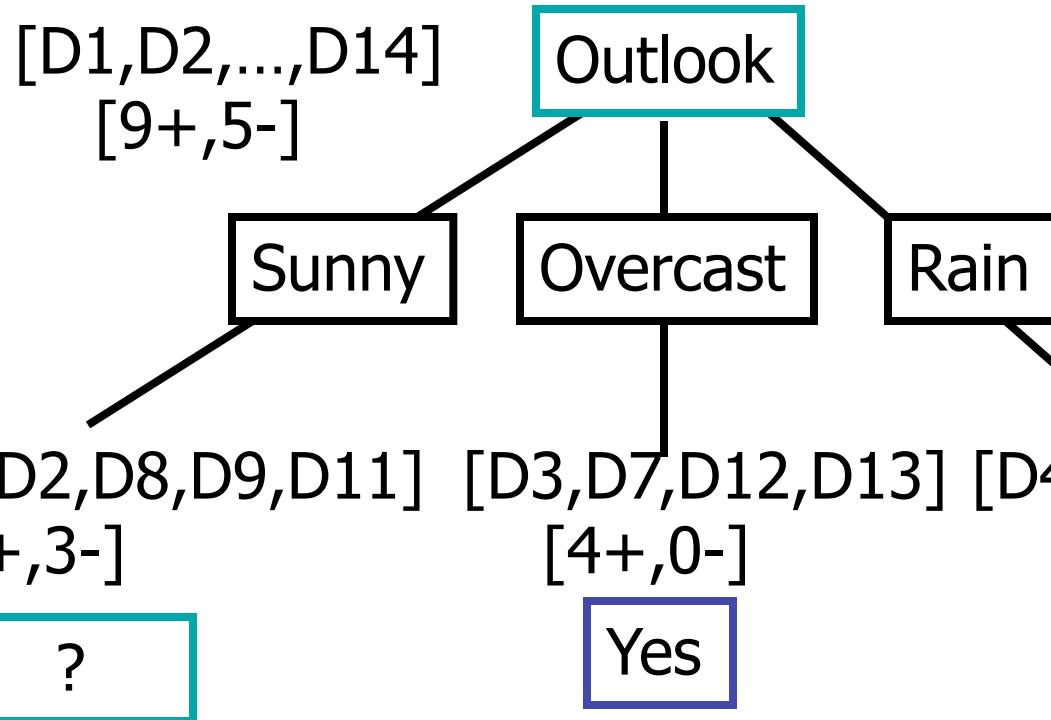
Sélection de l'attribut suivant

$$S=[9+, 5-]$$
$$E=0.940$$



$$\begin{aligned} \text{Gain}(S, \text{Outlook}) &= 0.940 - (5/14) * 0.971 \\ &\quad - (4/14) * 0.0 - (5/14) * 0.971 \\ &= 0.247 \end{aligned}$$

Algorithme ID3



$$S_{\text{sunny}} = [D1, D2, D8, D9, D11] \quad [D3, D7, D12, D13] \quad [D4, D5, D6, D10, D14]$$

$$\quad [2+, 3-] \qquad \qquad [4+, 0-] \qquad \qquad [3+, 2-]$$

?

Yes

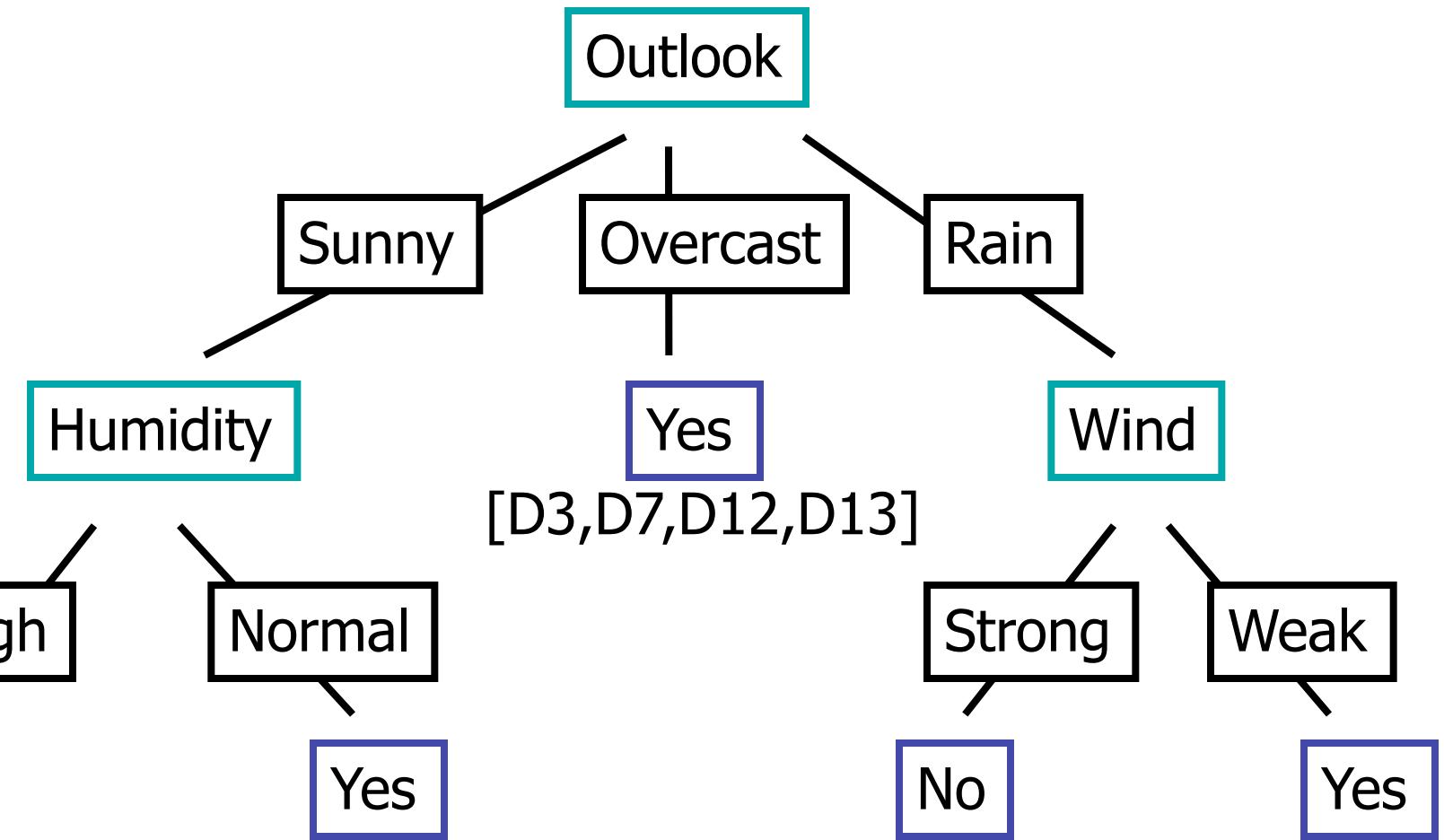
?

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = 0.970 - (3/5)0.0 - 2/5(0.0) = 0.970$$

$$\text{Gain}(S_{\text{sunny}}, \text{Temp.}) = 0.970 - (2/5)0.0 - 2/5(1.0) - (1/5)0.0 = 0.570$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = 0.970 = -(2/5)1.0 - 3/5(0.918) = 0.019$$

Algorithme ID3



Indice Gini

Utiliser l'indice Gini pour un partitionnement pur

$$Gini(S) = 1 - \sum_{i=1}^c p_i^2$$

$$Gini(S_1, S_2) = \frac{n_1}{n} Gini(S_1) + \frac{n_2}{n} Gini(S_2)$$

pi est la fréquence relative de la classe c dans S

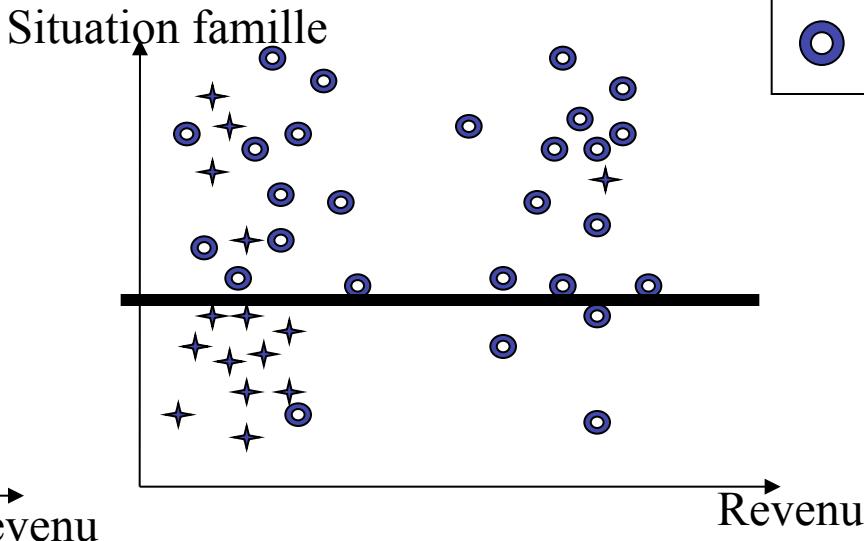
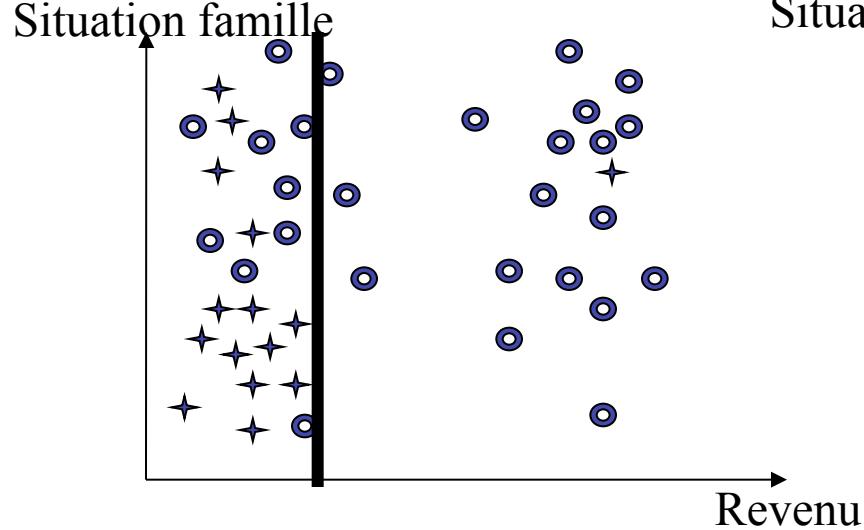
Si S est pur (classe unique), $Gini(S) = 0$

Gini(S1,S2) = Gini pour une partition de S en deux sous-ensembles S1 et S2 selon un test donné.

Trouver le branchement (split-point) qui **minimise** l'indice Gini

Nécessite seulement les distributions de classes

Indice Gini - Exemple



Fraude
Pas fraude

Calcul de Gini nécessite une **Matrice de dénombrement**

	Non	Oui
<80K	14	9
>80K	1	18

$$\text{Gini(split)} = \mathbf{0.31}$$

	Non	Oui
M	5	23
F	10	4

$$\text{Gini(split)} = \mathbf{0.34}$$

Attributs énumératifs – indice GIN

CarType		
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

Partage en plusieurs classes

CarType			
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

- Pour chaque valeur distincte, calculer le nombre d'instances de chaque classe
- Utiliser la **matrice de dénombrement** pour la prise de décision

Partage en deux “classes”
(trouver la meilleure partition de valeurs)

CarType		
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Attributs numériques – indice GIN

calcul efficace : pour chaque attribut,

- Trier les instances selon la valeur de l'attribut
- Entre chaque valeur de cette liste : un test possible (split)
- Evaluation de Gini pour chacun des test
- Choisir le split qui minimise l'indice gini

Fraude	No	No	No	Yes	Yes	Yes	No	No	No	No	
Revenu imposable											
Valeurs triées →	60	70	75	85	90	95	100	120	125	220	
Positions Split →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400	

Algorithme CART

Indice de Gini

$$I = 1 - \sum_i^n f_i^2$$

- N = nombre de classes à prédire
- Fi = fréquence de la classe i dans le nœud

Plus l'indice de Gini est bas, plus le nœud est pure

Algorithme CART

Problèmes des arbres trop étoffés

- Complexité de l'arbre, trop de règles
- Trop spécifique aux données d'apprentissage
 - Règles non reproductibles (« surapprentissage »)
- Trop peu d'individus dans les feuilles (aucune signification réelle)
 - minimum conseillé : 20-30 individus

Solution → Élagage

Algorithme CART

Processus d'élagage de CART

- Création de l'arbre maximum
 - Toutes les feuilles des extrémités sont pures
- Élagages successifs de l'arbre
- Retient l'arbre élagué pour lequel le taux d'erreur estimé mesuré sur un échantillon test est le plus bas possible

Avantages

Résultats explicites

- Arbre
- Règles de décisions simples
- Modèle facilement programmable pour affecter de nouveaux individus

Peu de perturbation des individus extrêmes

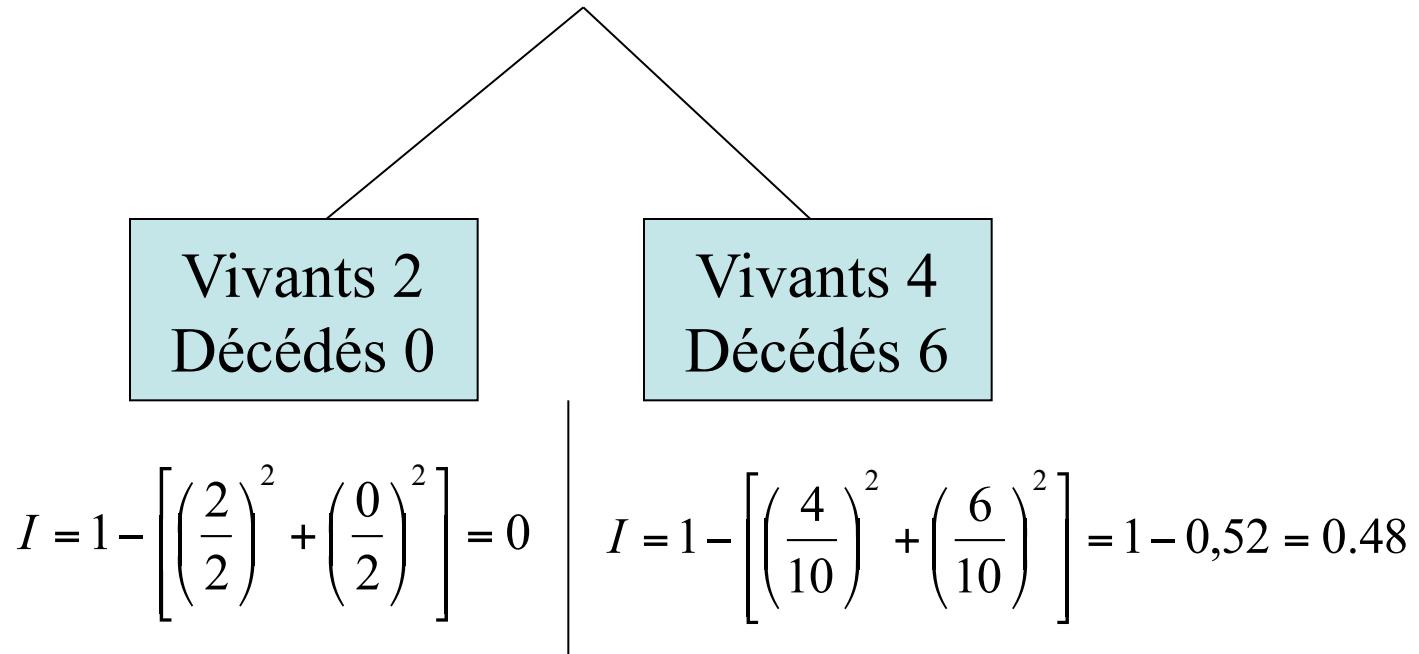
- Isolés dans des petites feuilles

Peu sensible au bruit des variables non discriminantes

- Non introduites dans le modèle

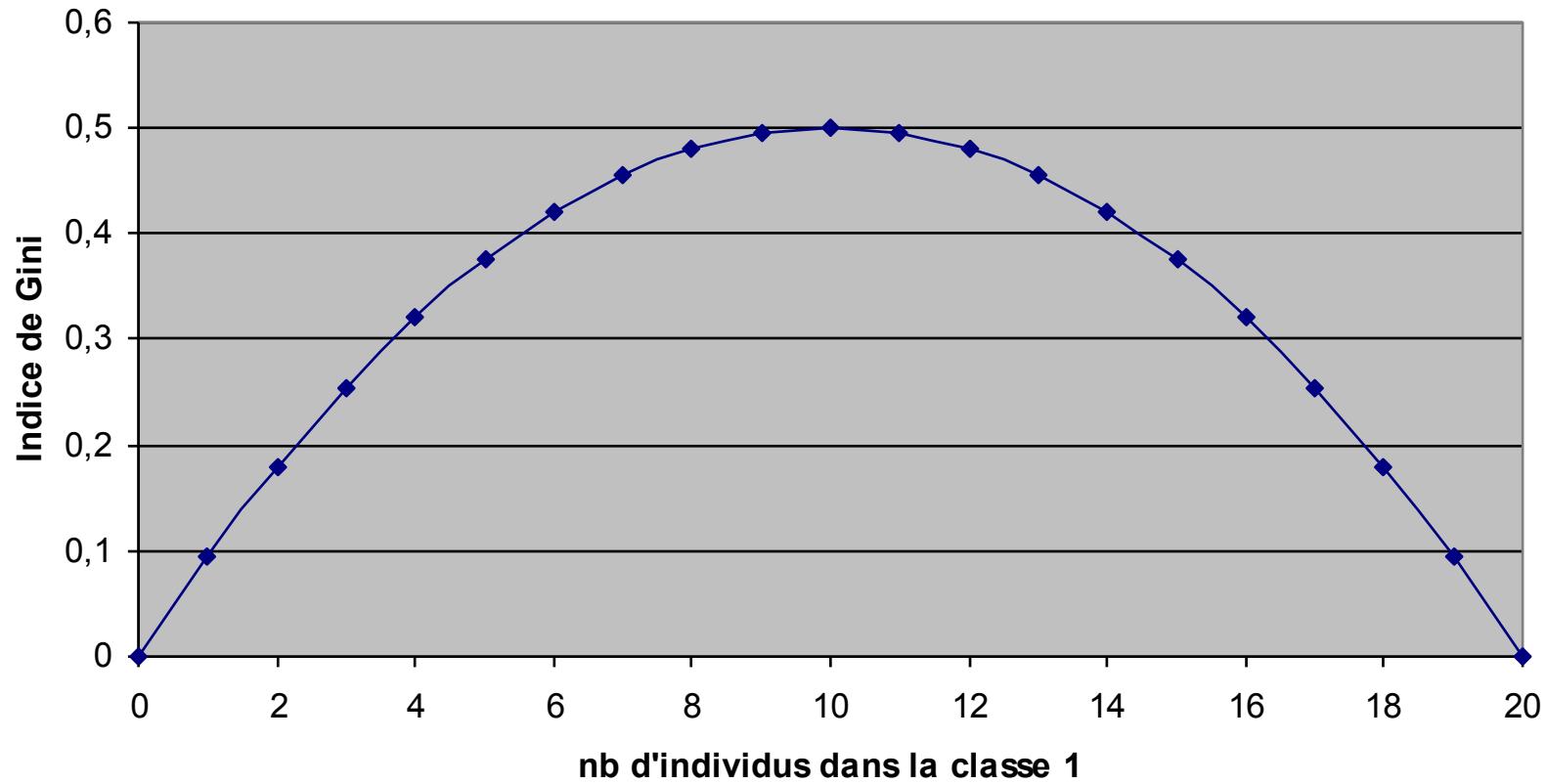
Algorithme CART

Exemple :



Algorithme CART

Indice de Gini (20 individus et 2 classes)



Algorithme CART

Ainsi,

- En séparant 1 nœud en 2 nœuds fils on cherche la plus grande hausse de la pureté
- La variable la plus discriminante doit maximiser :

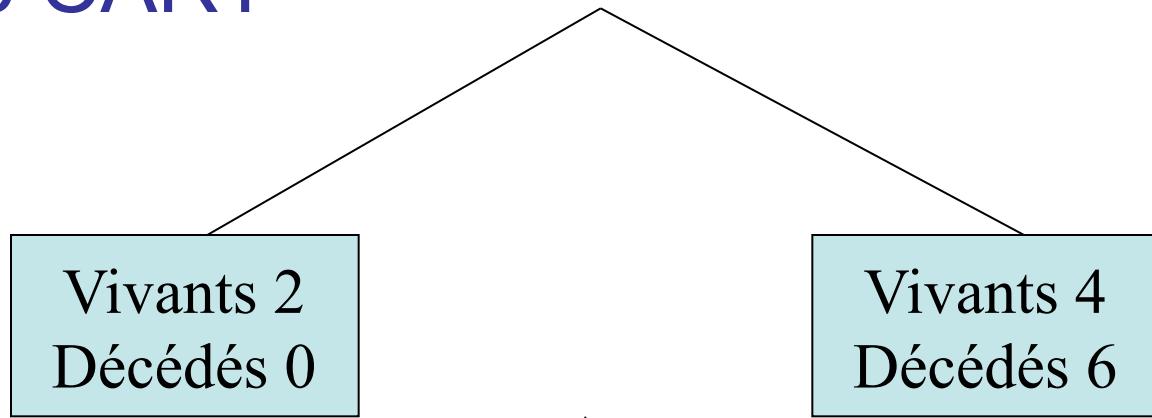
$$\text{IG(avant sep.)}-[\text{IG(fils1)}+\text{IG(fils2)}]$$

Algorithme CART

Répartition des individus dans les nœuds

- Quand l'arbre est construit : critères de division connus
- On affecte chaque individu selon les règles obtenues → remplissage des feuilles
 - Pour chaque feuille : plusieurs classes C
 - P_c = Proportion d'individus de la feuille appartenant à la classe c
 - On affecte à la feuille la classe pour laquelle P_c est la plus grande

Algorithme CART



$P(\text{vivants})=1$

$P(\text{décédés})=0$

→ Feuille « vivants »

Taux d'erreur feuille = 0

$P(\text{vivants})=0,4$

$P(\text{décédés})=0,6$

→ Feuille « Décédés »

Taux d'erreur feuille = 0,4

Taux d'erreur global de l'arbre = somme pondérée des taux d'erreur des feuilles

Pondération = proba qu'un individu soit dans la feuille (= taille de la feuille)

Algorithme CART

Problèmes des arbres trop étoffés

- Complexité de l'arbre, trop de règles
- Trop spécifique aux données d'apprentissage
 - Règles non reproductibles (« surapprentissage »)
- Trop peu d'individus dans les feuilles (aucune signification réelle)
 - minimum conseillé : 20-30 individus

Solution → Élagage

Méthodes à base d'arbres de décision

CART (BFO'80 - Classification and regression trees, variables numériques, Gini, Elagage descendant)

C5 (Quinlan'93 - dernière version ID3 et C4.5, attributs d'arité quelconque, entropie et gain d'information)

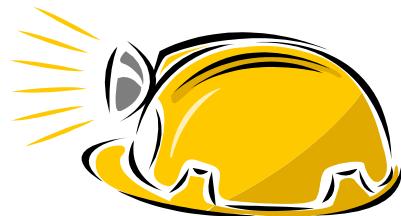
SLIQ (EDBT'96 — Mehta et al. IBM)

SPRINT (VLDB'96—J. Shafer et al. IBM)

PUBLIC (VLDB'98 — Rastogi & Shim)

RainForest (VLDB'98 — Gehrke, Ramakrishnan & Ganti)

CHAID (Chi-square Automation Interaction Detection – variables discrètes)



Arbres de décision - Avantages

Compréhensible pour tout utilisateur (lisibilité du résultat – règles - arbre)

Justification de la classification d'une instance (racine → feuille)

Tout type de données

Robuste au bruit et aux valeurs manquantes

Attributs apparaissent dans l'ordre de pertinence → tâche de pré-traitement (sélection d'attributs)

Classification rapide (parcours d'un chemin dans un arbre)

Outils disponibles dans la plupart des environnements de data mining

Arbres de décision - Inconvénients

Sensibles au nombre de classes : performances se dégradent

Evolutivité dans le temps : si les données évoluent dans le temps, il est nécessaire de relancer la phase d'apprentissage

Construction du modèle plus ou moins coûteuse

Classification bayésienne : Pourquoi ? (1)

Apprentissage probabiliste :

- calcul explicite de probabilités sur des hypothèses
- Approche pratique pour certains types de problèmes d'apprentissage

Incrémental :

- Chaque instance d'apprentissage peut de façon incrémentale augmenter/diminuer la probabilité qu'une hypothèse est correcte
- Des connaissances a priori peuvent être combinées avec les données observées.

Classification bayésienne : Pourquoi ? (2)

Prédiction Probabiliste :

- Prédit des hypothèses multiples, pondérées par leurs probabilités.

Référence en terme d'évaluation :

- Même si les méthodes bayésiennes sont coûteuses en temps d'exécution, elles peuvent fournir des solutions optimales à partir desquelles les autres méthodes peuvent être évaluées.

Classification bayésienne

Le problème de classification peut être formulé en utilisant les probabilités a-posteriori :

- $P(C|X)$ = probabilité que le tuple (instance)
- $X = \langle x_1, \dots, x_k \rangle$ est dans la classe C

Par exemple

- $P(\text{classe}=N | \text{outlook}=\text{sunny}, \text{windy}=\text{true}, \dots)$

Idée : affecter à une instance X la classe C telle que $P(C|X)$ est maximale

Estimation des probabilités a-posteriori

Théorème de Bayes :

- $P(C|X) = P(X|C) \cdot P(C) / P(X)$

$P(X)$ est une constante pour toutes les classes

$P(C)$ = fréquence relative des instances de la classe C

C tel que $P(C|X)$ est maximal =

C tel que $P(X|C) \cdot P(C)$ est maximal

Problème : calculer $P(X|C)$ est non faisable !

Classification bayésienne naïve

Hypothèse Naïve : indépendance des attributs

$$P(x_1, \dots, x_k | C) = P(x_1 | C) \cdot \dots \cdot P(x_k | C)$$

$P(x_i | C)$ est estimée comme la fréquence relative des instances possédant la valeur x_i (i -ème attribut) dans la classe C

Non coûteux à calculer dans les deux cas

Classification bayésienne – Exemple (1)

Estimation de $P(x_i|C)$

$$P(p) = 9/14$$

$$P(n) = 5/14$$

Outlook	
$P(\text{sunny} p) = 2/9$	$P(\text{sunny} n) = 3/5$
$P(\text{overcast} p) = 4/9$	$P(\text{overcast} n) = 0$
$P(\text{rain} p) = 3/9$	$P(\text{rain} n) = 2/5$
Temperature	
$P(\text{hot} p) = 2/9$	$P(\text{hot} n) = 2/5$
$P(\text{mild} p) = 4/9$	$P(\text{mild} n) = 2/5$
$P(\text{cool} p) = 3/9$	$P(\text{cool} n) = 1/5$

Humidity	
$P(\text{high} p) = 3/9$	$P(\text{high} n) = 4/5$
$P(\text{normal} p) = 6/9$	$P(\text{normal} n) = 1/5$
Windy	
$P(\text{true} p) = 3/9$	$P(\text{true} n) = 3/5$
$P(\text{false} p) = 6/9$	$P(\text{false} n) = 2/5$

Classification bayésienne – Exemple (1)

Classification de X :

- Une instance inconnue $X = \langle \text{rain}, \text{hot}, \text{high}, \text{false} \rangle$
- $P(X|p) \cdot P(p) =$
 $P(\text{rain}|p) \cdot P(\text{hot}|p) \cdot P(\text{high}|p) \cdot P(\text{false}|p) \cdot P(p) = 3/9 \cdot 2/9 \cdot 3/9 \cdot 6/9 \cdot 9/14 = 0.010582$
- $P(X|n) \cdot P(n) =$
 $P(\text{rain}|n) \cdot P(\text{hot}|n) \cdot P(\text{high}|n) \cdot P(\text{false}|n) \cdot P(n) = 2/5 \cdot 2/5 \cdot 4/5 \cdot 2/5 \cdot 5/14 = 0.018286$
- Instance X est classifiée dans la classe n (ne pas jouer)

Classification bayésienne – l'hypothèse d'indépendance

... fait que le calcul est possible

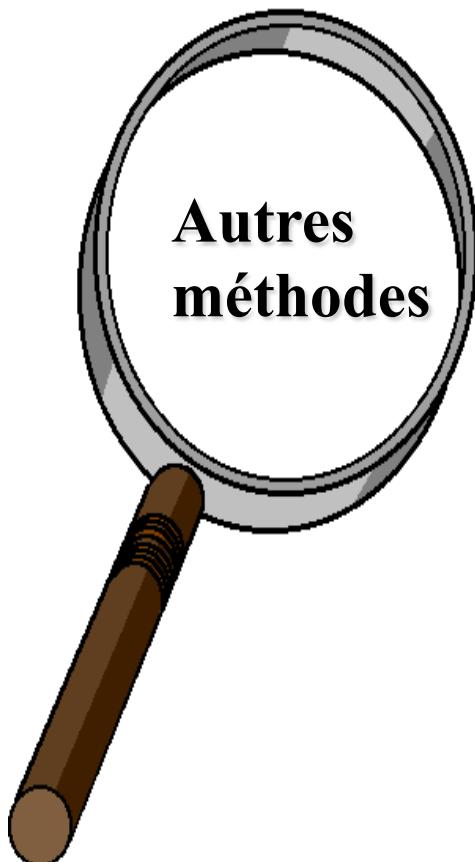
... trouve un modèle de classification optimal si hypothèse satisfaite

... mais est rarement satisfaite en pratique, étant donné que les attributs (variables) sont souvent corrélés.

Pour éliminer cette limitation :

- Réseaux bayésiens, qui combinent le raisonnement bayésien et la relation causale entre attributs
- Arbres de décision, qui traitent un attribut à la fois, considérant les attributs les plus importants en premier

Autres méthodes de classification



- Réseaux bayésiens
- Algorithmes génétiques
- Case-based reasoning
- Ensembles flous
- Rough set
- Analyse discriminante (Discriminant linéaire de Fisher, Algorithme Closest Class Mean - CCM-)
- Chaînes de Markov cachées

Classification - Résumé

La **classification** est un problème largement étudié

La **classification, avec ses nombreuses extensions, est probablement la technique la plus répandue**

- Modèles
 - Arbres de décision
 - Règles d'induction
 - Modèles de régression
 - Réseaux de neurones
- 
- Facile à comprendre
- Difficile à comprendre

Classification - Résumé

L'extensibilité reste une issue importante pour les applications

Directions de recherche : classification de données non relationnels, e.x., texte, spatiales et données multimédia

Classification - Références

- J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufman, 1993.
- J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81-106, 1986.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984.
- S. M. Weiss and C. A. Kulikowski. Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. Morgan Kaufman, 1991.
- D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning internal representation by error propagation. In D. E. Rumelhart and J. L. McClelland (eds.) Parallel Distributed Processing. The MIT Press, 1986

Outils pour le Data Mining

Comment Choisir un outil ?

Systèmes commerciaux de data mining possèdent peu de propriétés communes :

- Différentes méthodologies et fonctionnalités de data mining
- Différents types d'ensembles de données

Pour la sélection d'un outil, on a besoin d'une analyse multi-critère des systèmes existants



Critères de choix d'un logiciel

Variété des algorithmes de data mining, de statistique et de préparation des données

- + simple d'avoir tout dans un seul outil

Qualité des algorithmes implémentés

- documentation éditeur pas toujours accessible

Capacité à traiter de grands volumes de données

- peut être cruciale à partir de plusieurs centaines de milliers d'individus à traiter

Types de données gérés

- exemple : choix influencé si l'entreprise possède déjà un infocentre SAS...

Existence d'un langage de programmation évolué

- Convivialité du logiciel et facilités à produire des rapports

Prix !

Ce que l'on peut attendre d'un logiciel

Algorithmes de statistique et de data mining :

- classement (analyse discriminante linéaire, régression logistique binaire ou polytomique, modèle linéaire généralisé, régression logistique PLS, arbres de décision, réseaux de neurones, k-plus proches voisins...)
- prédiction (régression linéaire, modèle linéaire général, régression robuste, régression non-linéaire, régression PLS, arbres de décision, réseaux de neurones, + proches voisins...)
- Clustering (centres mobiles, nuées dynamiques, k-means, classification hiérarchique, méthode mixte, réseaux de Kohonen...)
- analyse des séries temporelles
- détection des associations

Ce que l'on peut attendre d'un logiciel

Fonctions de préparation des données

- manipulation de fichiers (fusion, agrégation, transposition...)
- visualisation des individus, coloriage selon critère
- détection, filtrage et winsorisation des extrêmes
- analyse et imputation des valeurs manquantes
- transformation de variables (recodage, standardisation, normalisation automatique, discréétisation...)
- création de nouvelles variables (fonctions logiques, chaînes, statistiques, mathématiques...)
- sélection des discréétisations, des interactions et des variables les plus explicatives

Ce que l'on peut attendre d'un logiciel

Présentation des résultats

- visualisation des résultats
- manipulation des tableaux
- bibliothèque de graphiques (2D, 3D, interactifs...)
- navigation dans les arbres de décision
- affichage des courbes de performances (ROC, lift, gain...)
- indice de Gini, aire sous la courbe ROC
- facilité d'incorporation de ces éléments dans un rapport
- Gestion des métadonnées
- variables définies identiquement pour tous les fichiers du projet (identifiant, cible, exclusions...)
- définition de groupes de variables

Ce que l'on peut attendre d'un logiciel

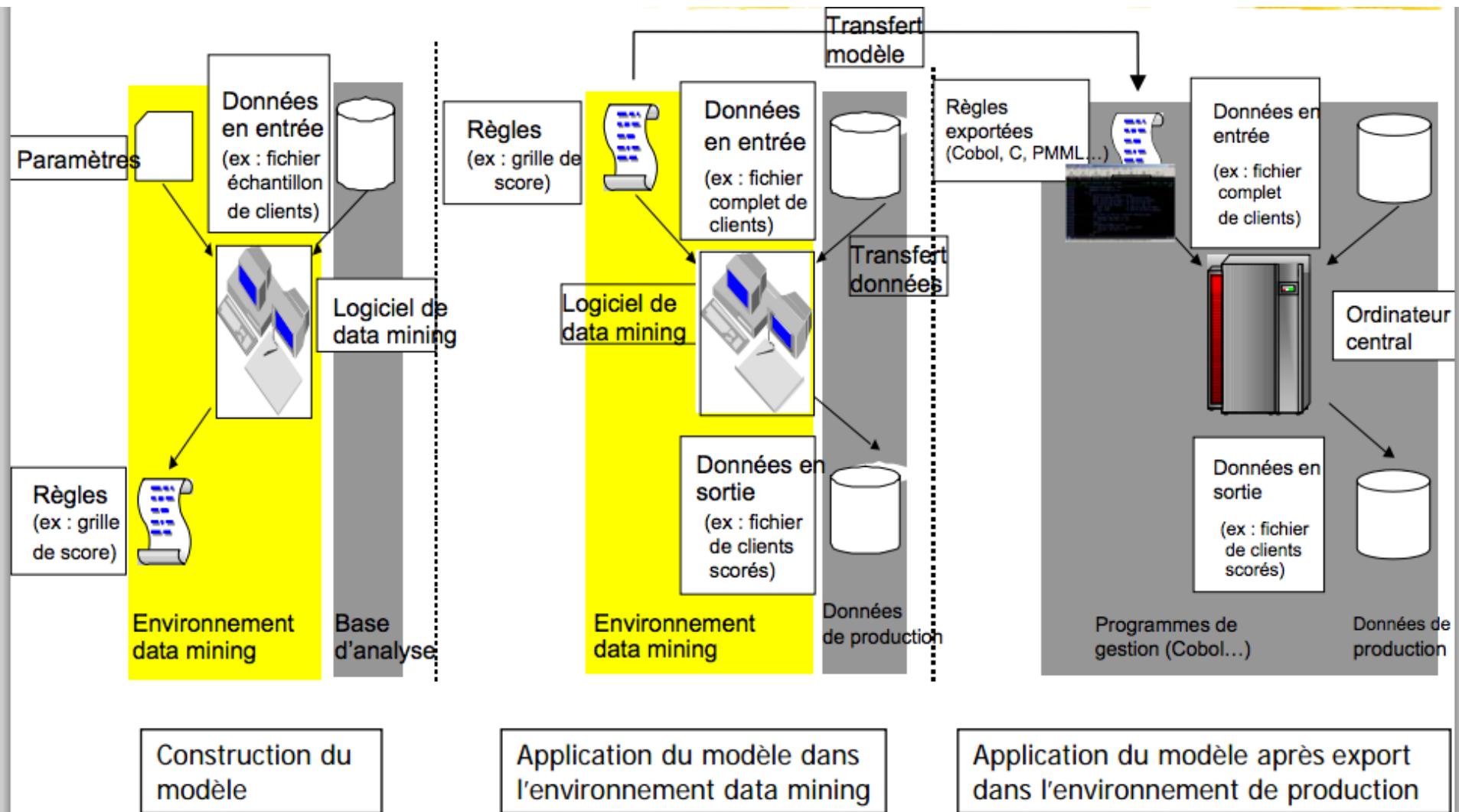
- Plates-formes supportées (Windows, Unix, Sun, IBM MVS...)
- Formats d'entrée/sortie des données gérés :
tables Oracle, Sybase, DB2, SAS, fichiers Excel, à plat...
- Enchaînements programmés de plusieurs algorithmes
- Volume de données pouvant être raisonnablement traité

Pour plus de puissance

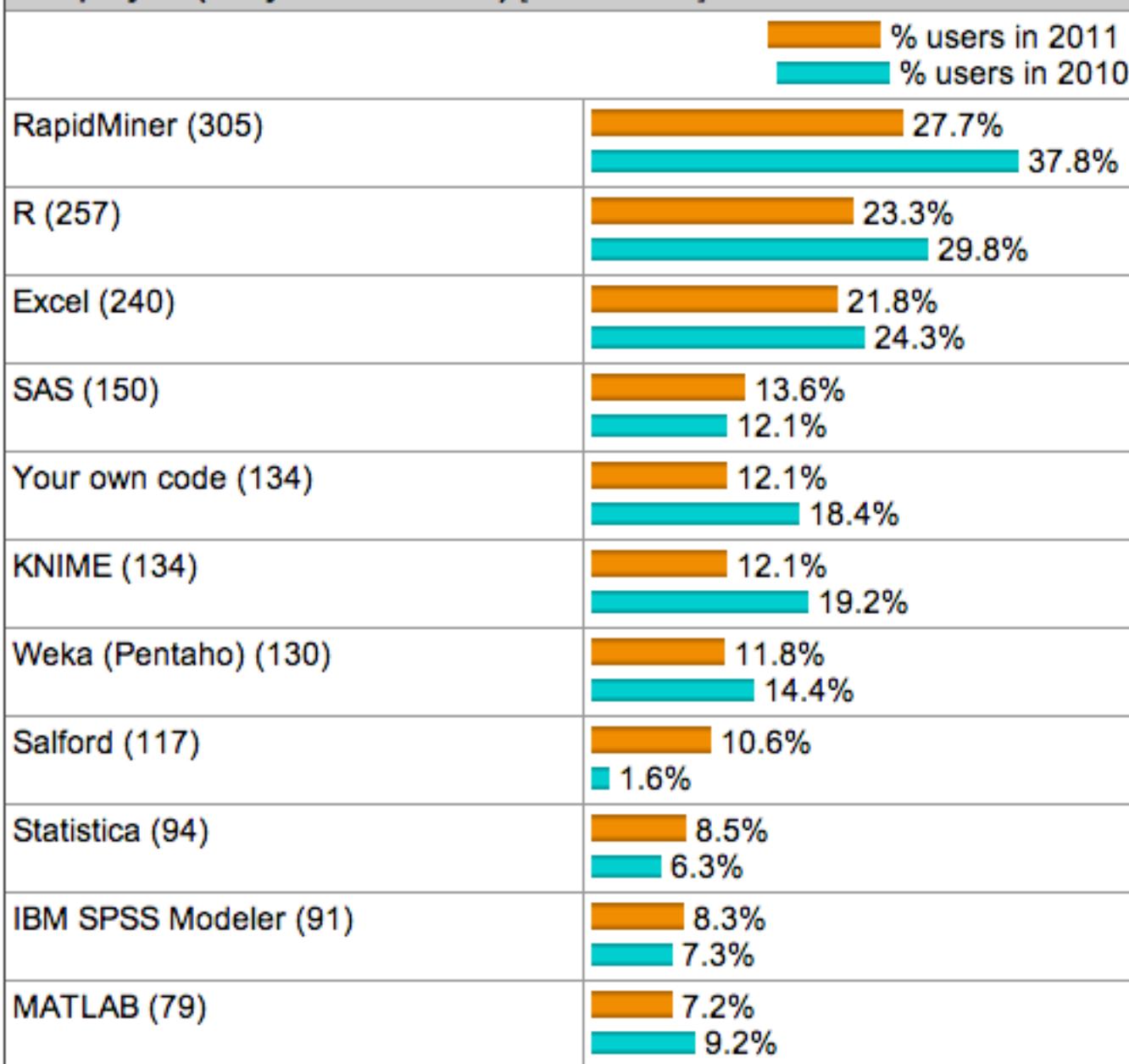
architecture client-serveur : calculs sur le serveur et visualisation des résultats sur le client
algorithmes parallélisés

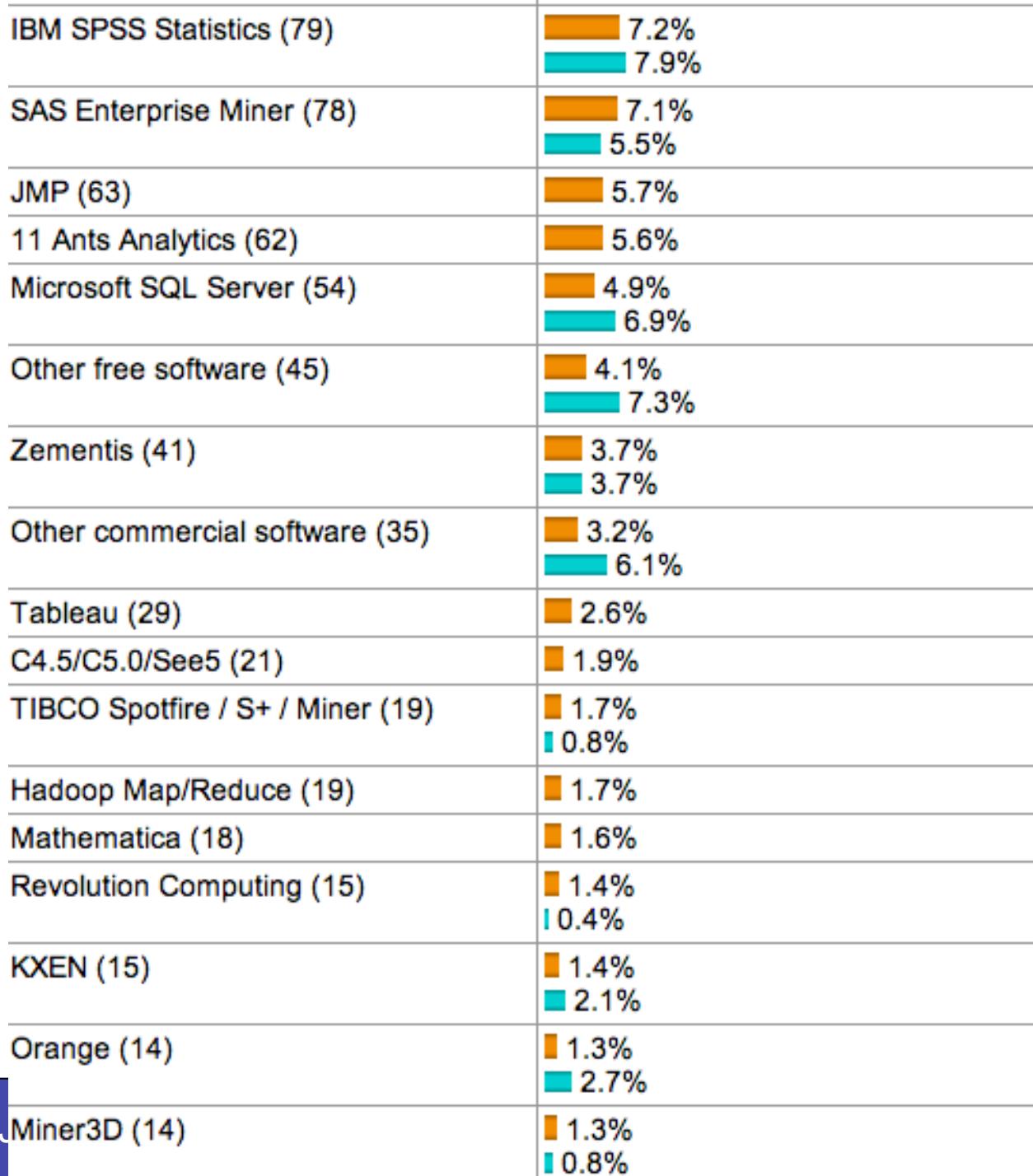
Exécution en mode interactif ou différé

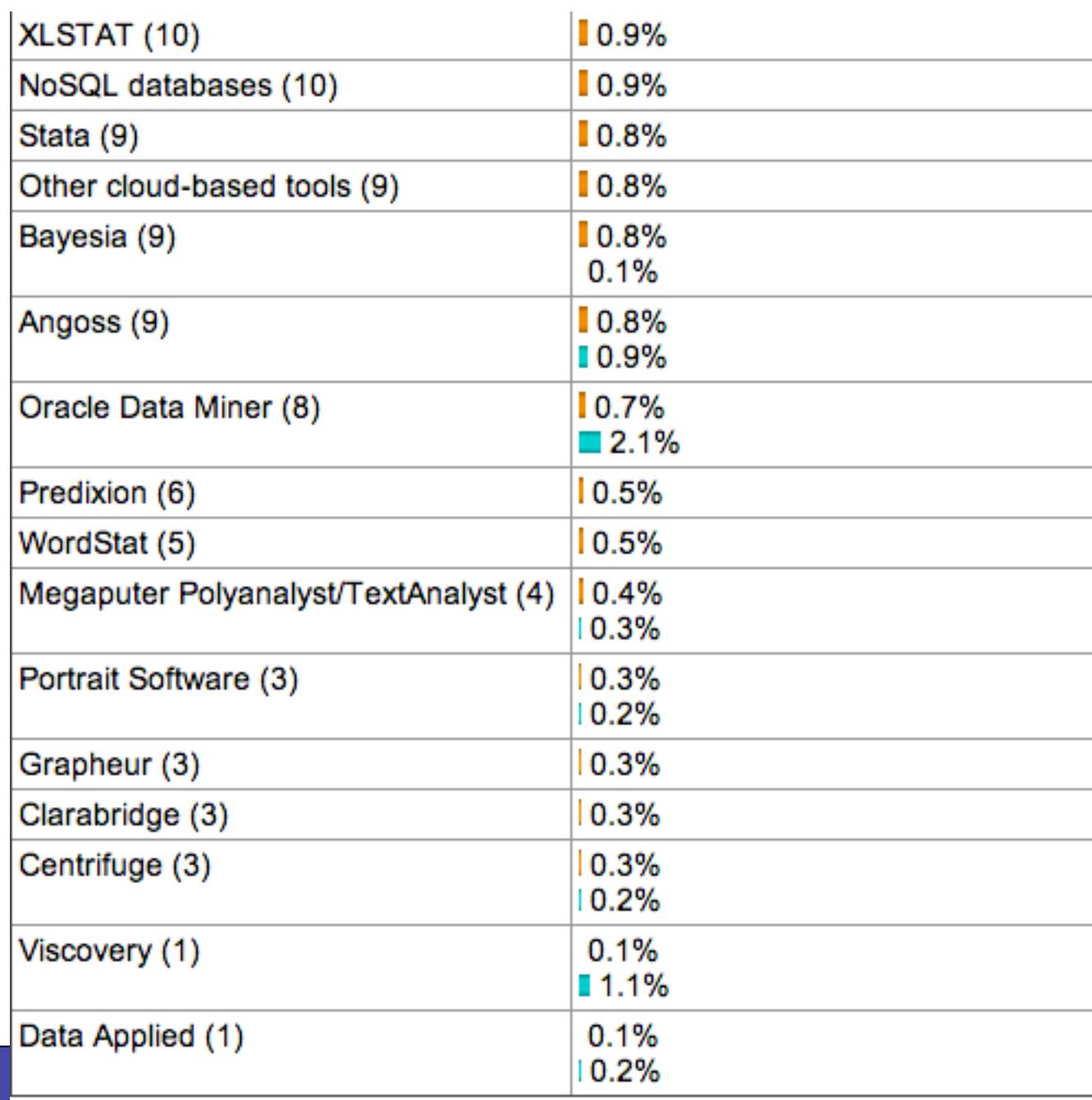
Portabilité des modèles construits (C, XML, Java, SQL...)



Which data mining/analytic tools you used in the past 12 months for a real project (not just evaluation) [1103 voters]







Outils non présents dans le sondage

IBM Cognos (used by 6% of data miners in 2010)

Minitab (9%)

FICO Fair Isaac (3%)

SAP Business Objects/NetWeaver (4%)

Teradata Warehouse Miner (2%)

Unica Predictive Insight (2%)

Logiciels multi-techniques	Insight - S-PLUS R (libre) Weka (gratuit) Tanagra (gratuit)	SAS Enterprise Miner SPSS Clementine Statsoft - Statistica Data Miner Insight - Insightful Miner SPAD KXEN
Logiciels mono-techniques	Salford Systems CART Isoft Alice Neuralware Predict DataLab (spécialiste du prétraitement des données)	SPSS Answer Tree
	Logiciels micros	Logiciels gros systèmes

Logiciels de data mining (poids légers : dizaines de milliers de lignes)

Produit	Spécialité (le cas échéant)	Éditeur
Stat Lab		SLP InfoWare (Gemplus)
StartMiner	Réseaux de neurones – Arbres de décision	Grimmersoft
Alice	Arbres de décision	Isoft
Predict	Réseaux de neurones	Neuralware
NeuroOne	Réseaux de neurones	Netral
Wizwhy	Associations	Wizsoft
WEKA		« open source » (logiciel gratuit)
R		« open source » (initialement développé à l'Université d'Auckland, Nelle-Zélande)
DATALAB	Prétraitement des données	Complex Systems

Logiciels de data mining (poids moyens : centaines de milliers de lignes)

Produit	Spécialité (le cas échéant)	Éditeur
4Thought	Réseaux de neurones	Cognos
KnowledgeSEEKER	Arbres de décision	Angoss
KnowledgeSTUDIO		Angoss
C5.0 (Unix) See5 (Windows)	Arbres de décision	RuleQuest Research
Data Mining Suite		Salford Systems
CART	Arbres de décision	Salford Systems
Polyanalyst		Megaputer
S-PLUS		Insightful
TANAGRA		Laboratoire ERIC de l'Université de Lyon

Logiciels de data mining (poids lourds : millions de lignes)

Produit	Spécialité (le cas échéant)	Éditeur
KXEN	Théorie de l'apprentissage de Vapnik	KXEN
Intelligent Miner	Classification relationnelle – Réseaux de neurones	IBM
Microsoft Analysis Services	Arbres de décision – clustering	Microsoft
Oracle Data Mining		Oracle
SPAD		SPAD
SPSS		SPSS
Clementine		SPSS
Statistica Data Miner		Statsoft
Insightful Miner		Insightful
SAS/STAT		SAS
Entreprise Miner		SAS

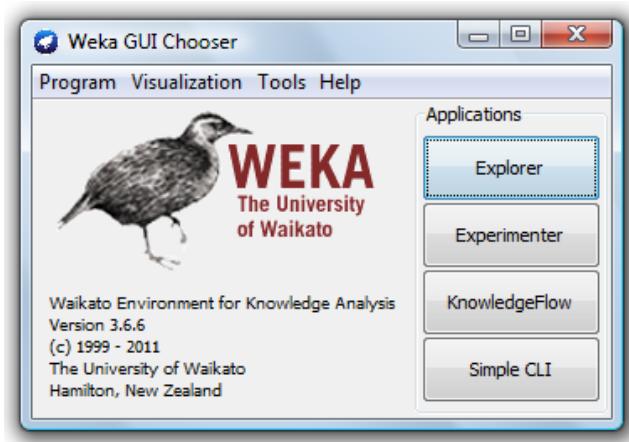
Exemple d'outils (1)

- Intelligent Miner d'IBM
 - Intelligent Miner for Data (IMA)
 - Intelligent Miner for Text (IMT)
 - Tâches : groupage de données, classification, recherche d'associations, etc.
- Entreprise Miner de SAS
 - SAS : longue expérience en statistiques
 - Outil «complet» pour le DM

Weka

Logiciel d'apprentissage machine :

- Traitement de données
- Forage de données
- Comparaison d'algorithmes
- Etc.



Site web:

<http://www.cs.waikato.ac.nz/ml/weka/index.html>

SAS Enterprise Miner

Société : SAS Institute Inc.

Création : Mai 1998

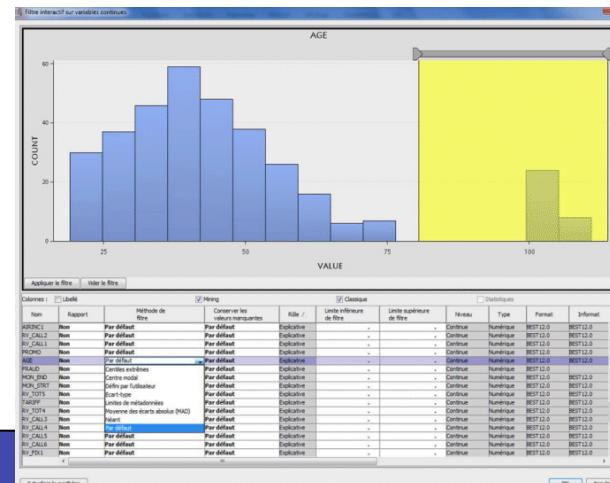
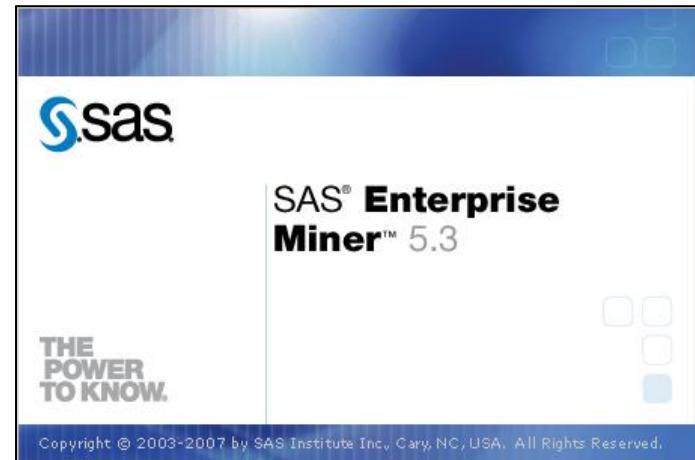
Plate-formes : Windows , Unix, Linux

Utilisation

- Réduction des coûts
- Maîtrise des risques
- Fidélisation
- Prospection

Outils de data warehouse

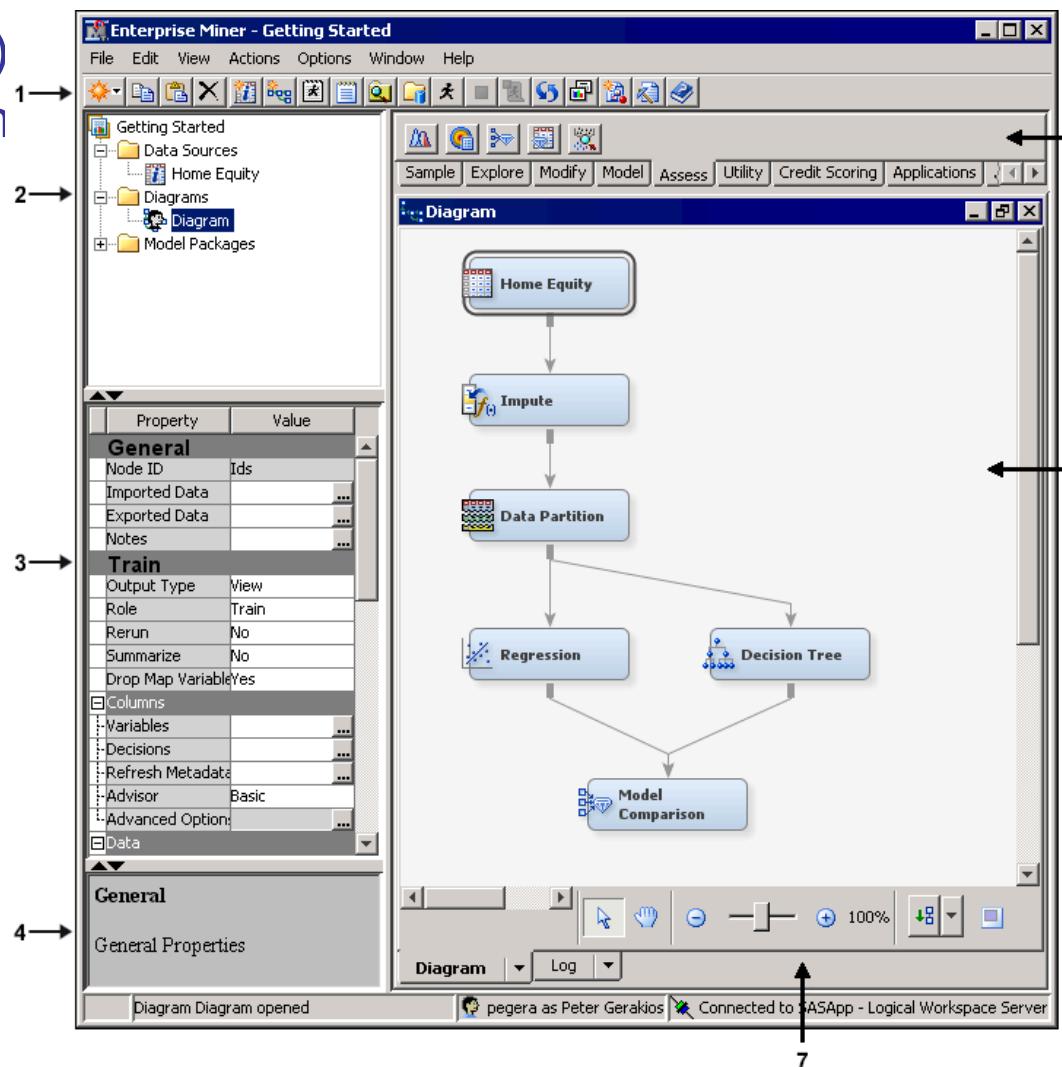
[http://www.sas.com/offices/europe/france/software/technologies/
datamining/datamining.html](http://www.sas.com/offices/europe/france/software/technologies/datamining/datamining.html)



SAS Enterprise Miner

Interface graphique (icônes)

Construction d'un diagramm



SAS Enterprise Miner

Deux types d'utilisateurs

- Spécialistes en statistiques
- Spécialistes métiers (chef de projet, études...)

Techniques implémentées

- Arbres de décision
- Régression
- Réseaux de neurones



Preparing the Data



Parsing



Quantifying



Transforming



Reducing/Combining



Analyzing

Alice

Société : ISoft

Création : 1988

Plate-formes :



Utilisation

- Marketing : études de marché, segmentation ...
- Banque, Assurance : scoring, analyse de risques, détection de fraudes
- Industrie : contrôle qualité, diagnostic, segmentation, classification, construction de modèles, prédition et simulation

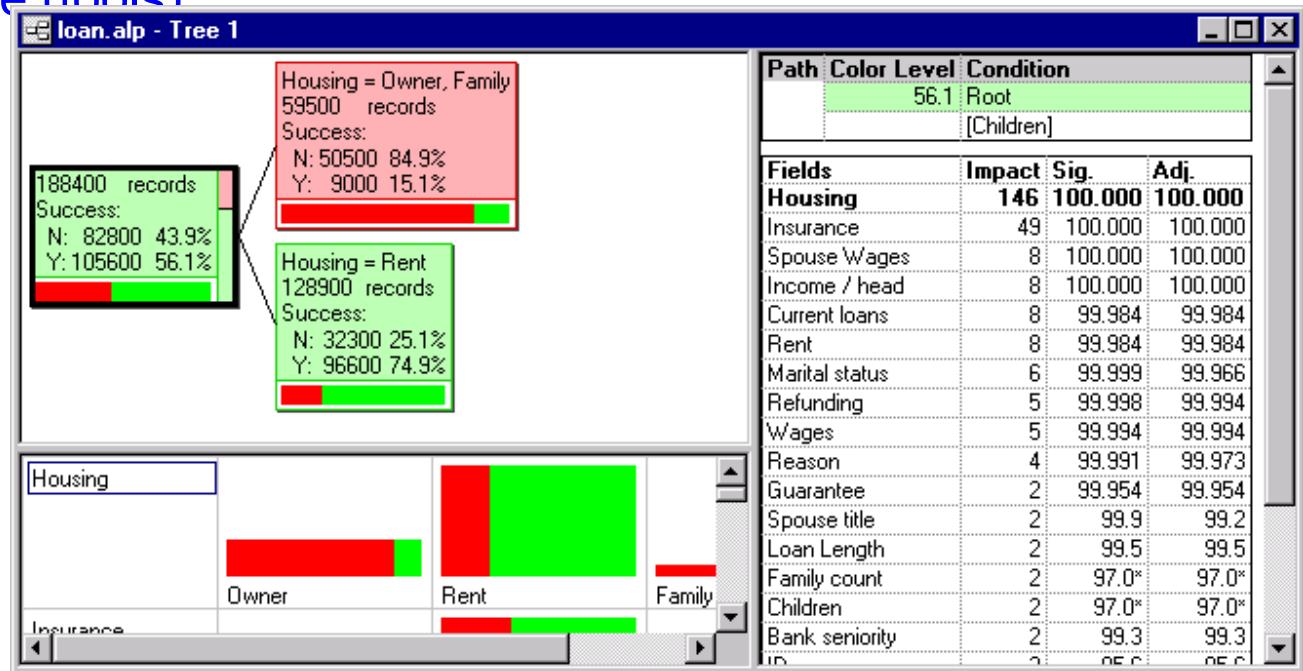
SITE WEB http://www.isoft.fr/html/prod_alice.htm

FONCTIONNALITES

- ALICE / Arbre de décision interactif
- ALICE / CLUSTERING: regroupe les individus semblables en classes

Alice

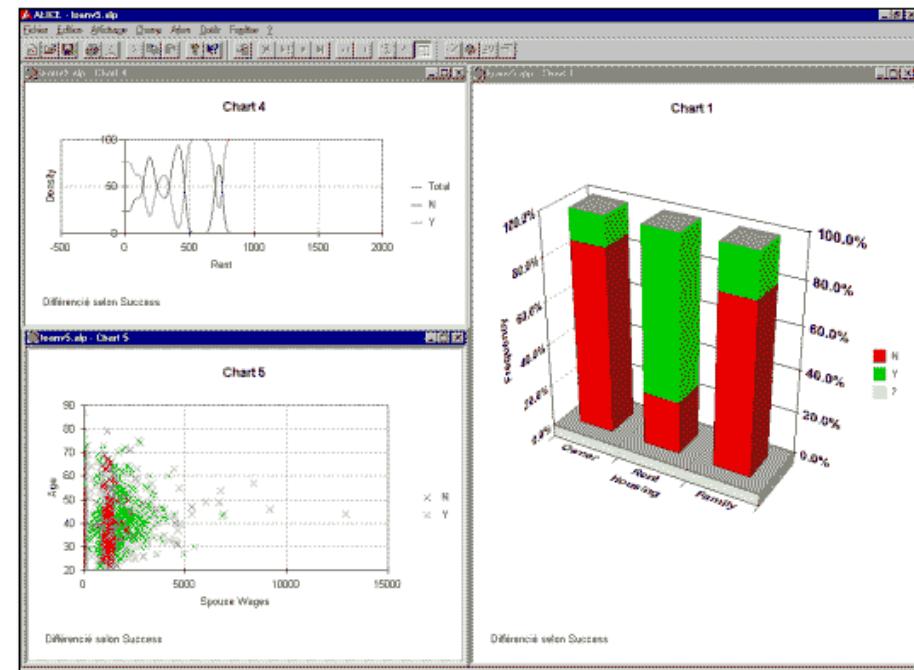
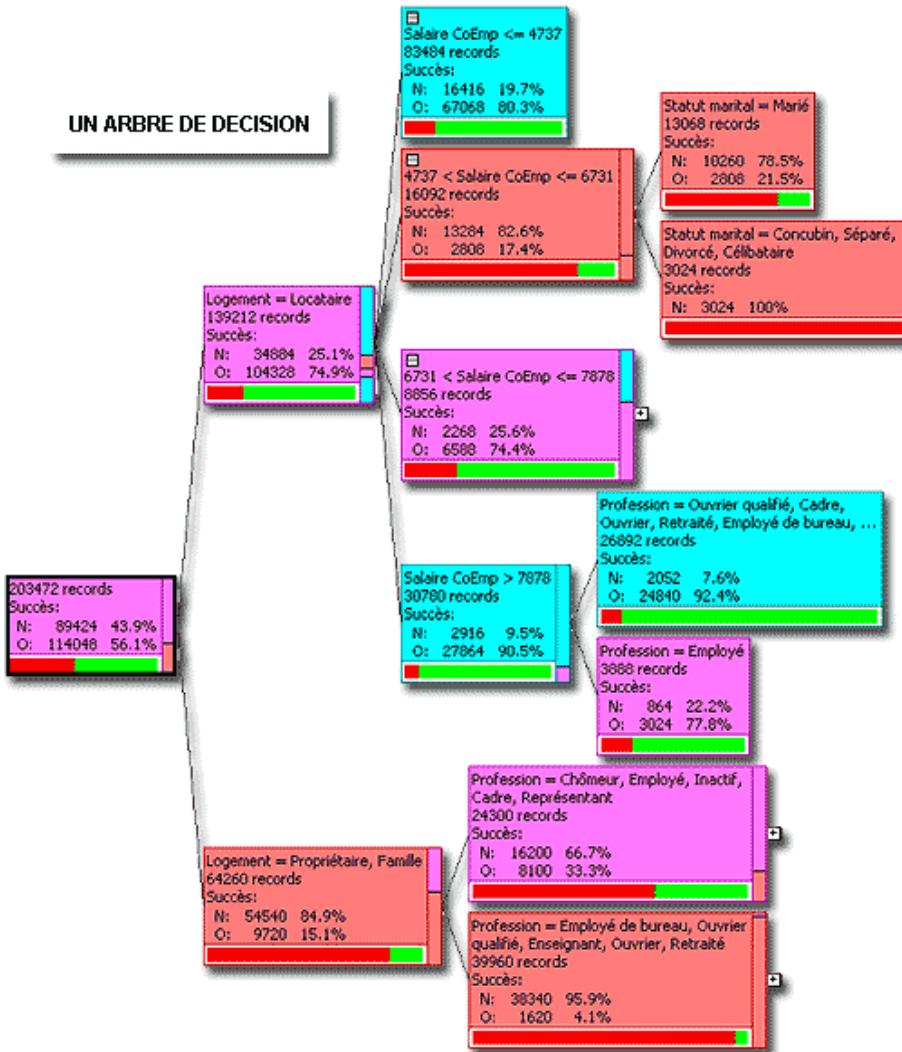
Interface graphique (tools)



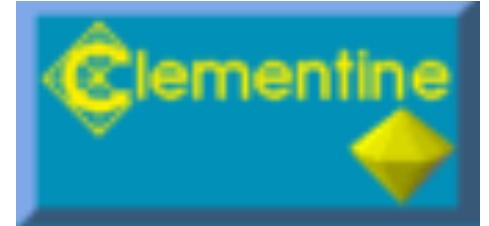
Type d'utilisateur : responsables opérationnels

Alice

UN ARBRE DE DECISION



Clémentine



Société : ISL (Integral Solutions Limited) racheté par SPSS et IBM

Création : 1994

Plate-formes : Windows NT, Unix

Utilisation

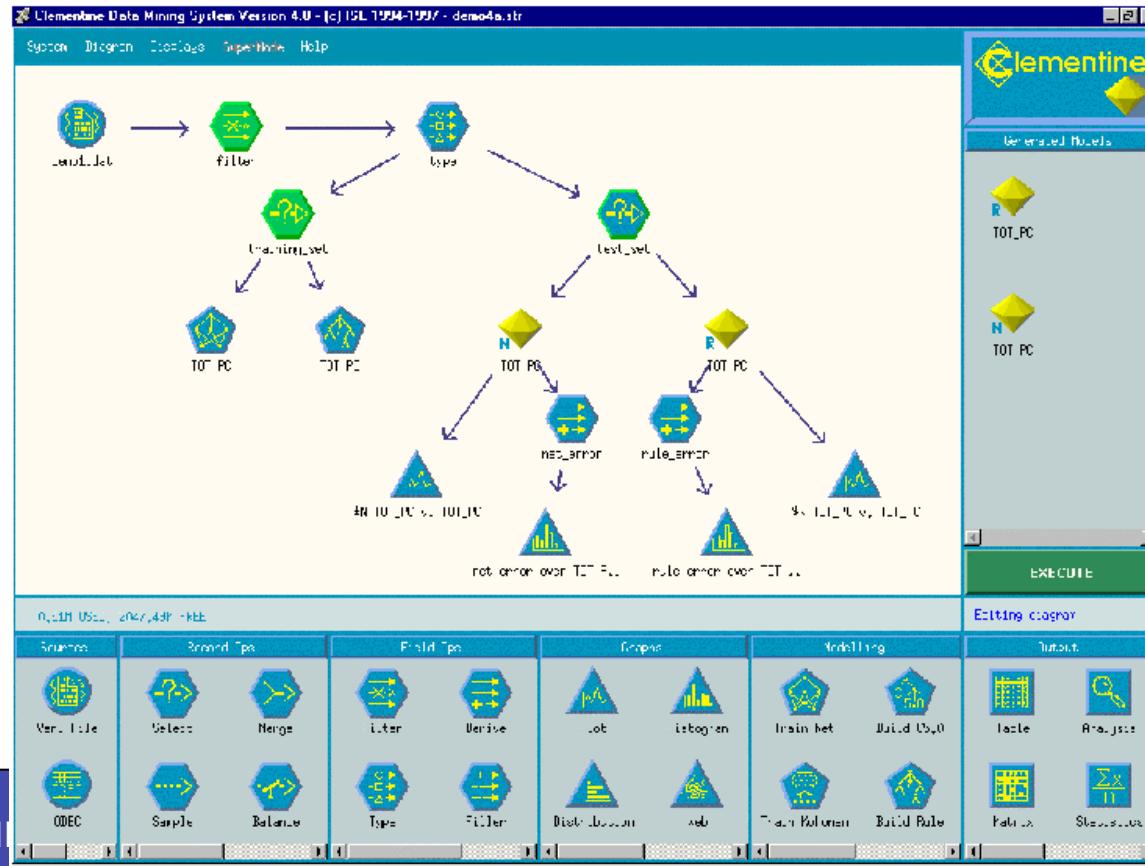
- Prévision de parts de marché
- Détection de fraudes
- Segmentation de marché
- Implantation de points de vente ...

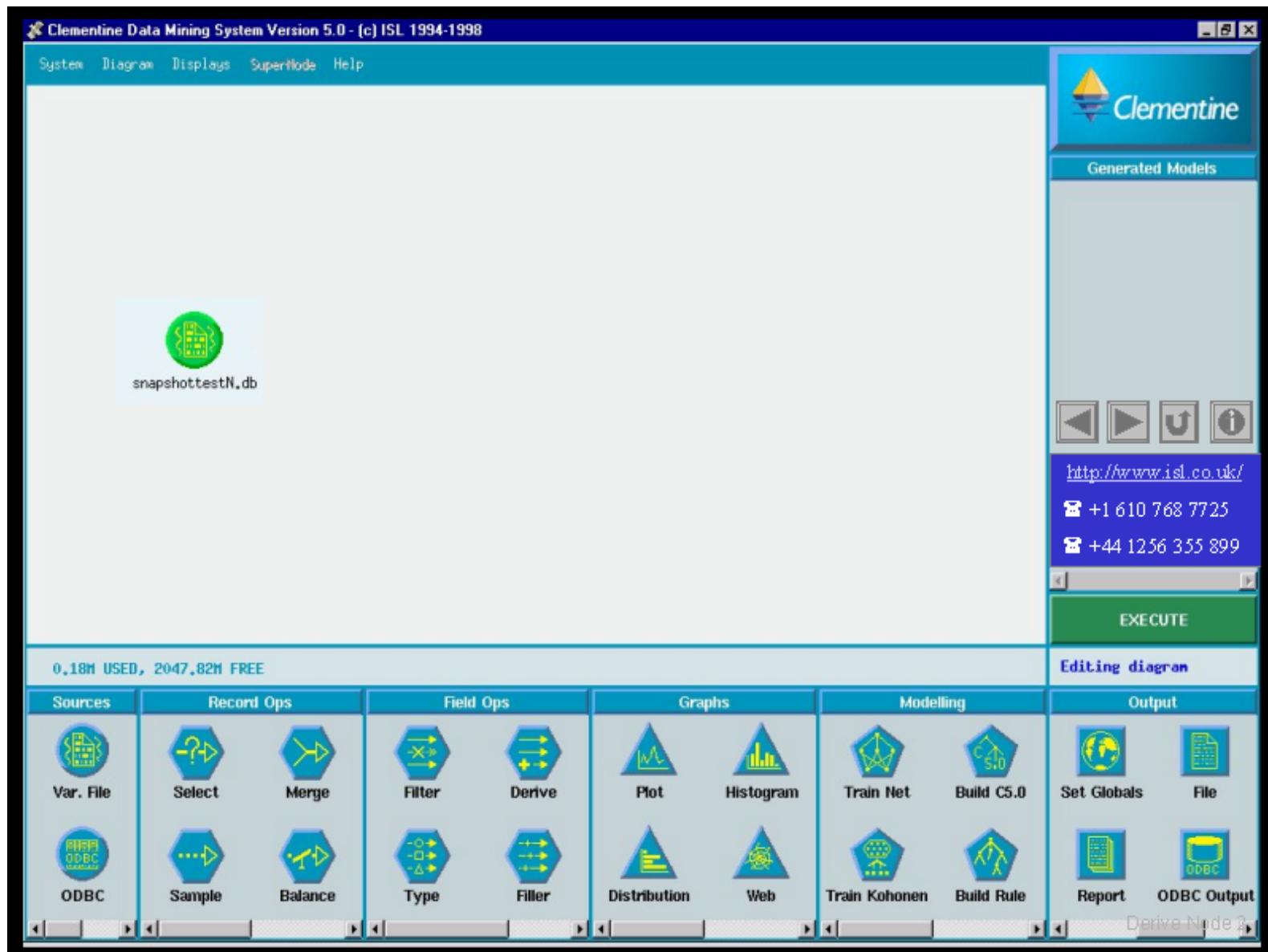
Techniques

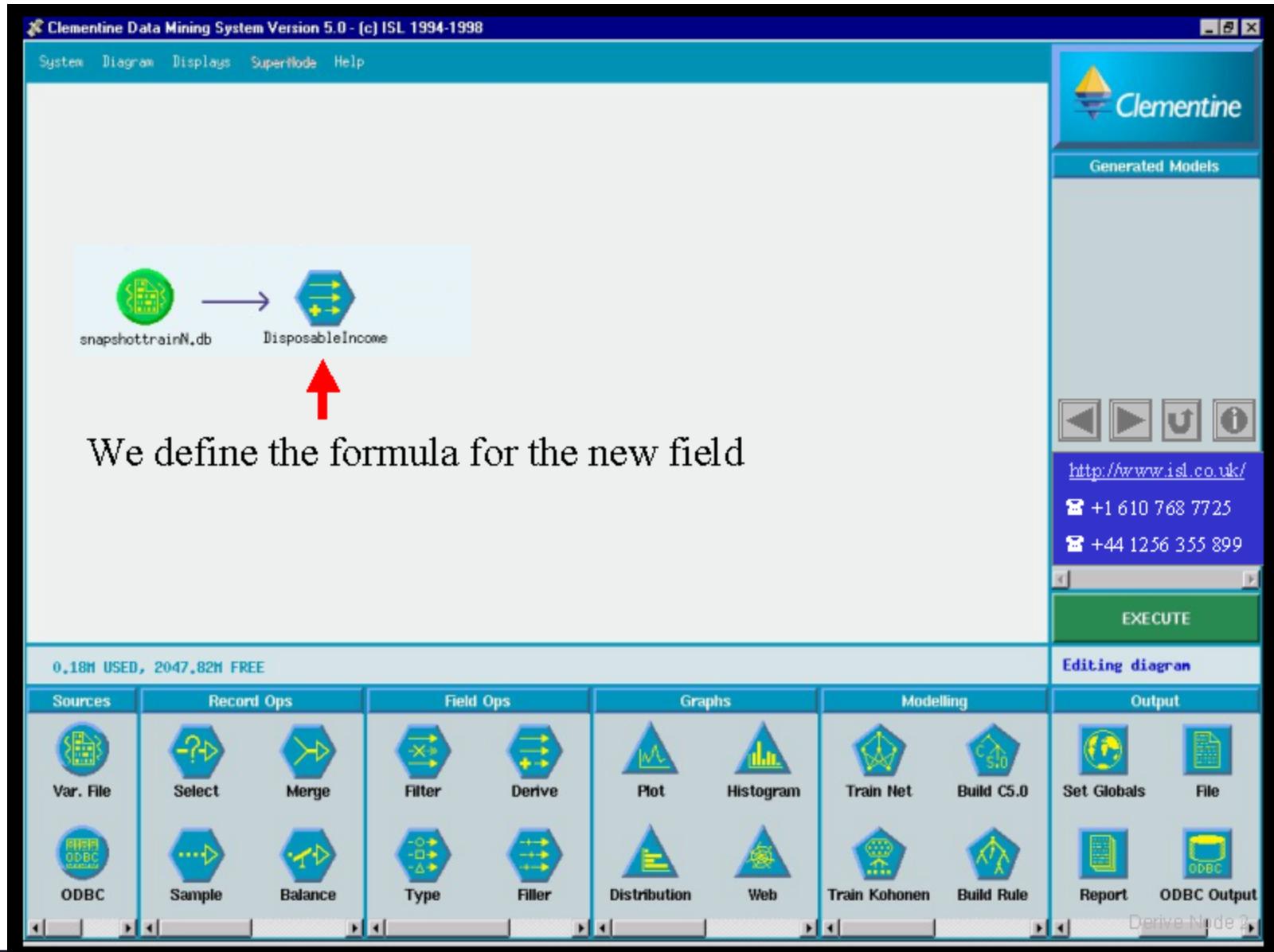
- Rule Induction
- Graph
- Clustering
- Association Rules
- Linear Regression
- - Neural Networks

Clémentine

Interface simple, puissante et complète
interface conviviale



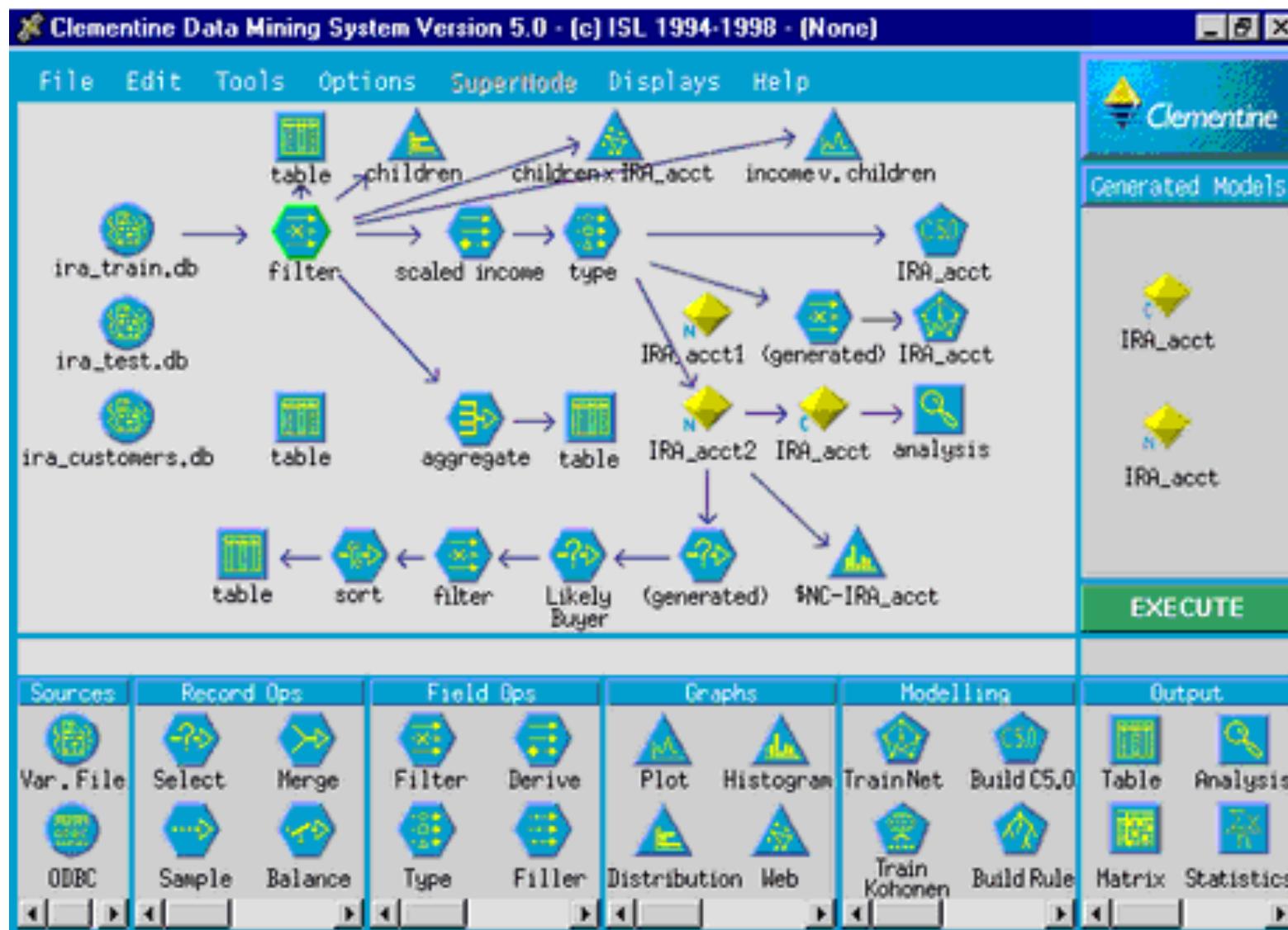




We define the formula for the new field

The screenshot shows the Clementine Data Mining System interface. A central dialog box titled "DisposableIncome" is open, showing the configuration for a new field. The "Field" section has "New field name: DisposableIncome" and "Type: Conditional". The "Field Derivation" section contains the formula: "If: children == 0", "Then: income", and "Else: income / children". Below the dialog are two nodes: "snapshottrainN.db" (represented by a green hexagon) and "DisposableIncome" (represented by a blue hexagon with a double-headed arrow). A red arrow points from the text "We define the formula for the new field" up towards the dialog. The main workspace below shows various data mining operations like Sources, Record Ops, Field Ops, Graphs, Modelling, and Output, along with a toolbar and status bar.

We define the formula for the new field



Forecast Pro

Société : Business Forecast Systems

Création : 1997

Plate-formes : Windows 95, NT

Utilisation

- Tous domaines activités et secteurs
- Notamment la prévision (5 types différents)

Site web : <http://www.forecastpro.com/>

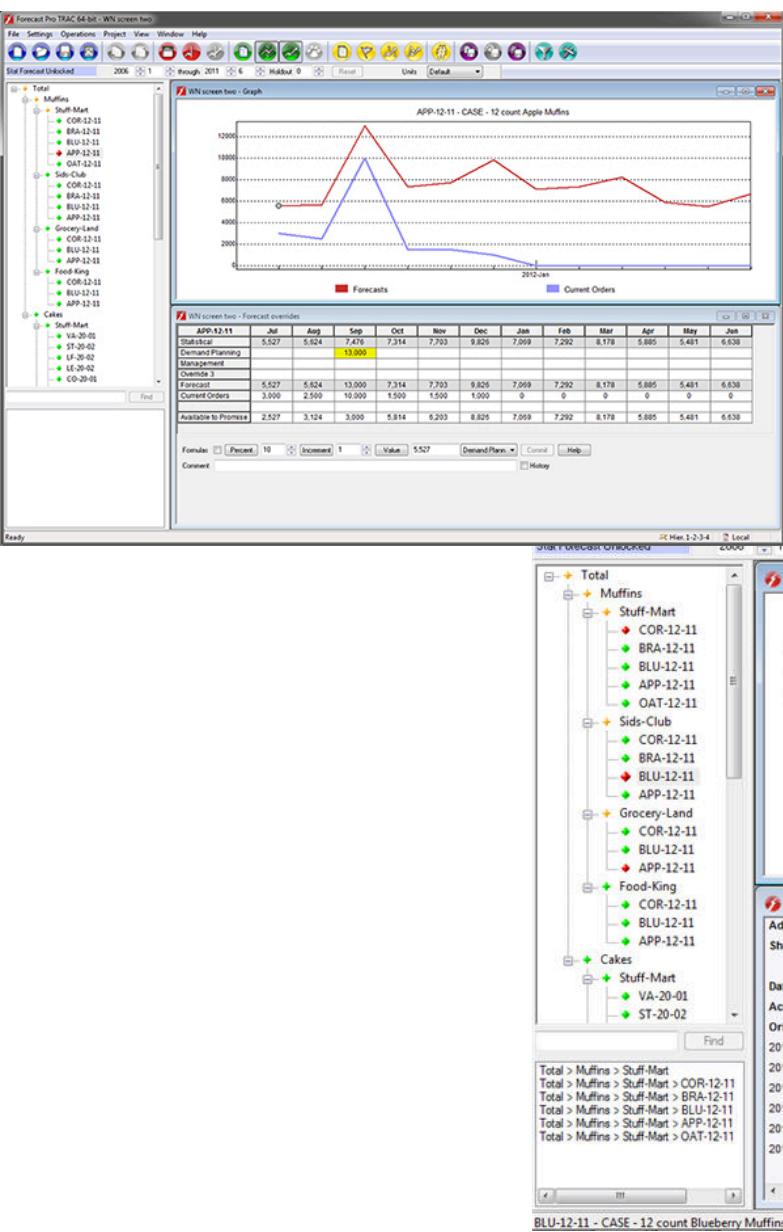
Clémentine

Types d'utilisateurs : PME/PMI, administrations, consultants, universitaires, chefs de projets,...

Facilité d'utilisation (connaissances en statistiques non requises)

Vaste palette de choix graphiques

- Valeurs observées, prévisions, valeurs calculées sur l'historique, intervalles de confiance, diagnostics (erreurs)





Intelligent Miner

Société : IBM

Création : 1998

Plate-formes : AIX, OS/390, OS/400, Solaris, Windows 2000 & NT

Utilisation

- Domaines où l'aide à la décision est très importante (exemple : domaine médical)
- Analyse de textes

Fortement couplé avec DB2 (BD relationnel)

Intelligent Miner

Deux versions

- Intelligent Miner for Data (IMD)
- Intelligent Miner for Text (IMT)

Types d'utilisateurs : spécialistes ou professionnels expérimentés

Parallel Intelligent Miner

Intelligent Miner

L'IMD

- Sélection et codage des données à explorer
- Détermination des valeurs manquantes
- Agrégation de valeurs
- Diverses techniques pour la fouille de données
 - Règles d'association (Apriori), classification (Arbres de décision, réseaux de neurones), clustering, détection de déviation (analyse statistique & visualisation)
- Visualisation des résultats
- Algorithmes extensibles (scalability)

Intelligent Miner

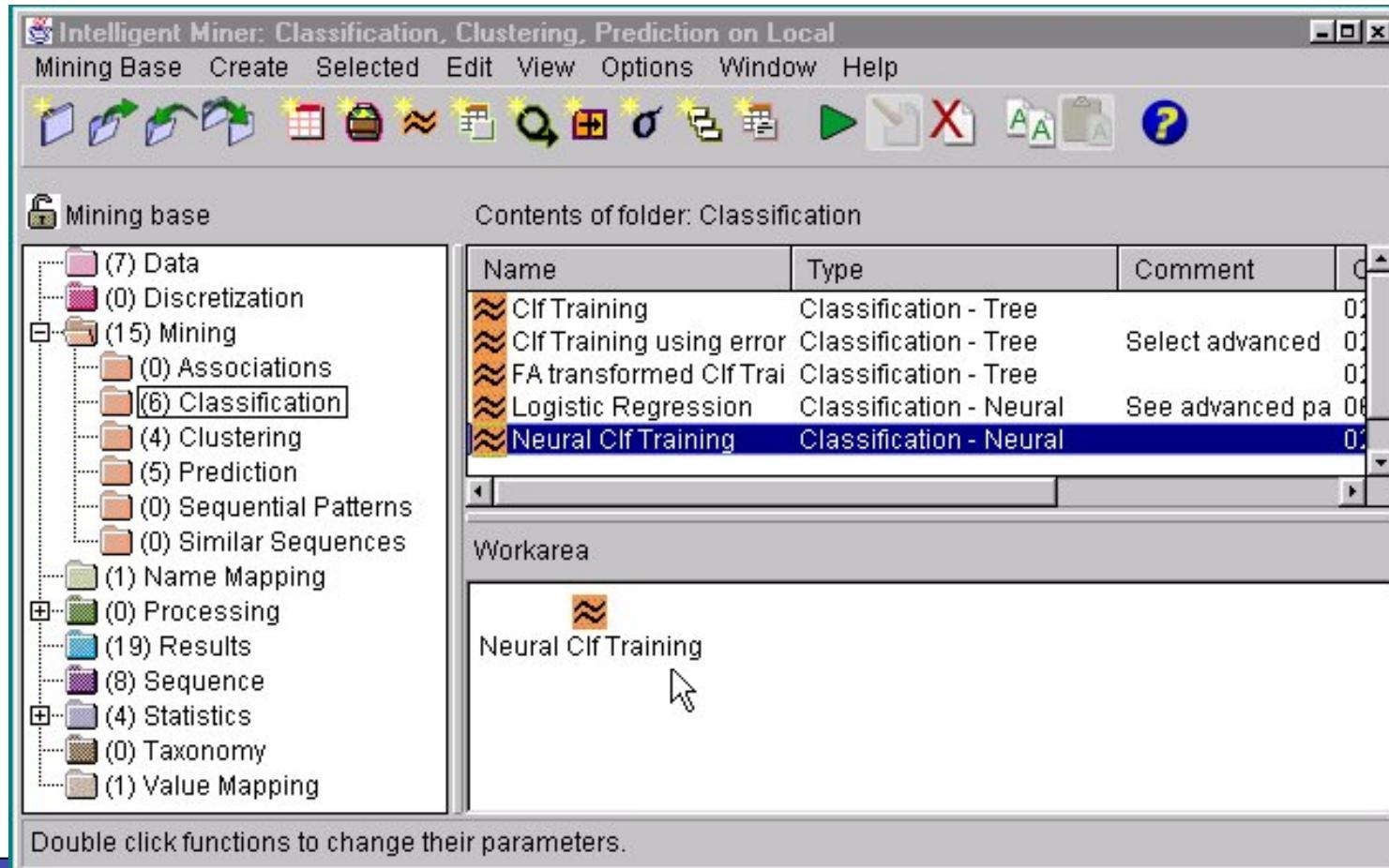
IMT = analyse de textes libres

Trois composants

- Moteur de recherche textuel avancé (TextMiner)
- Outil d'accès au Web (moteur de recherche NetQuestion et un méta-moteur)
- Outil d'analyse de textes (Text Analysis)

L'objectif général est de faciliter la compréhension des textes

Intelligent Miner



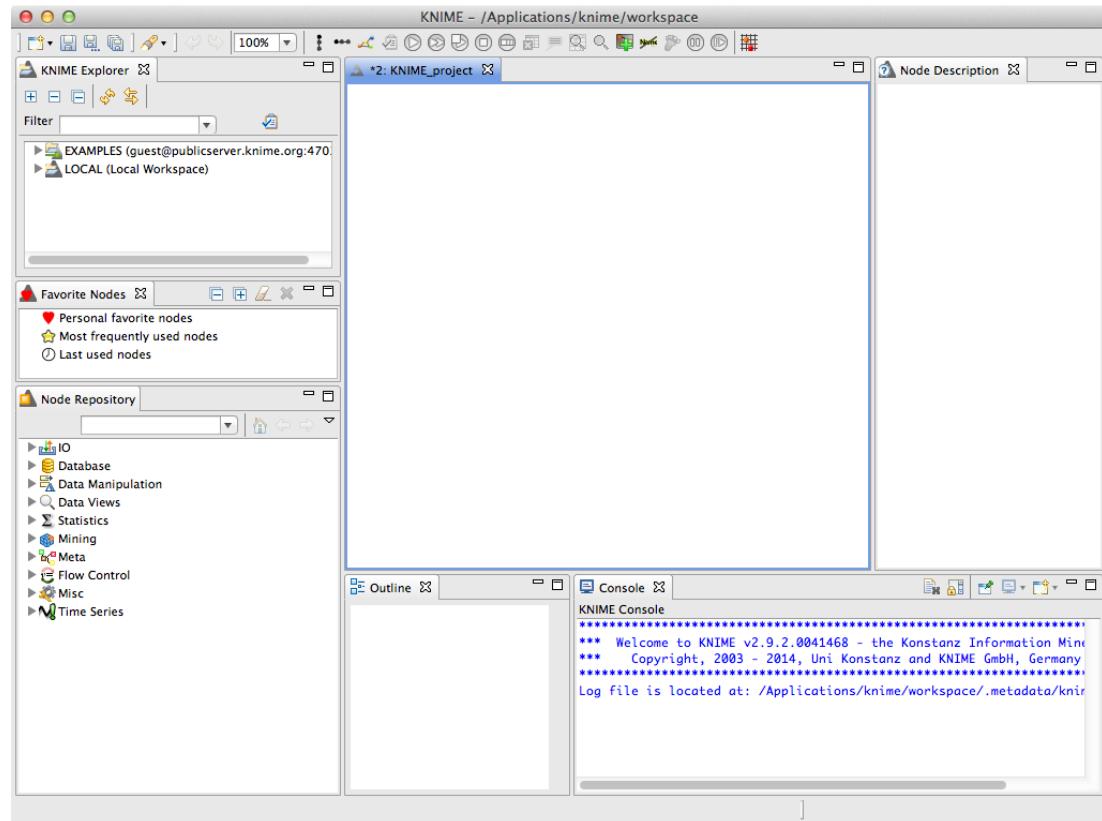
Outils Open Source

Quelques outils classiques

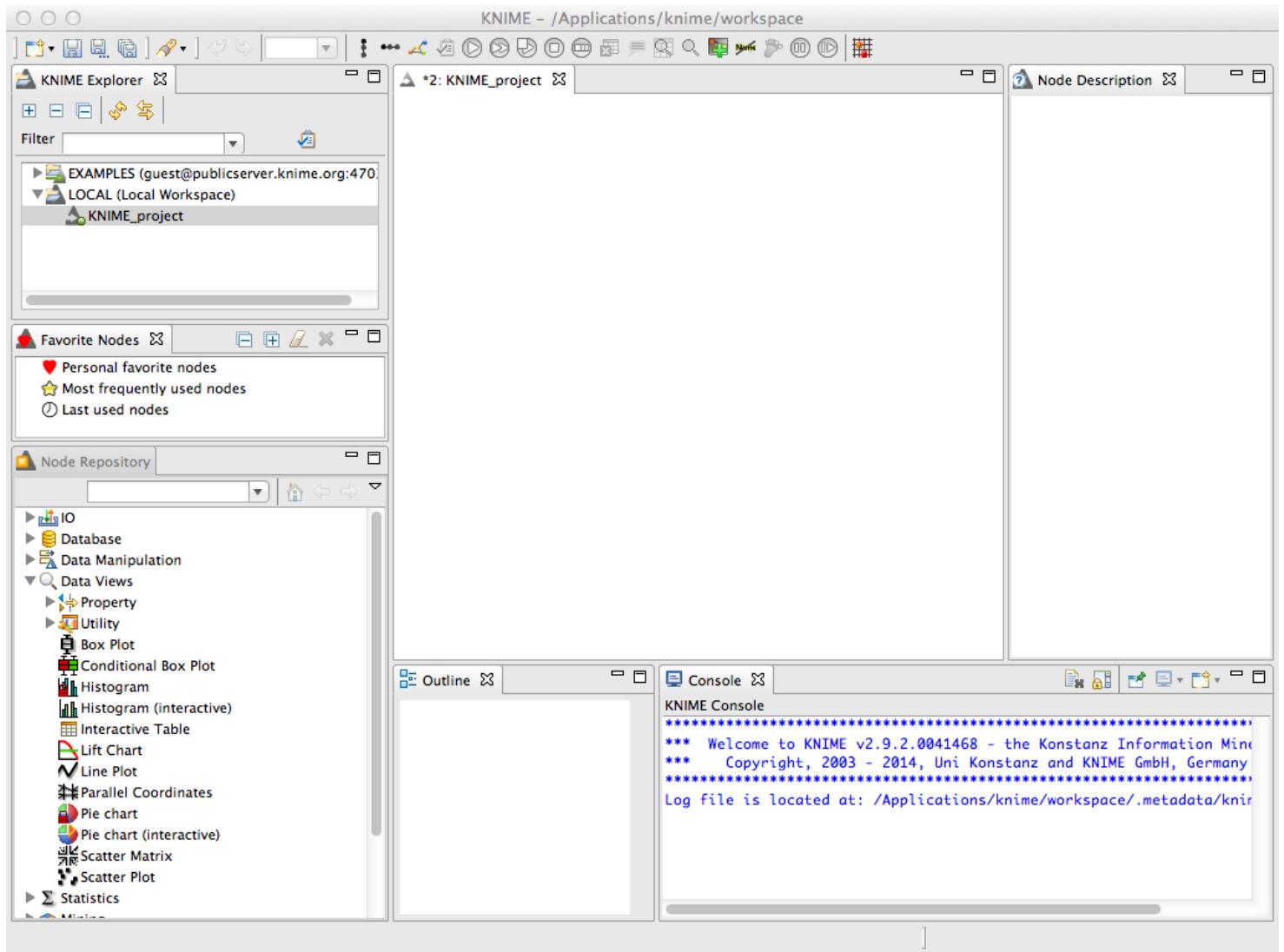
- Weka www.cs.waikato.ac.nz/ml/weka/
- Rapid Miner www.rapid-i.com
- Knime www.knime.org
- Orange orange.biolab.si

Knime

- Support Java

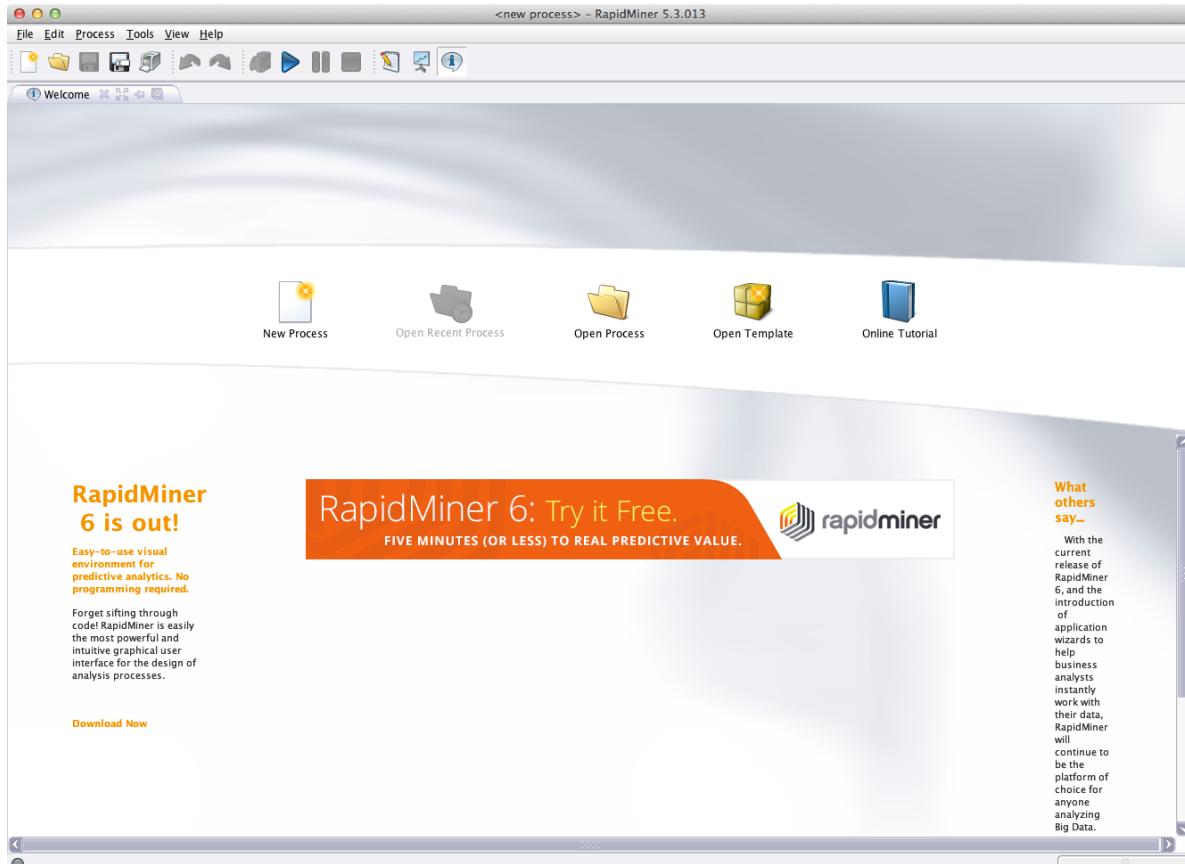


Knime



RapidMiner

Java



Autres techniques

Web mining (contenu, usage, ...)

Visual data mining (images)

Audio data mining (son, musique)

Data mining et requêtes d'interrogation “intelligentes”

Visualisation de données

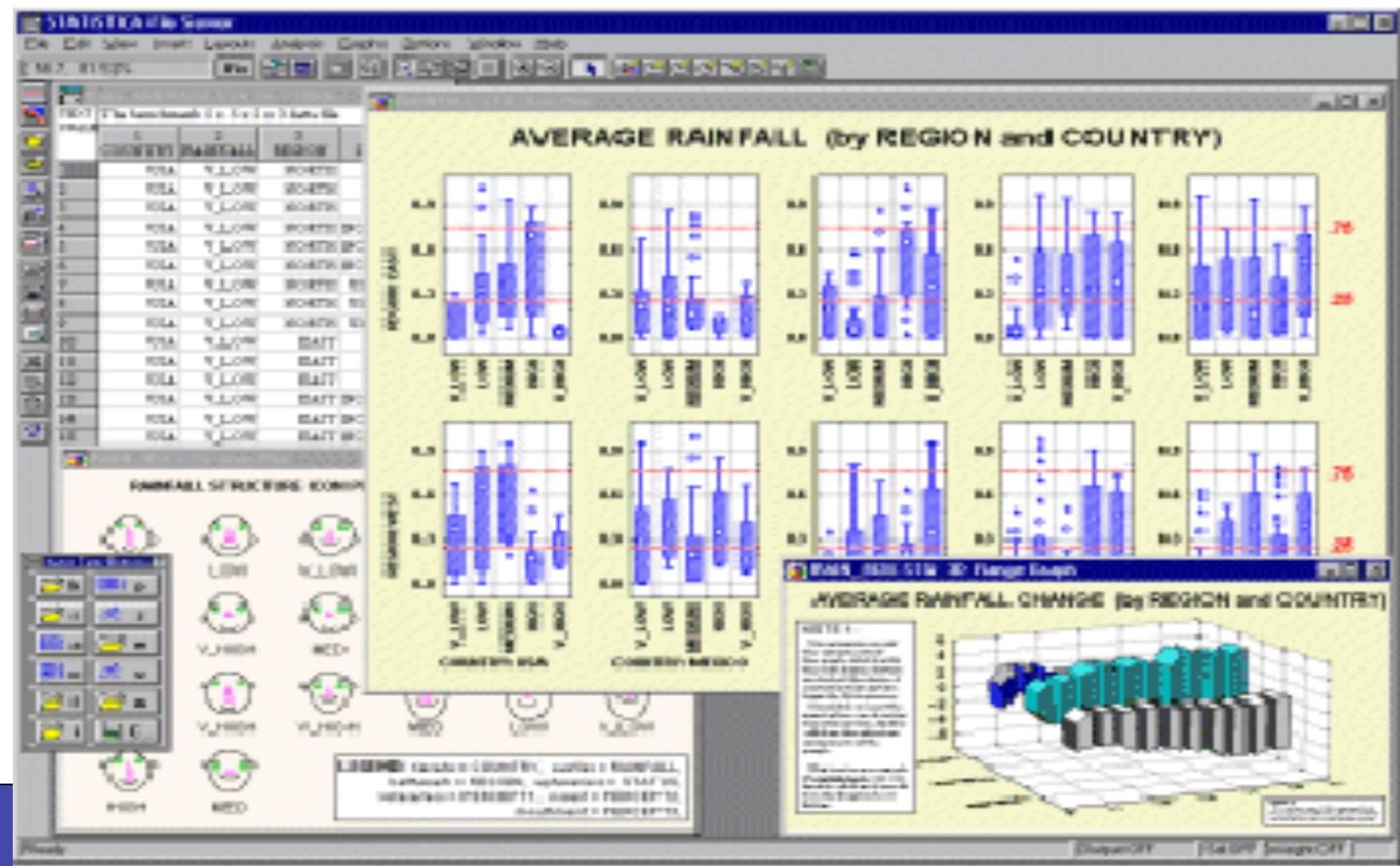
Données dans un base de données ou un entrepôt de données peuvent être visualisées :

À différents niveaux de granularité ou d'abstraction

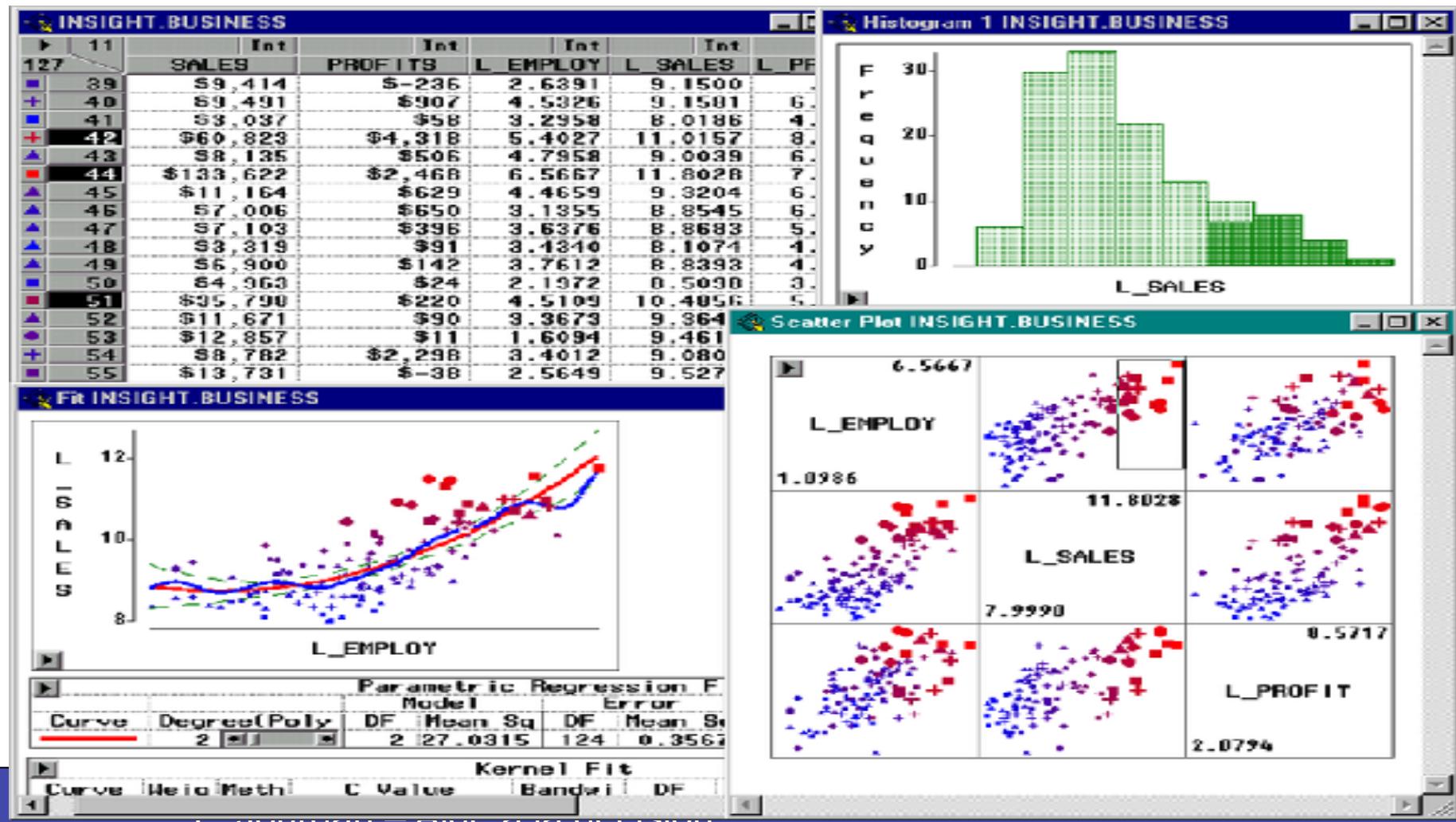
A l'aide de différentes combinaisons d'attributs ou dimensions

Résultats des outils de Data Mining peuvent être présentées sous diverses formes visuelles

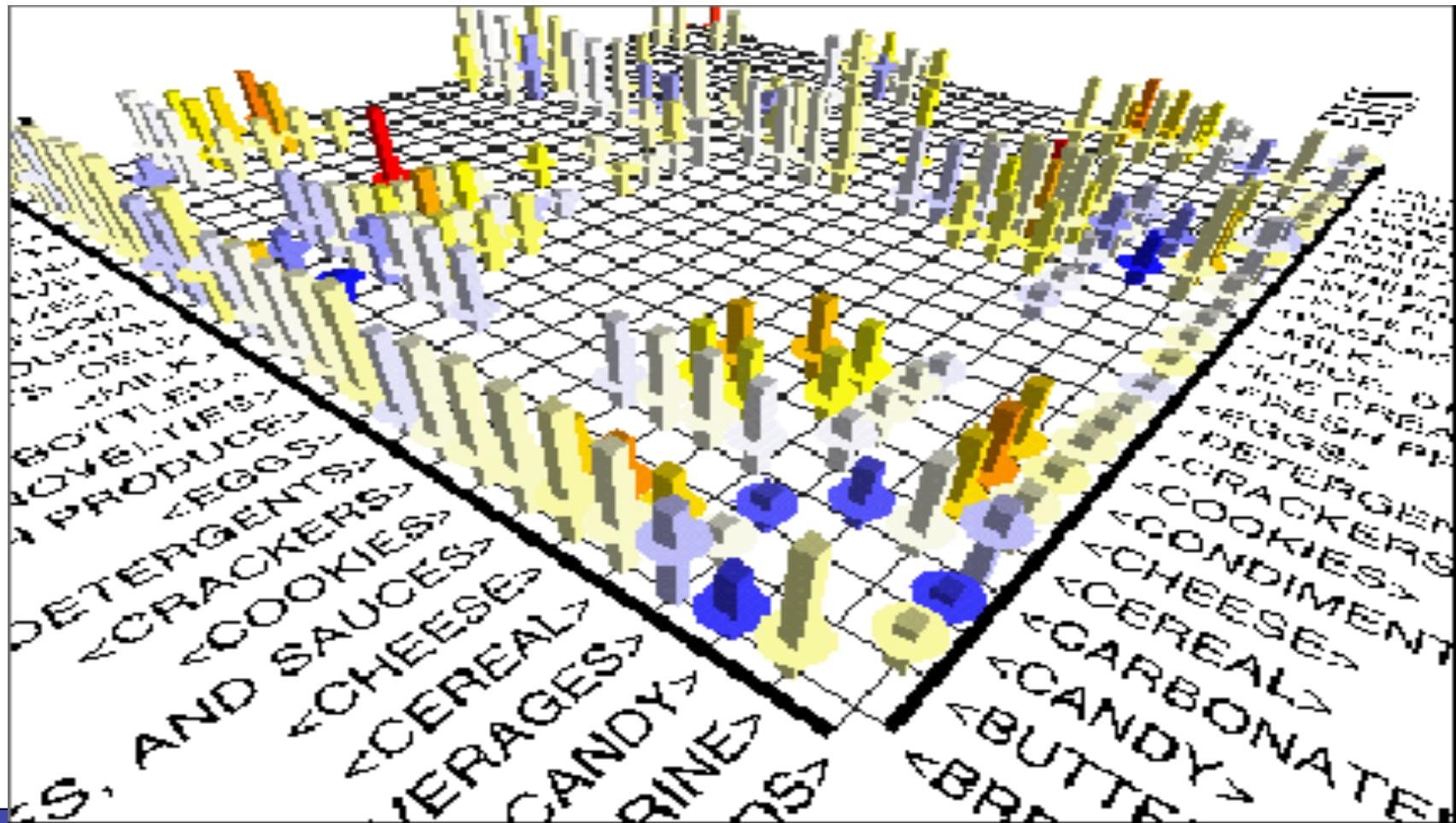
Box-plots dans StatSoft



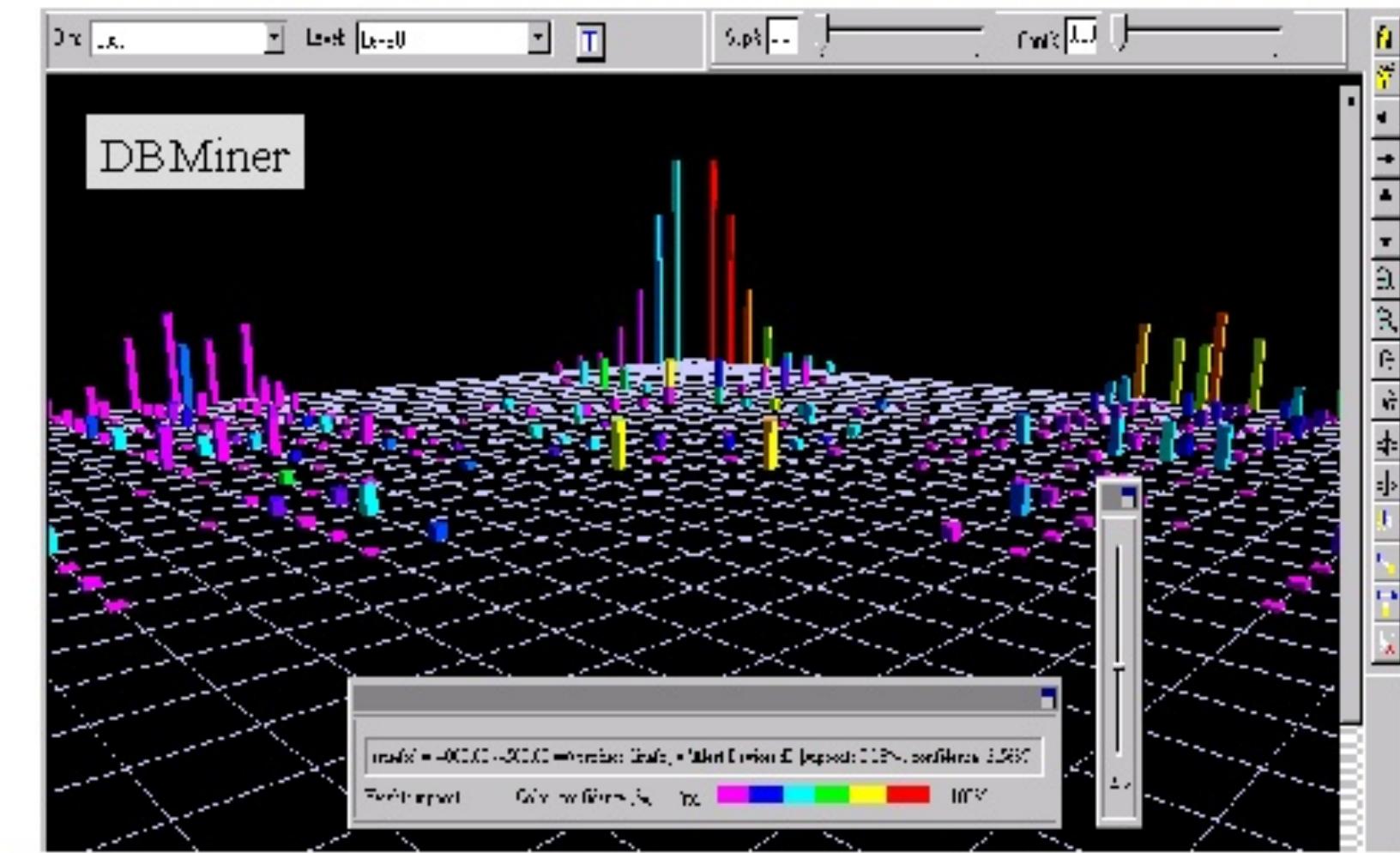
Scatter-plots dans SAS Enterprise Miner



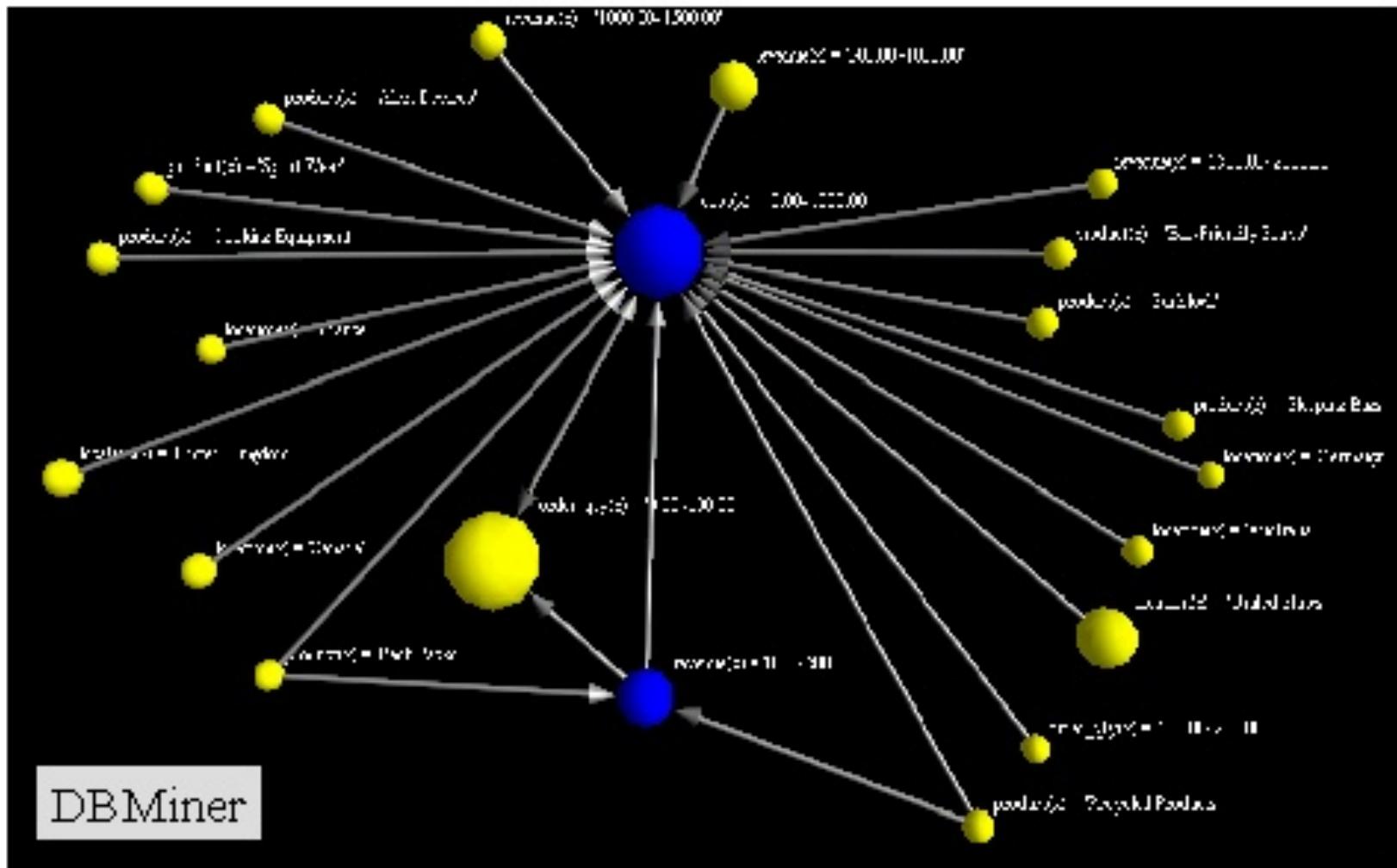
Règles d'association dans MineSet 3.0



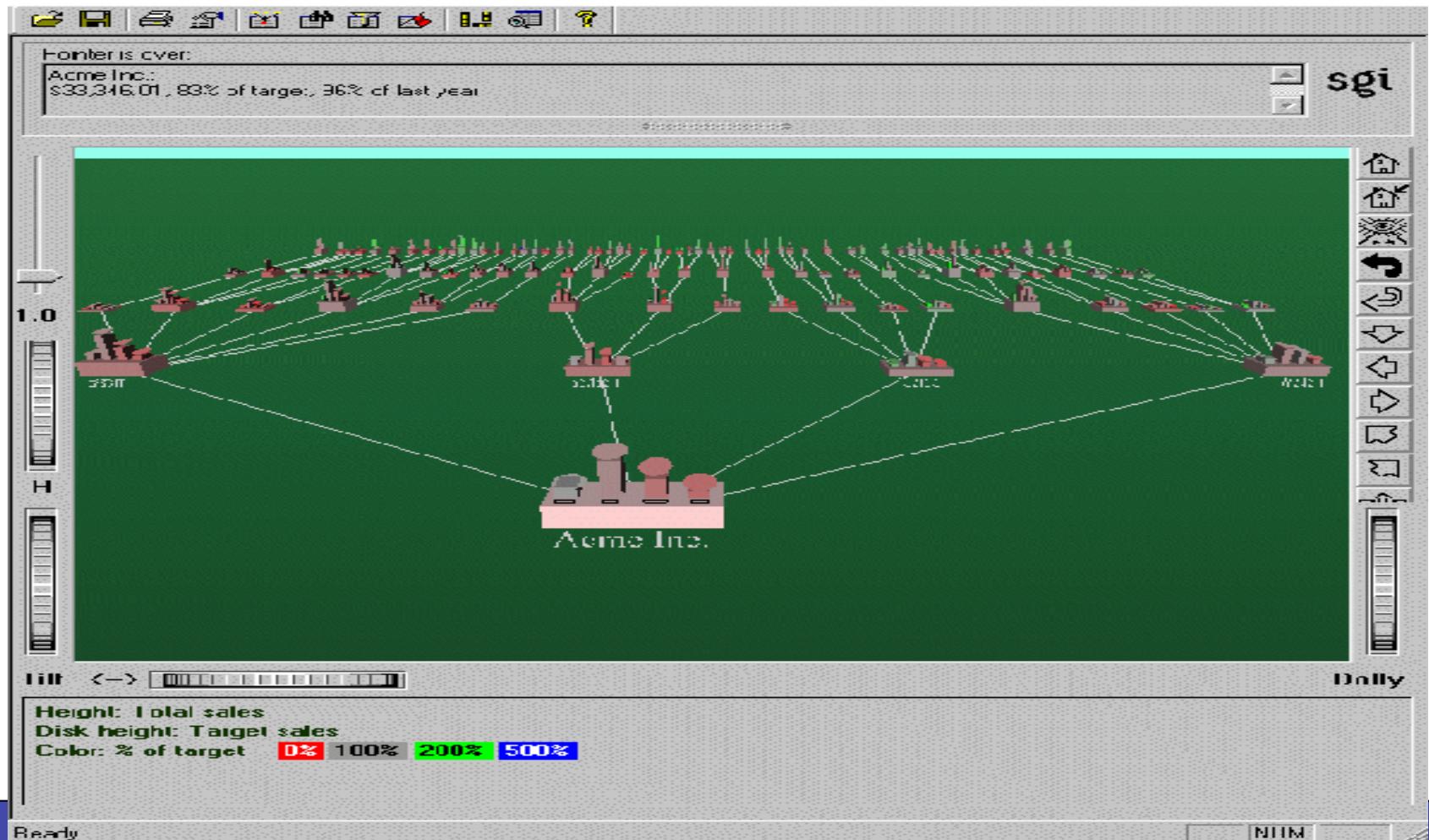
Visualization of Association Rule in Plane Form



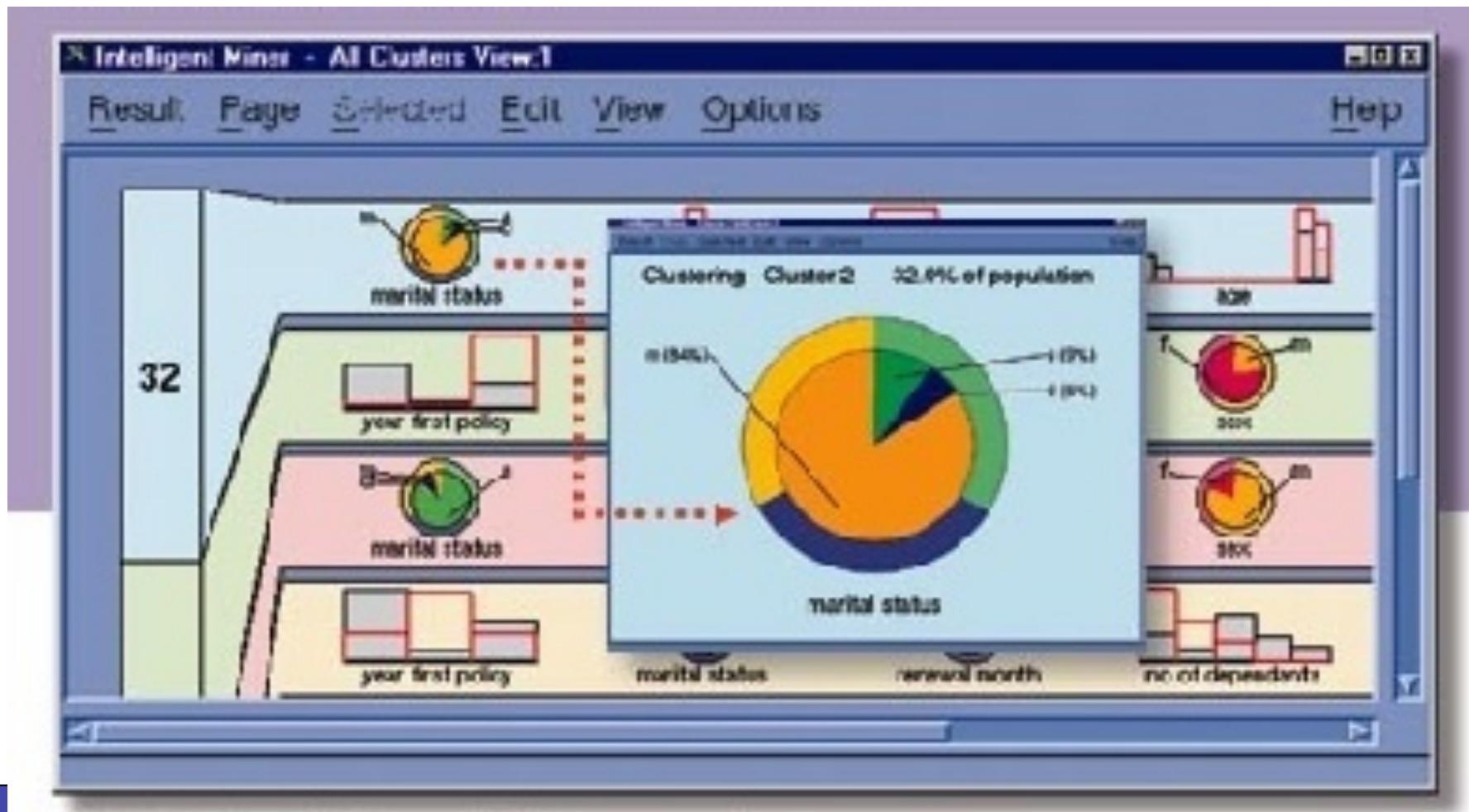
Visualization of Association Rule Using Rule Graph



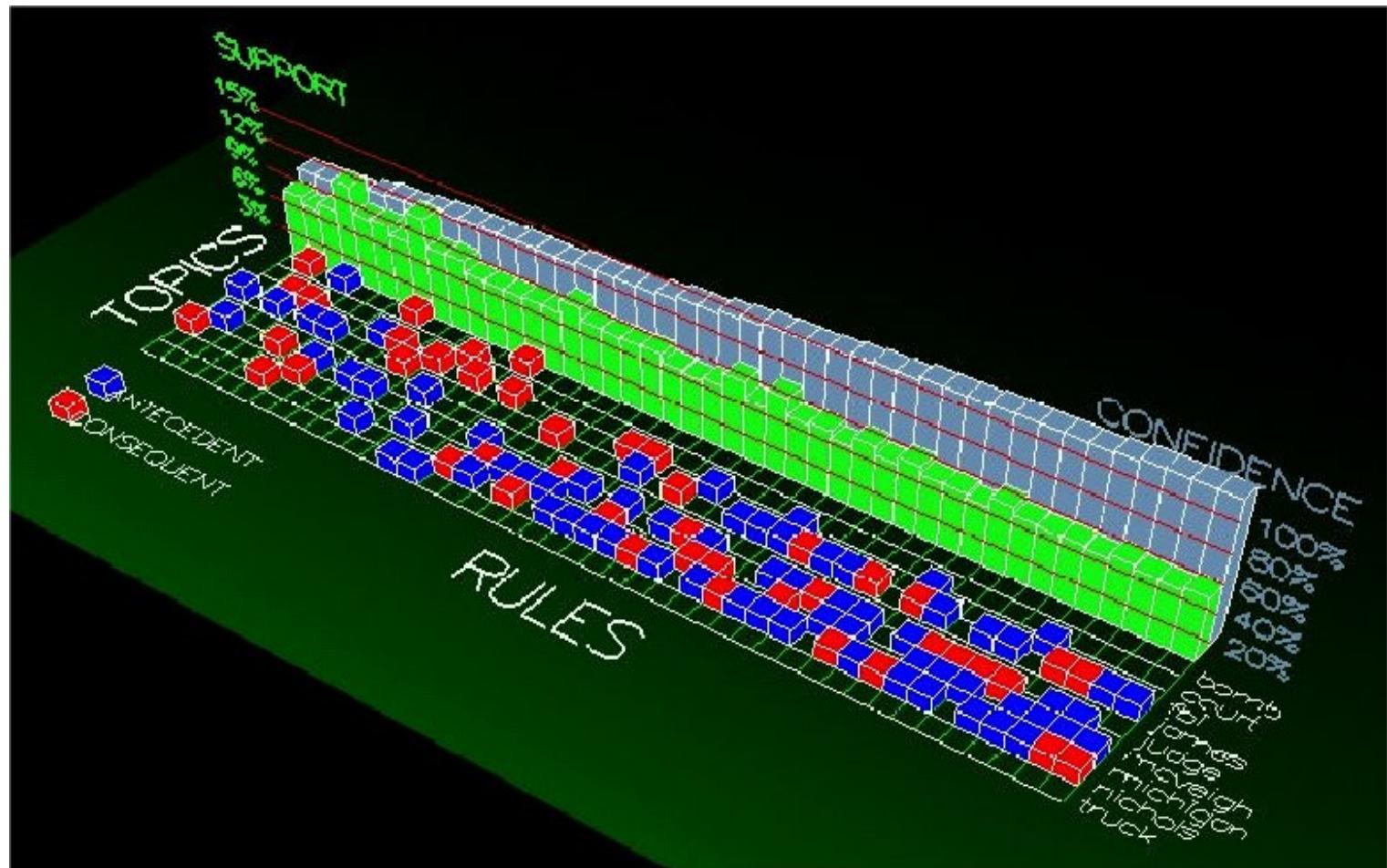
Arbres de décision dans MineSet 3.0



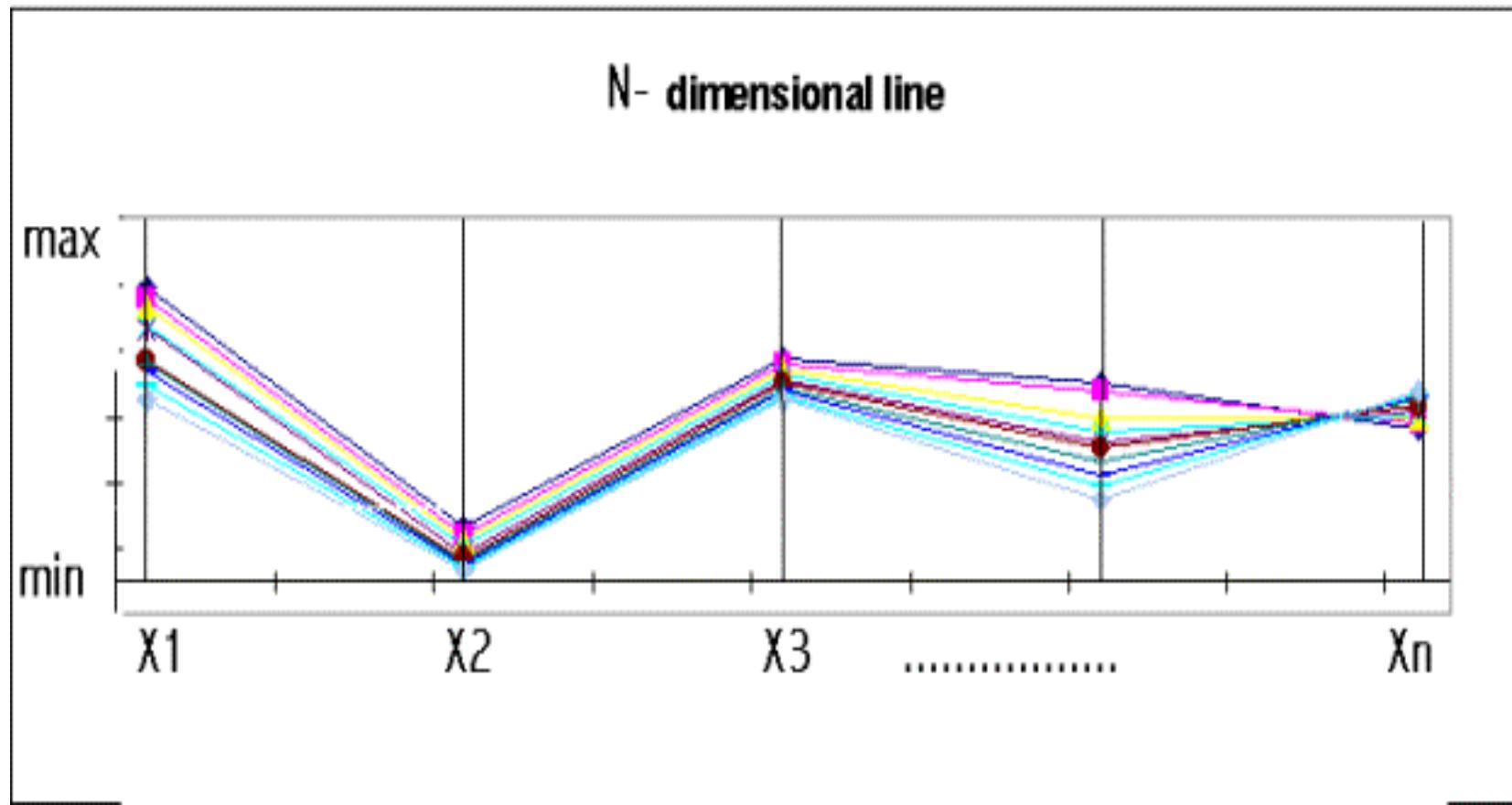
Clusters dans IBM Intelligent Miner



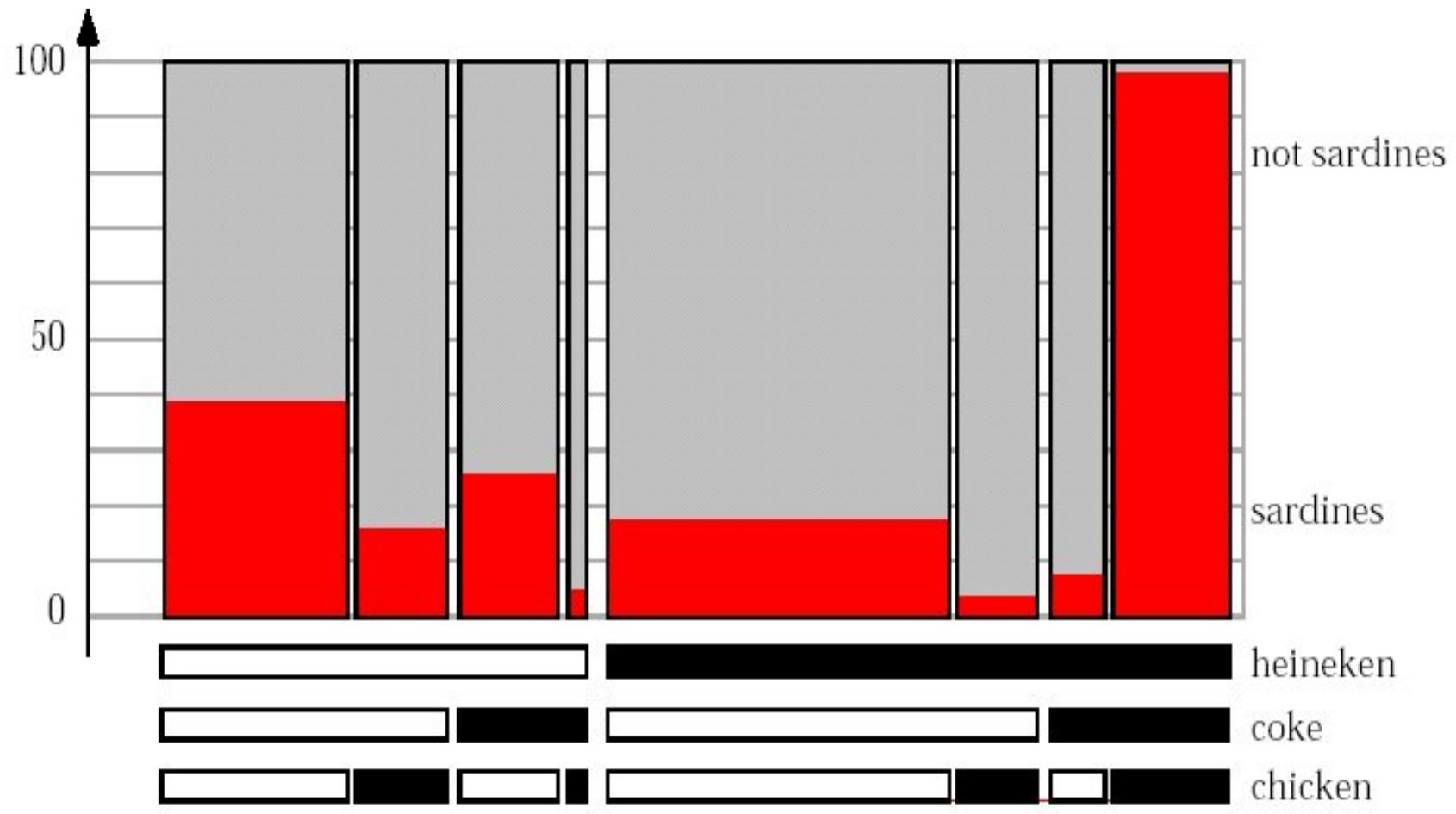
Règles d'association : La visualisation 3D



Règles d'association : Le N-Dimensional Line



Règles d'association : Le Double Decker Plot



Résumé

- **Data mining** : découverte automatique de "patterns" intéressants à partir d'ensembles de données de grande taille
- **KDD (Knowledge discovery) est un processus :**
 - pré-traitement
 - data mining
 - post-traitement
- **Domaines d'application :** distribution, finances, biologie, médecine, télécommunications, assurances, banques, ...

Résumé

- L'information peut être extraite à partir de différentes types de bases de données (relationnel, orienté objet, spatial, WWW, ...)
- Plusieurs fonctions de data mining (différents modèles) : clustering, classification, règles d'association, ...
- Plusieurs techniques dans différents domaines : apprentissage, statistiques, IA, optimisation,

Résumé

- **Plusieurs problèmes ouverts :**
 - Visualisation
 - Parallélisme et distribution
 - Issues de sécurité et confidentialité
- **Futur prometteur ...**