Laboratoire d'Informatique
Fondamentale de Lille

Fouille de données (*Data Mining*)
- Un tour d'horizon -

E-G. Talbi

talbi@lifl.fr

Introduction au Data Mining

- Définition du Data Mining
- Pourquoi le Data Mining ?
- Description du processus KDD (Knowledge Data Discovery)
- Applications
- Tâches et Techniques du Data Mining

Qu'est-ce que le DM ?

- Processus inductif, *itératif* et *interactif* de découverte dans les BD larges de modèles de données *valides*, *nouveaux*, *utiles* et *compréhensibles*.
  - Itératif : nécessite plusieurs passes
  - Interactif : l'utilisateur est dans la boucle du processus
  - Valides : valables dans le futur
  - Nouveaux : non prévisibles
  - Utiles : permettent à l'utilisateur de prendre des décisions
  - Compréhensibles : présentation simple

Notion d'induction [Peirce 1903]

- **Abduction** : diagnostic médical, ...
  - Toutes les voitures ont 4 roues
  - La Peugeot 206 a 4 roues
  - ==> La Peugeot 206 est une voiture

- **Déduction** : Raisonnement qui conclut à partir de prémisses et d'hypothèses à la vérité d'une proposition en usant des règles d'inférence
  - Toutes les voitures ont 4 roues
  - La Peugeot 206 est une voiture
  - ==> La Peugeot 206 a 4 roues

Notion d'induction [Peirce 1903]

- **Induction** : Généralisation d'une observation ou d'un raisonnement établis à partir de cas singuliers.

- Utilisée en Data mining (tirer une conclusion à partir d 'une série de faits, pas sûre à 100%)

  - La clio a 4 roues, La Peugeot 106 a 4 roues, La BMW M3 a 4 roues, La Mercedes 190 a 4 roues
  - ==> Toutes les voitures ont 4 roues

Motivations (1)

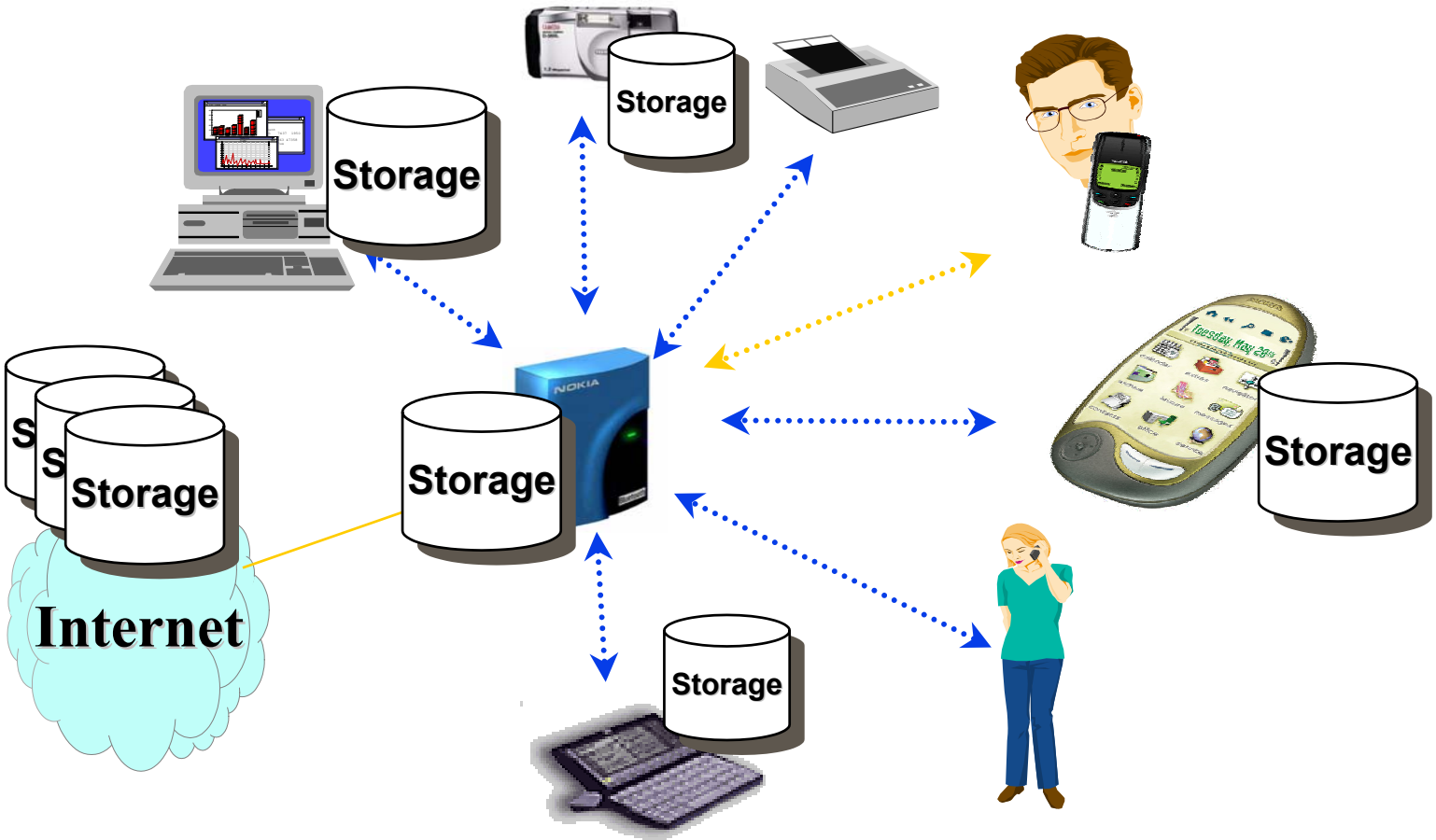
- Explosion des données
  - Masse importante de données (millions de milliards d'instances) : elle double tous les 20 mois.
    - BD très larges - Very Large Databases (VLDB)
  - Données multi-dimensionnelles (milliers d'attributs)
    - BD denses
  - Inexploitables par les méthodes d'analyse classiques
  - Collecte de masses importantes de données (Gbytes/heure)
    - Données satellitaires, génomiques (micro-arrays, …), simulations scientifiques, etc.
  - Besoin de traitement en temps réel de ces données

Motivations (2)

- Améliorer la productivité
  - Forte pression due à la concurrence du marché
  - Brièveté du cycle de vie des produits
  - Besoin de prendre des décisions stratégiques efficaces
    - Exploiter le vécu (données historiques) pour prédire le futur et anticiper le marché
    - individualisation des consommateurs (dé-massification).

- Croissance en puissance/coût des machines capables
  - de supporter de gros volumes de données
  - d'exécuter le processus intensif d'exploration
  - hétérogénéité des supports de stockage

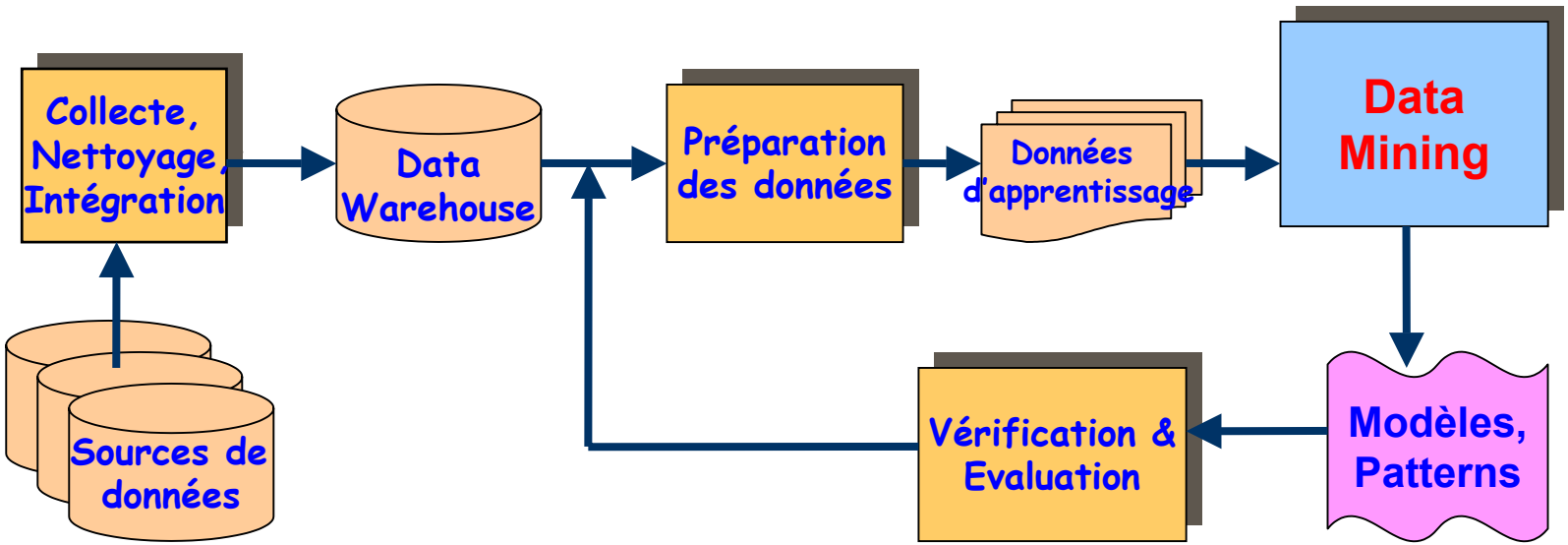
Motivations (3)



Masse importante de données – supports hétérogènes

Le processus de découverte de connaissances

- Data mining : coeur de KDD (Knowledge Data Discovery).



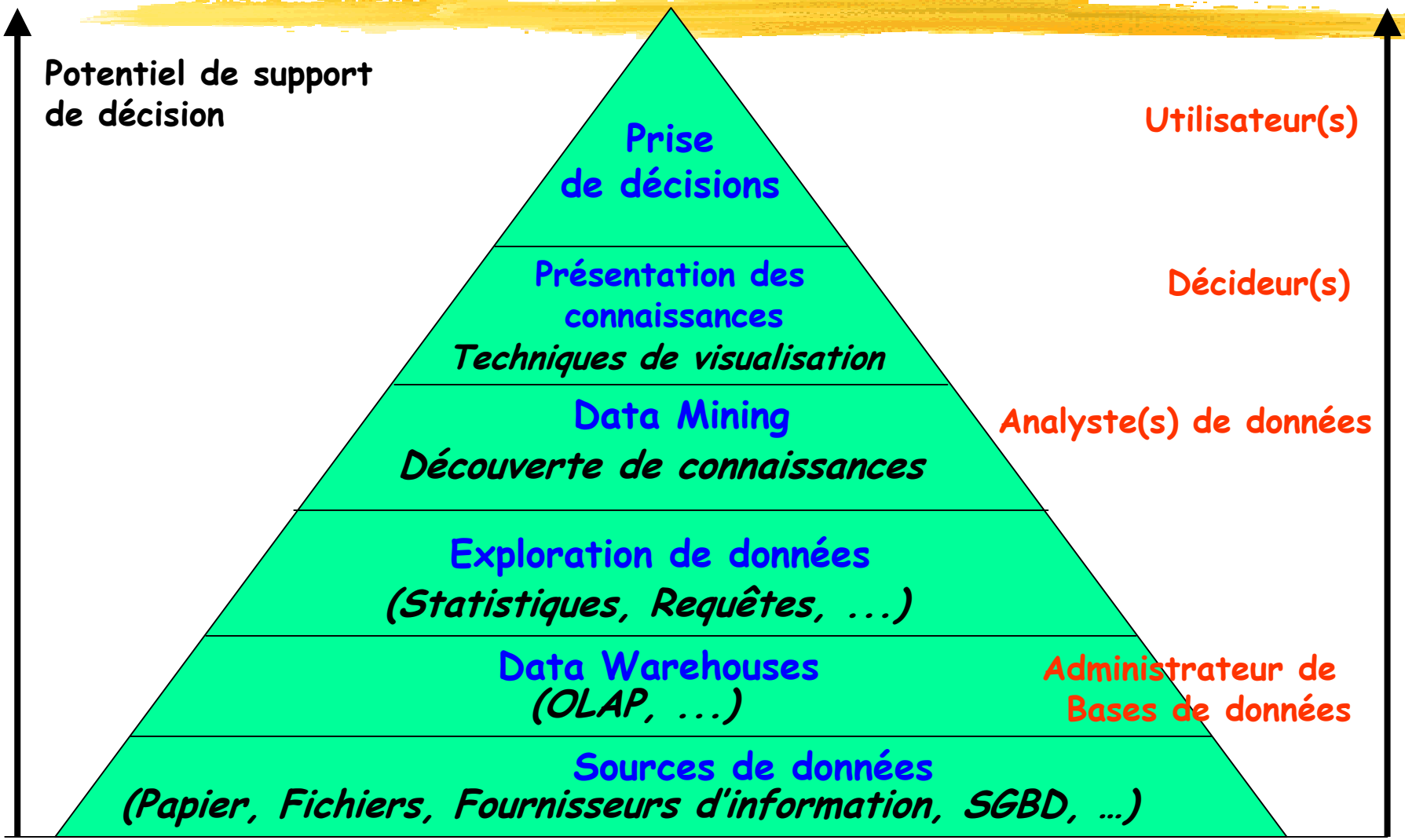
Démarche méthodologique (1)

- 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, etc.
- Appliquer les techniques de fouille de données
  - Choisir le bon algorithme

Démarche méthodologique (2)

- Visualiser, évaluer et interpréter les modèles découverts
  - Analyser la connaissance (intérêt)
  - Vérifier sa validité (sur le reste de la base de données)
  - Réitérer le processus si nécessaire
- Gérer la connaissance découverte
  - La mettre à la disposition des décideurs
  - L'échanger avec d'autres applications (système expert, …)
  - etc.

Data Mining et aide à la décision



Objectifs

- Développer des techniques et systèmes *efficaces* et *extensibles* pour l'exploration de :
  - BD larges et multi-dimensionnelles
  - Données distribuées

- Faciliter l'utilisation des systèmes de DM
  - Limiter l'intervention de l'utilisateur
  - Représentation simple de la connaissance
  - Visualisation sous forme exploitable

Communautés impliquées

- Intelligence artificielle et apprentissage
- Bases de données
- Analyse de données (statistiques)
- Visualisation
- Recherche opérationnelle et optimisation
- Informatique parallèle et distribuée
- Etc.

Data Mining et Statistiques

- Data mining : Exploratoire, Data-driven modeling
- Statistiques : Confirmatoire, User-driven modeling

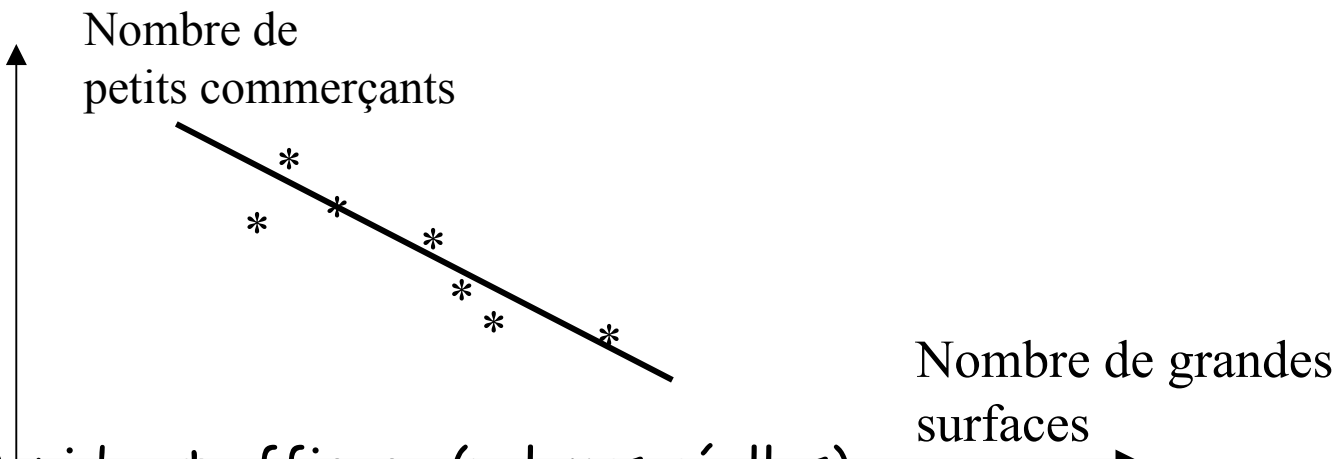
- Distribution d 'une seule variable : moyenne, médiane, variance, écart-type, …

- Explorer les relation entre variables : coefficient de corrélation, …
- Découverte de la cause des relations entre de nombreuses variables est assez complexe.

- test du X2, …
- Réseaux bayésiens (probabilités conditionnelles)

Découverte de modèles fonctionnels

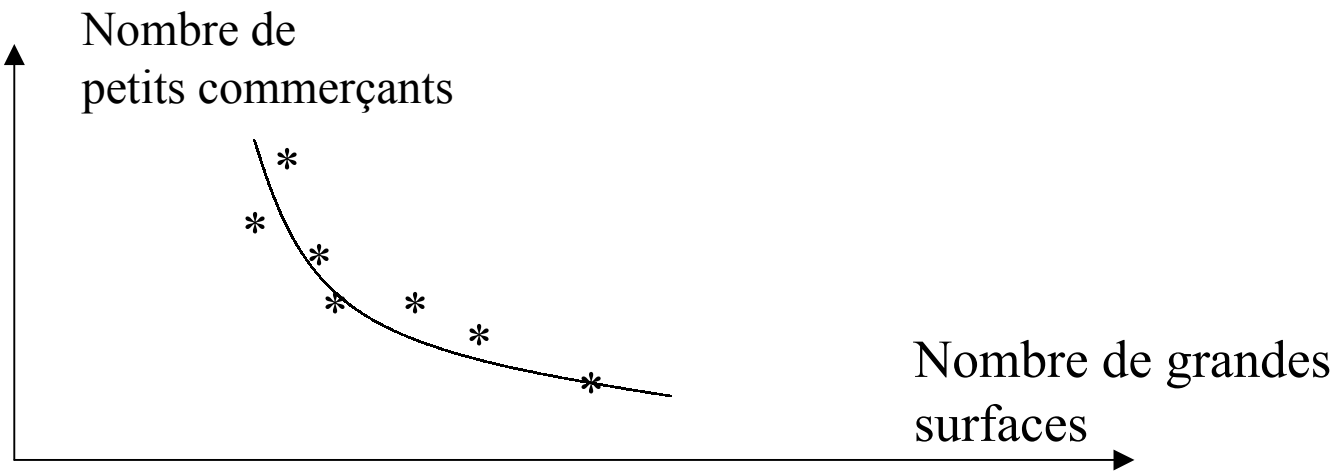
- Méthodes de régression :
  - régression linéaire : Y = aX+ b (a, b : valeurs réelles)



  - Rapide et efficace (valeurs réelles)
  - Insuffisante pour l 'analyse d'espace multidimentionnel

Découverte de modèles fonctionnels

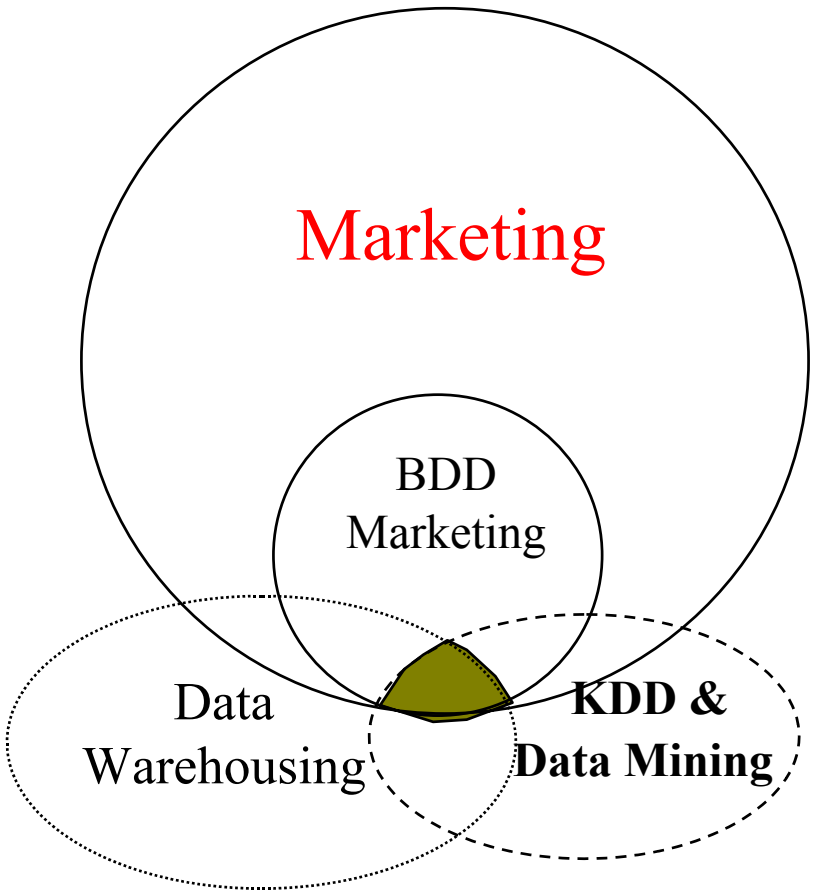
- **Kernel regression :** découvrir graphiquement la fonction à utiliser, peut être une courbe



- Techniques statistiques inadéquates : nombre de facteurs important, modèles non linéaires.

Domaines d'application

- Prise de décision basée sur de nouvelles connaissances
- Ex., impact sur le marketing
- Le rôle et l'importance du KDD et DM est de plus en plus important
- Mais le DM n'est pas seulement dans le marketing...



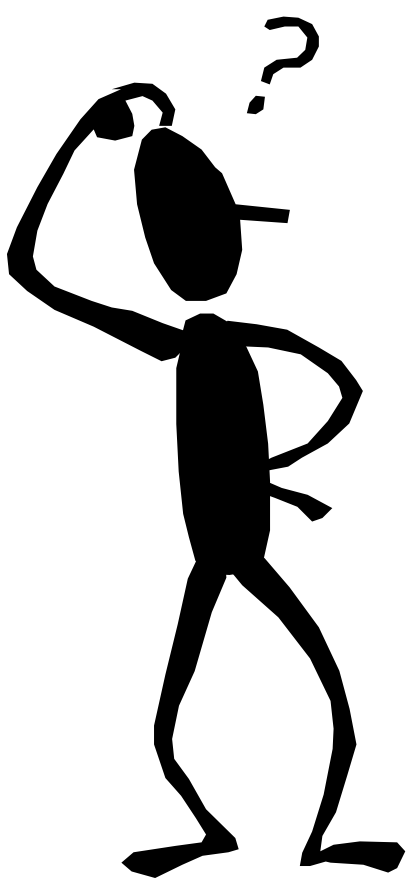
Domaines d'application

- Marketing direct : population à cibler (âge, sexe, profession, habitation, région, …) pour un publipostage.
- Gestion et analyse des marchés : Ex. Grande distribution : profils des consommateurs, modèle d 'achat, effet des périodes de solde ou de publicité, « panier de la ménagère »
- Détection de fraudes : Télécommunications, ...
- Gestion de stocks : quand commander un produit, quelle quantité demander, …
- Analyse financière : maximiser l 'investissement de portefeuilles d 'actions.

Domaines d'application

- Gestion et analyse de risque : Assurances, Banques (crédit accordé ou non)
- Compagnies aériennes
- Bioinformatique et Génome : ADN mining, …
- Médecine et pharmacie :
  - Diagnostic : découvrir d 'après les symptomes du patient sa maladie
  - Choix du médicament le plus approprié pour guérir une maladie donné
- Web mining, text mining, etc.

Exemple 1 - Marketing

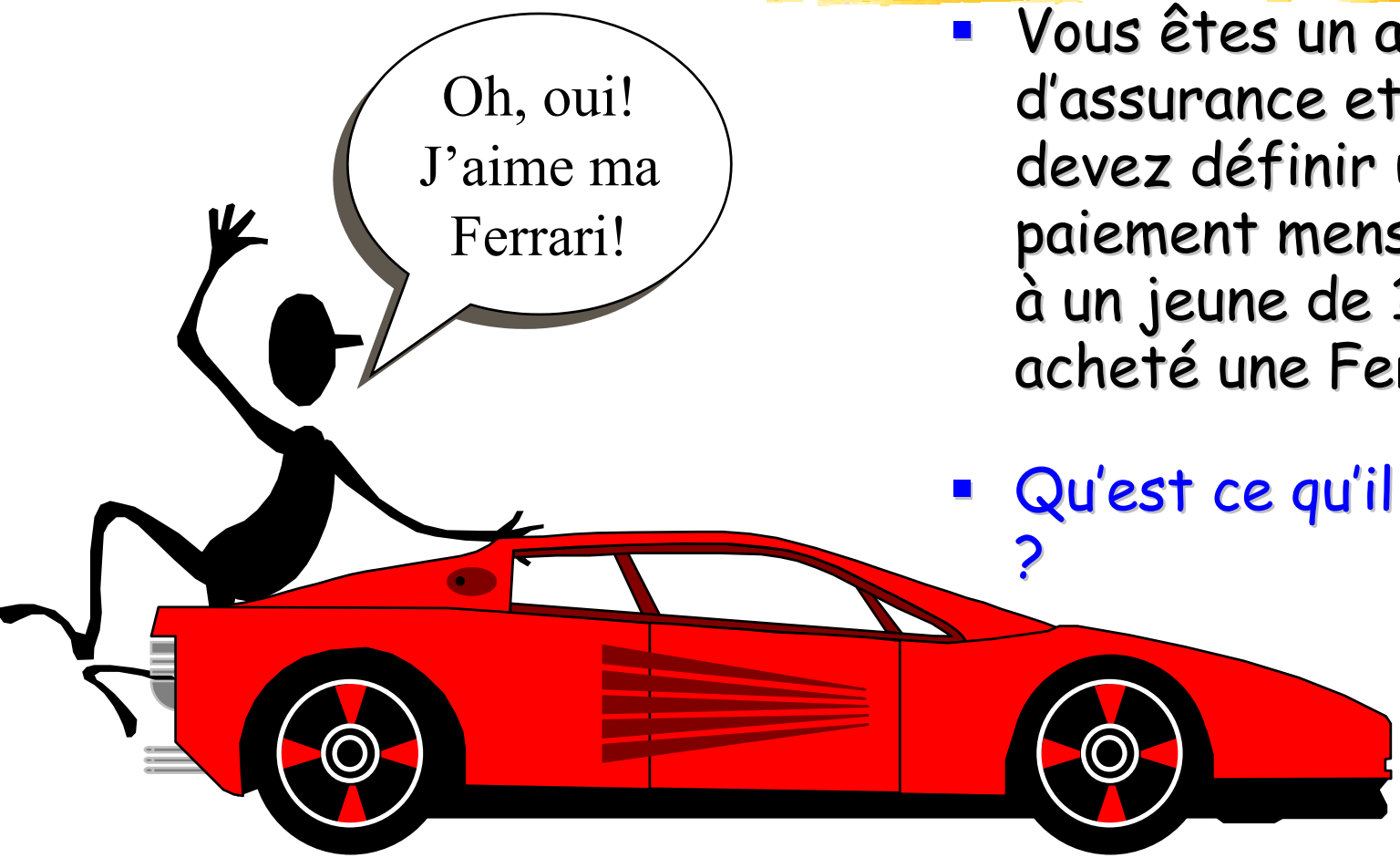


- Vous êtes gestionnaire marketing d'un opérateur de télécommunications mobiles :
  - Les clients recoivent un téléphone gratuit (valeur 150€) avec un contrat d'un an ; vous payer une commission de vente de 250€ par contrat
  - Problème : Taux de renouvellement (à la fin du contrat) est de 25%
  - Donner un nouveau téléphone à toute personne ayant expirer son contrat coûte cher.
  - Faire revenir un client après avoir quitter est difficile et coûteux.

Exemple 1 - Marketing

- Trois mois avant l'expiration du contrat, prédire les clients qui vont quitter :
  - Si vous voulez les garder, offrir un nouveau téléphone.

Exemple 2 - Assurances



- Vous êtes un agent d'assurance et vous devez définir un paiement mensuel adapté à un jeune de 18 ans qui a acheté une Ferrari.
- Qu'est ce qu'il faut faire ?

Exemple 2 - Assurances

- **Analyser les données de tous les clients de la compagnie.**
- **La probabilité d'avoir un accident est basée sur … ?**
  - Sexe du client (M/F) et l'âge
  - Modèle de la voiture, âge, adresse, ….
  - etc.
- **Si la probabilité d'avoir un accident est supérieure à la moyenne, initialiser la mensualité suivant les risques.**

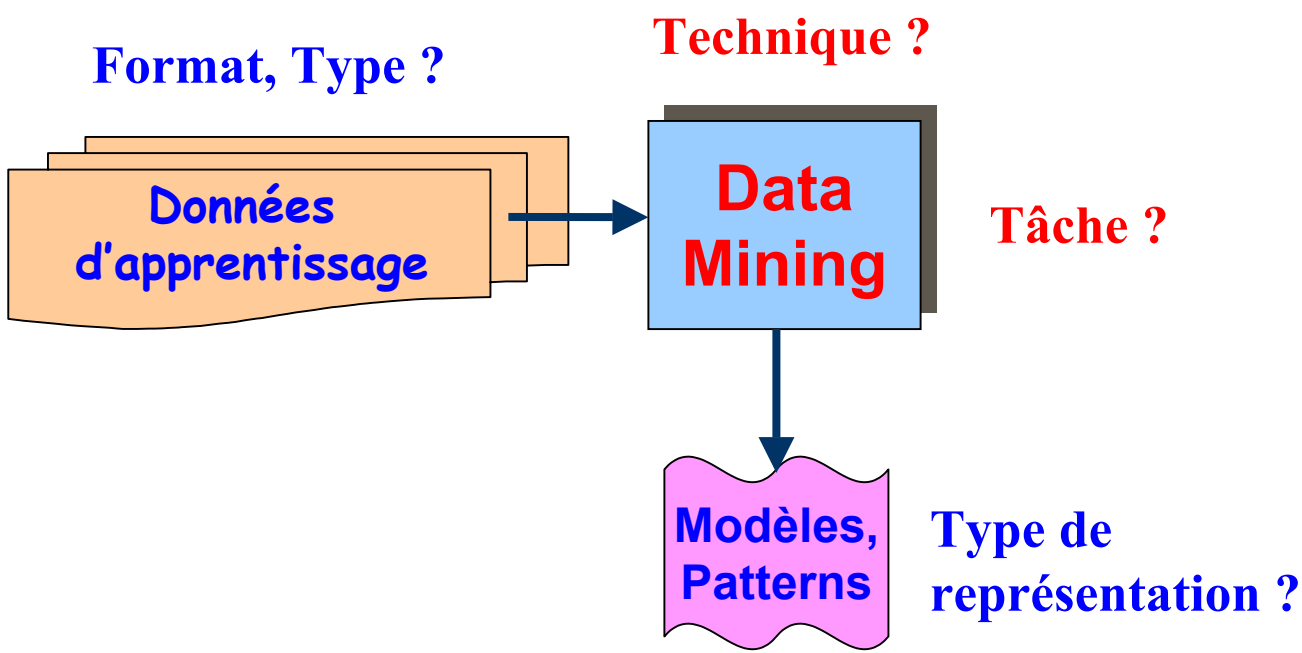
Exemple 3 – Banque - Télécom

- **Vous êtes à l'étranger et quelqu'un a volé votre carte de crédir ou votre mobile …**
- **compagnies bancaires …**
  - Utiliser les données historiques pour construire un modèle de comportement frauduleux et utiliser le data mining pour identifier des instances similaires.
- **compagnies téléphoniques …**
  - Analyser les "patterns" qui dérivent du comportement attendu (destinataire, durée, etc.)

Exemple 4 - Web

- **Les logs des accés Web sont analysés pour …**
  - Découvrir les préférences des utilisateurs
  - Améliorer l'organisation du site Web
- **De manière similaire …**
  - L'analyse de tous les types d'informations sur les logs
  - Adaptation de l'interface utilisateur/service

Paramètres d'un processus KDD



Les données

- Valeurs des champs des enregistrements des tables de l'entropot (base de données)
- 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, …

Tâches du Data Mining

- Classification
- Clustering (Segmentation)
- Recherche d'associations
- Recherche de séquences
- Détection de déviation

Classification

- Elle permet de prédire si une instance de donnée est membre d'un groupe ou d'une classe prédéfinie.

- Classes
  - Groupes d'instances avec des profils particuliers
  - Apprentissage supervisé : classes connues à l'avance

  - Applications : marketing direct (profils des consommateurs), grande distribution (classement des clients), médecine (malades/non malades), etc.
  - Exemple : les acheteurs de voitures de sport sont de jeunes citadins ayant un revenu important

Clustering (Segmentation)

- Partitionnement logique de la base de données en clusters
  - Clusters : groupes d'instances ayant les mêmes caractéristiques
  - Apprentissage non supervisé (classes inconnues)
  - Pb : interprétation des clusters identifiés
  - Applications : Economie (segmentation de marchés), médecine (localisation de tumeurs dans le cerveau), etc.

Règles d'association

- Corrélations (ou relations) entre attributs (méthode non supervisée)
- Applications : grande distribution, gestion des stocks, web (pages visitées), etc.
- Exemple
  - BD commerciale : panier de la ménagère
  - Articles figurant dans le même ticket de caisse
  - Ex : achat de riz + vin blanc ==> achat de poisson
  - Achats bières et couches-culottes (USA, Week-end)

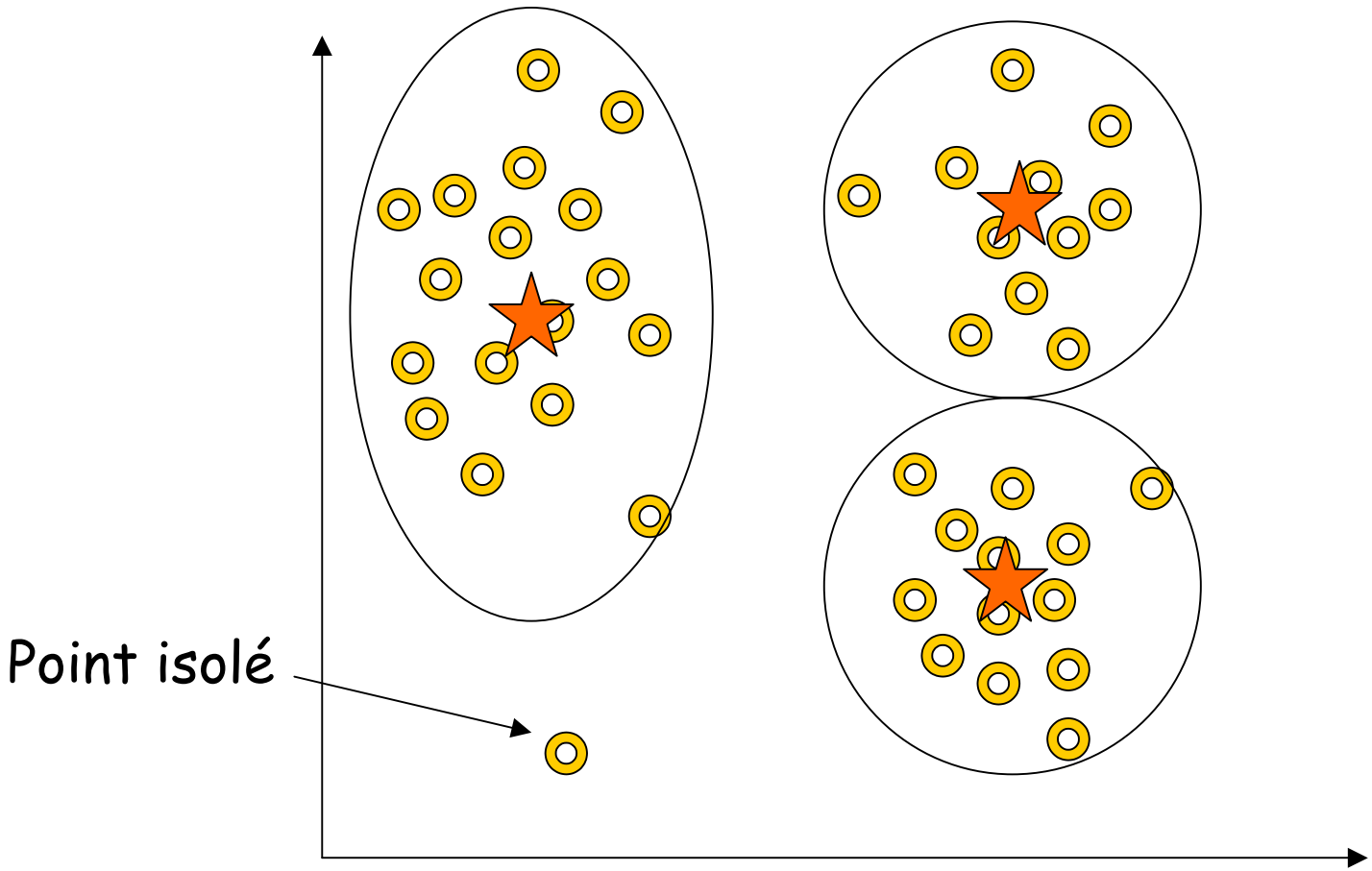
Recherche de séquences

- Recherche de séquences
  - Liaisons entre événements sur une période de temps
  - Extension des règles d'association
    - Prise en compte du temps (série temporelle)
    - Achat Télévision ==> Achat Magnétoscope d'ici 5 ans
  - Applications : marketing direct (anticipation des commandes), bioinformatique (séquences d'ADN), bourse (prédiction des valeurs des actions)
- Exemple
  - BD commerciale (ventes par correspondance)
  - Commandes de clients
  - Ex : 60% des consommateurs qui commandent la bière «Mort subite» commandent de l'aspro juste après
  - Séquences d'AND : ACGTC est suivie par GTCA après un gap de 9, avec une probabilité de 30%

Détection de déviation

- Instances ayant des caractéristiques les plus différentes des autres
  - Basée sur la notion de distance entre instances
  - Expression du problème
    - Temporelle : évolution des instances ?
    - Spatiale : caractéristique d'un cluster d'instances ?
- Applications
  - Détection de fraudes (transactions avec une carte bancaire inhabituelle en telemarketing)
- Caractéristiques
  - Problème d'interprétation : bruit ou exception (donc connaissance intéressante)

Illustration



Techniques utilisées

- K-moyennes, A-priori, K-NN
- Réseaux de neurones
- Algorithmes génétiques
- Chaînes de Markov cachées
- Arbres de décision
- Réseaux bayesiens
- Soft computing : ensembles flous
- …

Résumé - Introduction

- Data mining : découverte automatique de modèles intéressants à partir d'ensemble de données de grande taille
- KDD (knowledge data discovery) est un processus :
  - Pré-traitement (Pre-processing)
  - Data mining
  - Post-traitement (Post-processing)
- Pour le data mining, utilisation de différents …
  - Base de données (relationelle, orientée objet, spatiale, WWW, …)
  - Connaissances (classification, clustering, association, …)
  - Techniques (apprentissage, statistiques, optimisation, …)
  - Applications (génomique, télécom, banque, assurance, distribution, …)

Travaux pratiques :

Cadre du travail

WEKA 3.2

Waikato Environment for Knowledge Analysis

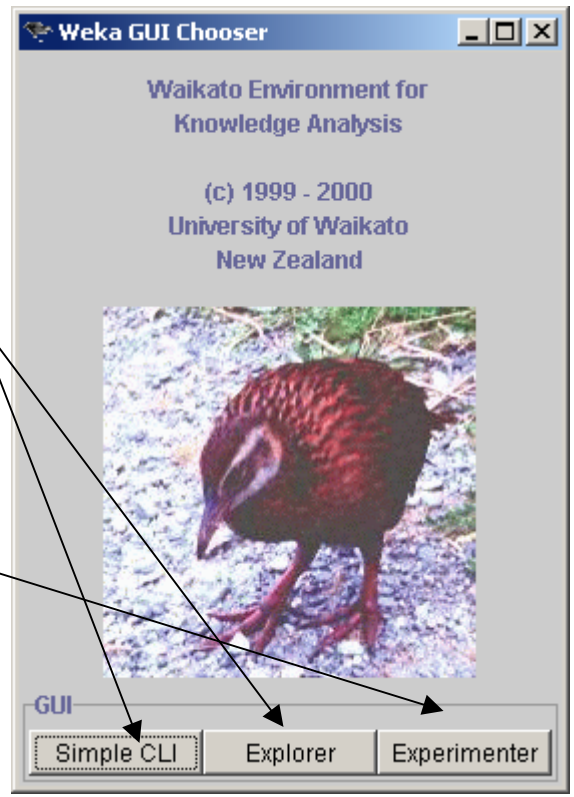
http://www.cs.waikato.ac.nz/ml/weka/
http://www.lifl.fr/~jourdan

WEKA

- Logiciel gratuit disponible sur le web : *http://www.cs.waikato.ac.nz/ml/weka/*
- Plate forme logicielle en Java tournant sous :
  - Windows
  - Linux
- Facile à prendre en main

WEKA

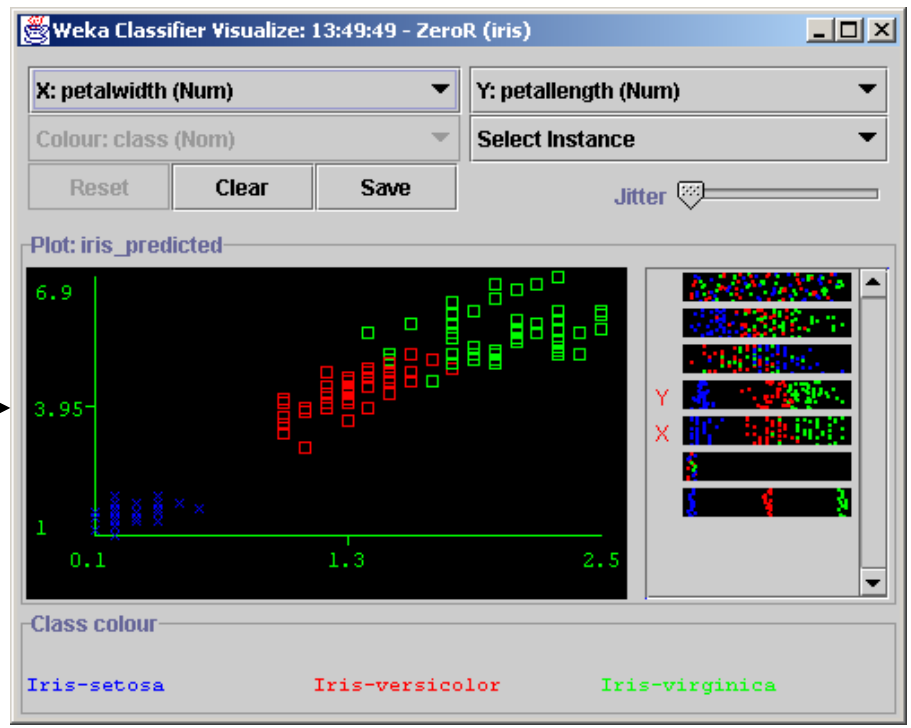
- Interface en ligne de commande
- Explorer (interface graphique)
  - Filtre
  - Apprentissage (clustering, classification, ...)
  - Sélection d'attributs
  - Visualisateur de données et de résultats
- Expérimenter (environnement d'expérience)
  - Test d'une méthode spécifique sur un ensemble de données avec des critères variés pour la comparaison de résultats



WEKA

- En entrée : fichiers, base de données, Url
- En sortie : affichage des résultats, sortie des résultats dans des fichiers, visualisation graphique …

Exemple de visualisation après une classification : une couleur représente une classe



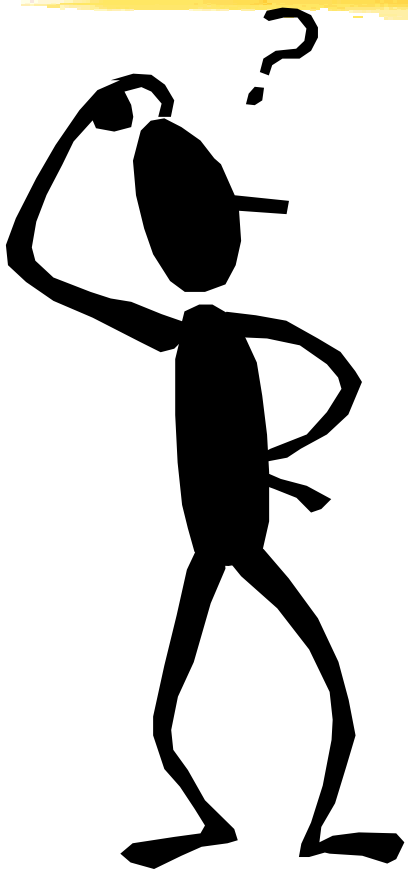
Weka - Explorer

Les fonctions disponibles :

- Filtre et Preprocess sur les données
- Classification
- Clustering
- Règles d'association
- Sélection d'attributs
- Visualisateur

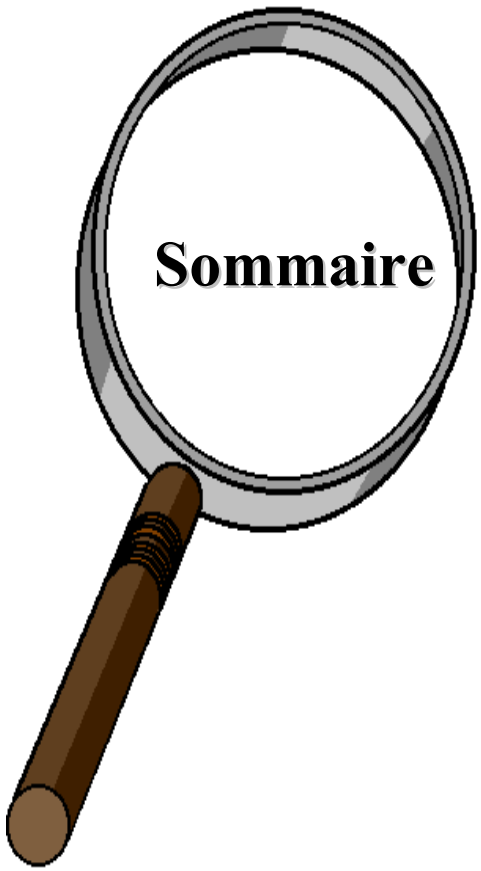
Plan du cours

- Clustering
- Classification
- Règles d'association
- Outils pour le Data Mining



Clustering (Segmentation)

Clustering - Plan



- Problèmatique du clustering
- Applications
- Similarité et types de données
- Méthodes de clustering
  - Méthodes de partitionnement
  - Méthodes hiérarchiques
  - Méthodes par voisinage dense
- Application réelle en génomique
- Résumé

Problèmatique

- 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)

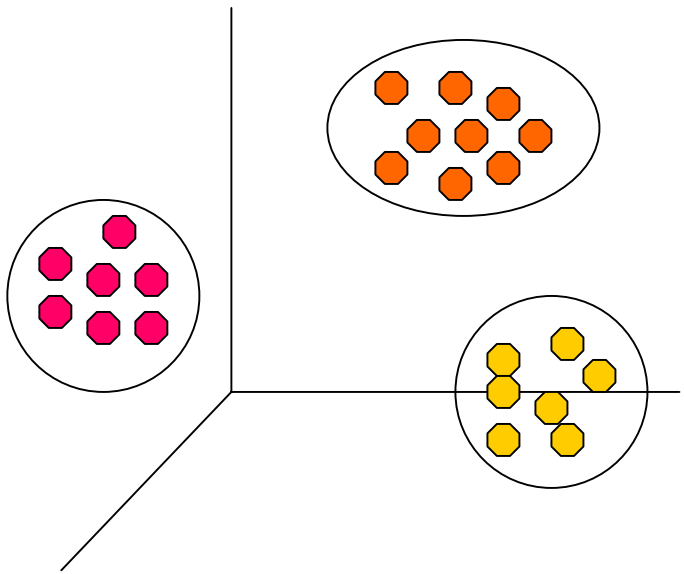
Qualité d'un clustering

- 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 abilité à découvrir certains ou tous les "patterns" cachés.

Objectifs du clustering

Minimiser les distances intra-cluster

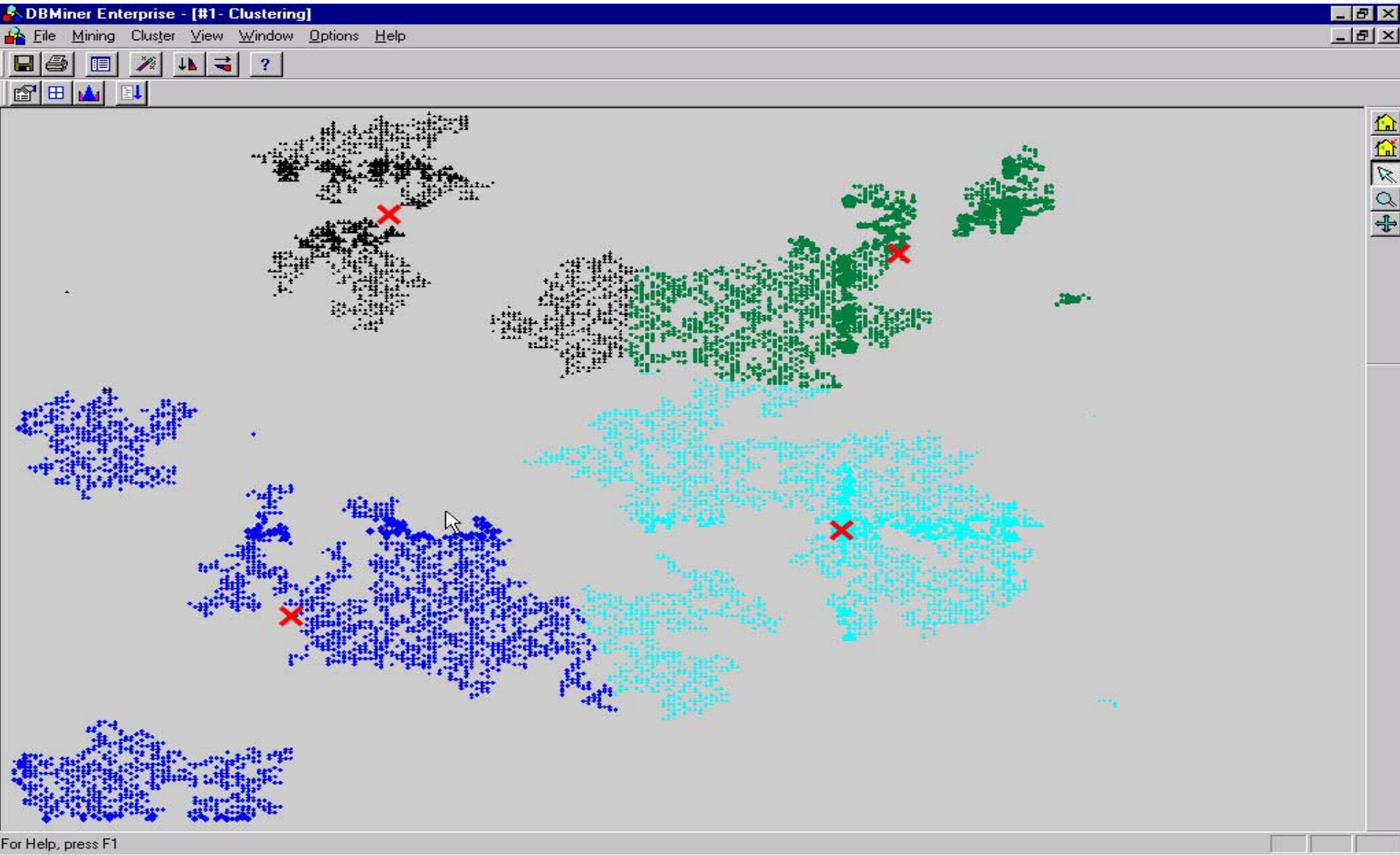
Maximiser les distances inter-clusters



Exemples d'applications

- **Marketing** : segmentation du marché en découvrant des groupes de clients distincts à partir de bases de doneées 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



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

Choix de la distance

- Propriétés d'une distance :

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \text{ iff } x = y$
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|/dmax (distance normalisée).
- Exemple : Age, taille, poids, …

Distance – Données numériques

- Combiner les distances : Soient x=(x1,…,xn) et y=(y1, …,yn)
- 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 : $d(x,y)=\sqrt[q]{\sum_{i=1}^{n}|x_i-y_i|^q}$

k=1 : distance de Manhattan.

k=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 (dissimilarité)

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

- **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**):

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

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.

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

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

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

Distance – Données énumératives

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

- Méthode 1: **correpondance simple**
  - **m**: # de correspondances, **p**: # total de variables

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

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 : $d(x,y)=d1(x1,y1) + \ldots + dn(xn,yn)$

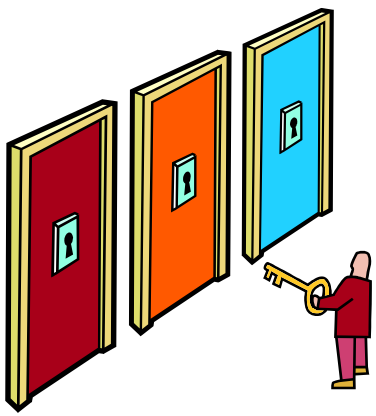
Données mixtes - Exemple 1

- Base de données « Cancer du sein »
  http://www1.ics.uci.edu/~mlearn/MLSummary.html
- #instances = 286 (Institut Oncologie, Yugoslavie)
- # attributs = 10
  - Classe : no-recurence-events, recurrence-events
  - Age : 10-19, 20-29, 30-39, 40-49, …, 90-99
  - Menopause : Lt40, Ge40, premeno
  - Taille de la tumeur : 0-4, 5-9, 10-14, …, 55-59
  - Inv-nodes : 0-2, 3-5, 6-8, …, 36-39 (ganglions lymphatiques)
  - Node-caps : Oui, Non
  - Deg-malig : 1, 2, 3 (Dégré de malignité)
  - Sein : Gauche, Droit
  - Breast-quad : left-up, left-low, right-up, right-low, central
  - Irradiation : Oui, Non

Données mixtes – Exemple 2

- Base de données « Diabète » : Diagnostic (OMS)
  http://www1.ics.uci.edu/~mlearn/MLSummary.html
- #instances = 768 (Arizona, USA)
- # attributs = 8
  - Nombre de grossesses
  - Concentration du taux de glucose dans le plasma
  - Pression sanguine diastolique (mm Hg)
  - Epaisseur de la graisse du triceps (mm)
  - Taux d'insuline après 2 heures (repas) (mu U/ml)
  - Indice de masse corporelle (poids en kg / (taille en m)^2)
  - Fonction « Diabete pedigree »
  - Age (ans)
  - Classe (Positif ou Négatif)

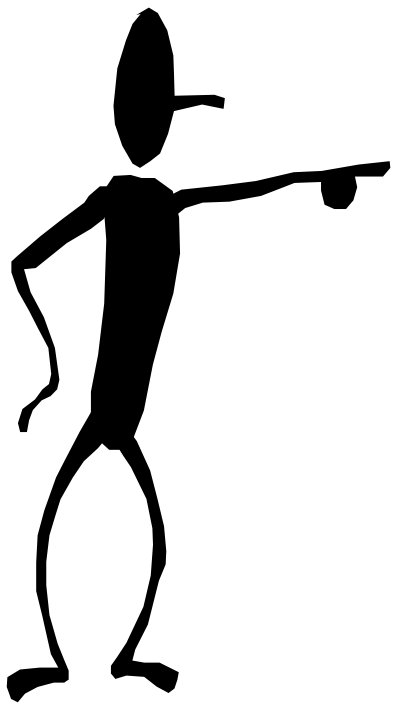
Méthodes de Clustering



- Méthode de partitionnement (K-moyennes)
- Méthodes hiérarchiques (par agglomération)
- Méthode par voisinage dense

- Caractéristiques
  - Apprentissage non supervisé (classes inconnues)
  - Pb : interprétation des clusters identifiés

Méthodes de clustering - Caractéristiques



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

Algorithme des k-moyennes (K-means)

- Entrée : un échantillon de m enregistrements x1, …, xm
- 1. Choisir k centres initiaux c1, …, ck
- 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, ci est la moyenne des éléments du groupe i.
- Aller en 2.

Illustration (1)

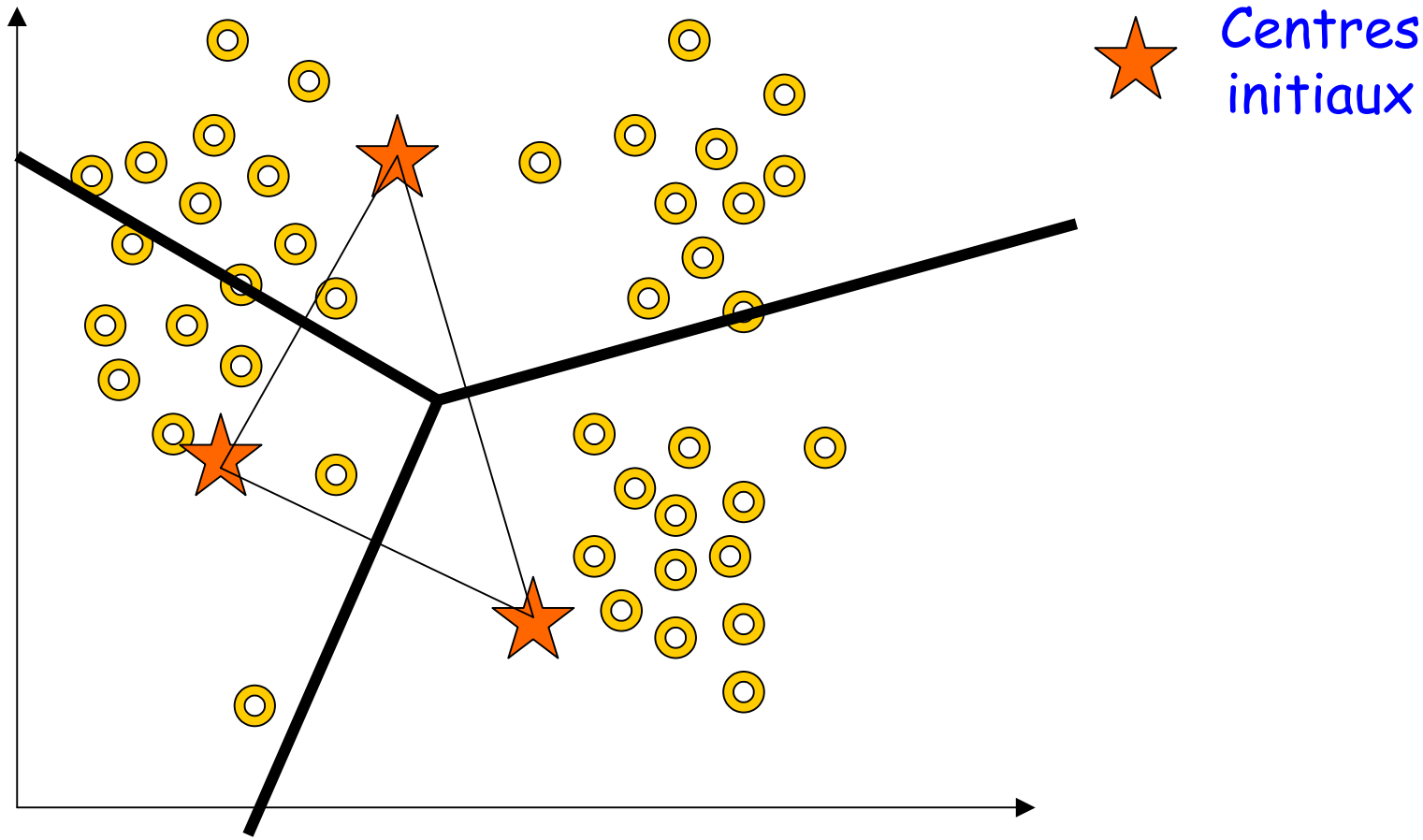
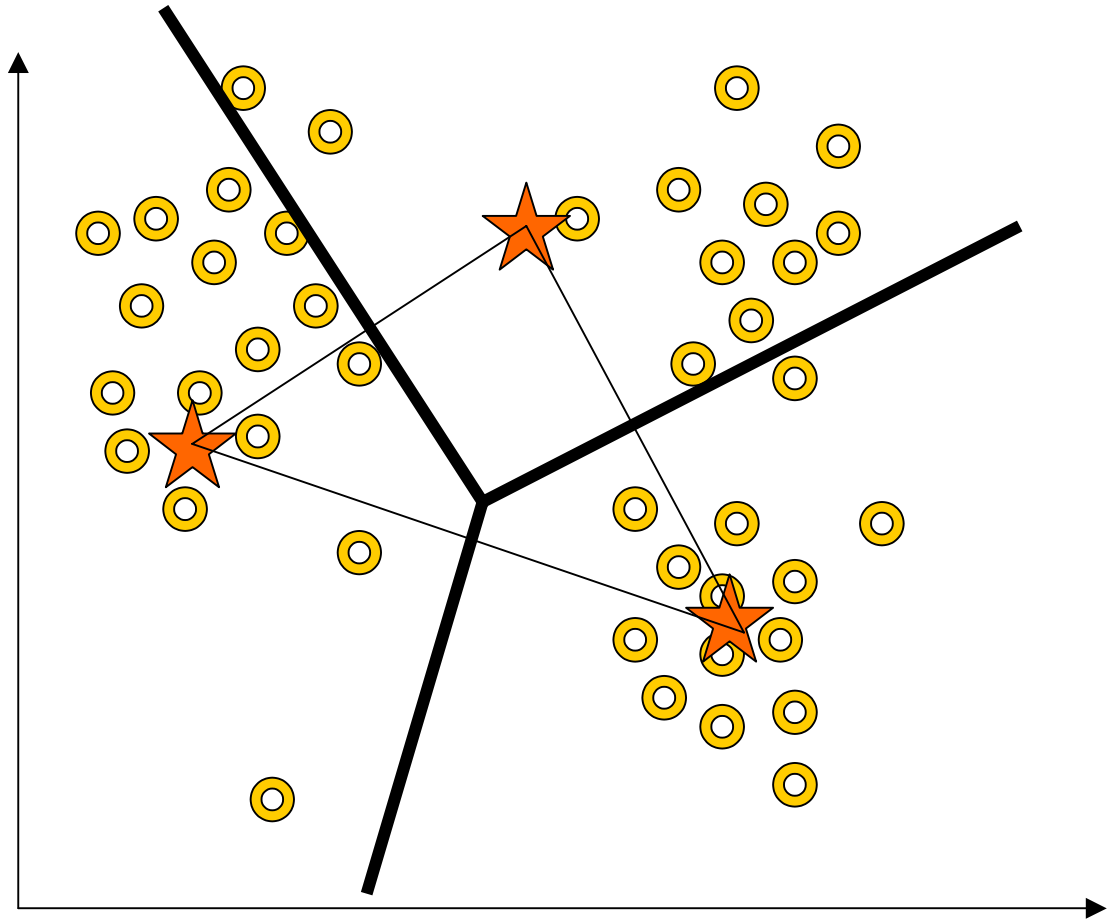
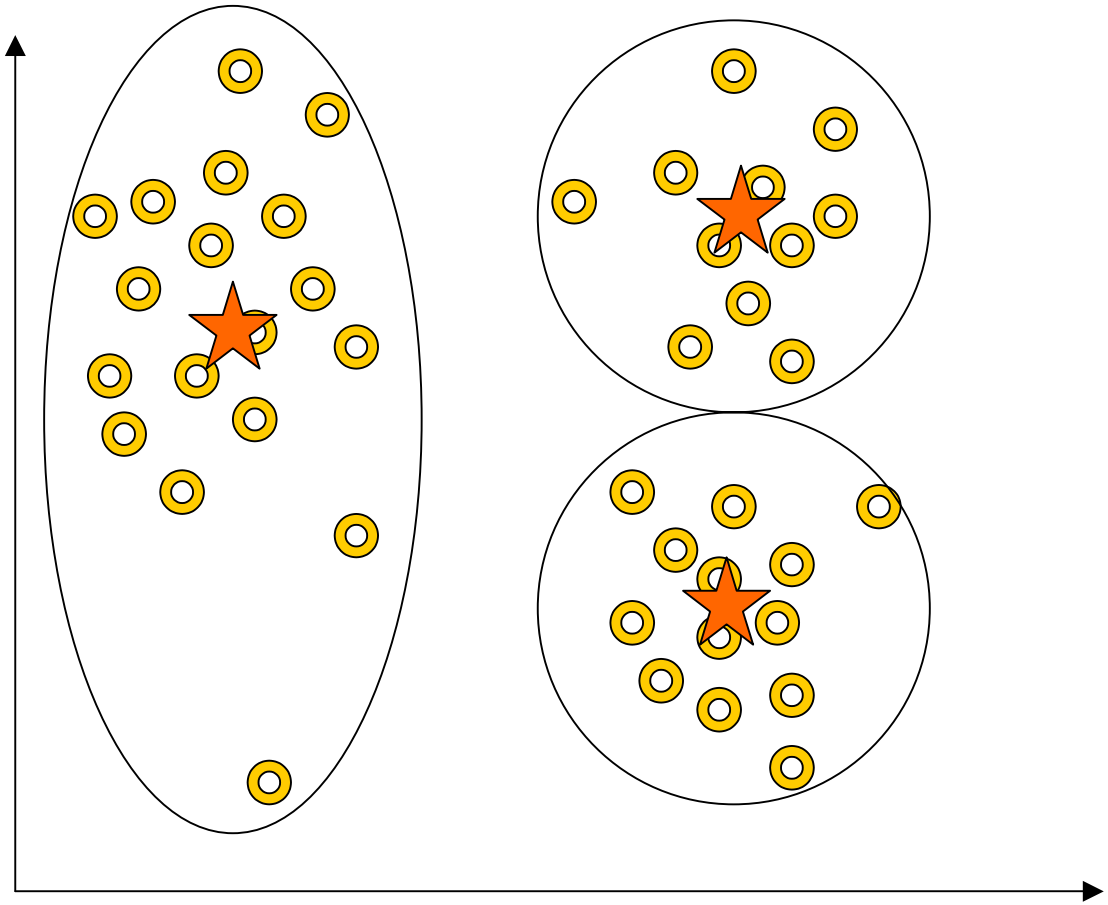


Illustration (2)



Nouveaux centres

Illustration (3)



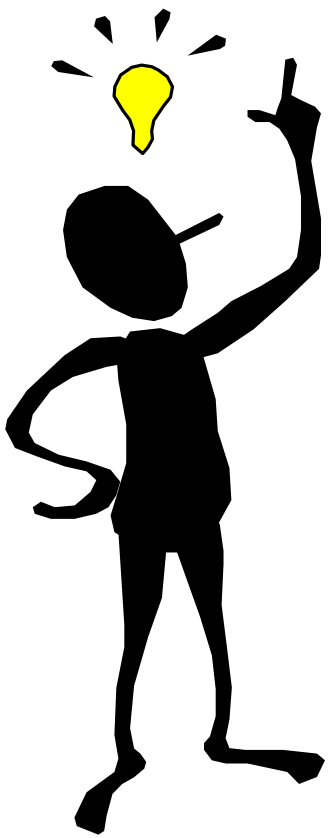
Centres finaux

Algorithme des k-moyennes : Exemple

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

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

K-moyennes : Avantages

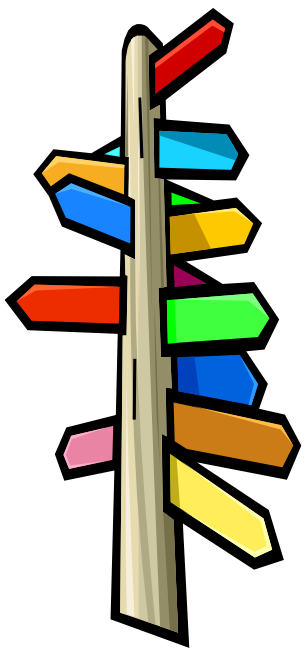


- **Relativement extensible** dans le traitement d'ensembles de taille importante
- **Relativement efficace** : $O(t.k.n)$, où n représente # objets, k # clusters, et t # iterations. Normalement, $k, t << n$.
- Produit généralement un **optimum local** ; un optimum global peut être obtenu en utilisant d'autres techniques telles que : algorithmes génétiques, …

K-moyennes : Inconvénients

- **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)

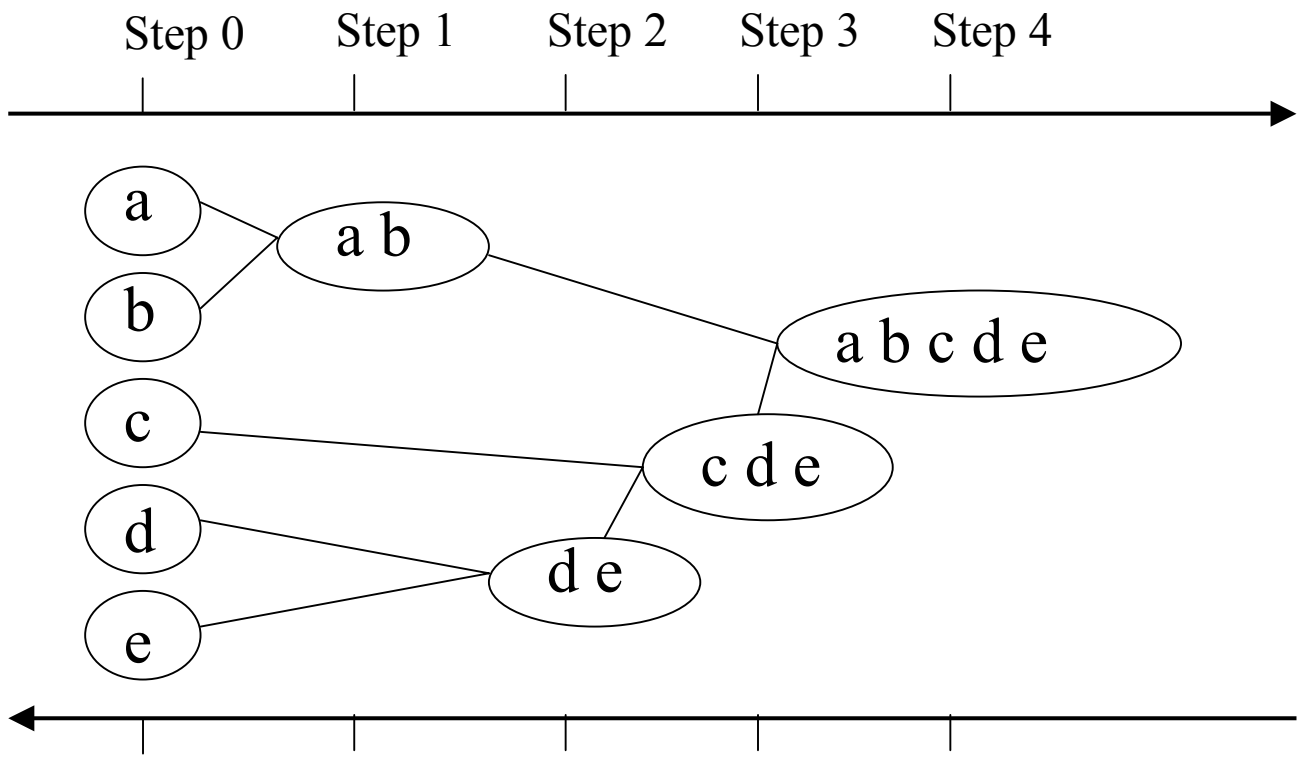
Méthodes hiérarchiques

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

Méthodes hiérarchiques

- Entrée : un échantillon de m enregistrements x1, …, xm
- 1. On commence avec m clusters (cluster = 1 enregistrement)
- 2. Grouper les deux clusters les plus « proches ».
- 3. S'arrêter lorsque tous les enregistrements sont membres d'un seul groupe
- 4. Aller en 2.

Arbre de clusters : Exemple



Arbre de clusters

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

- La hiérarchie de clusters est représentée comme un **arbre de clusters**, appelé **dendrogramme**
  - Les feuilles de l'arbre représentent les objets
  - Les noeuds intermédiaires de l'arbre représentent les clusters

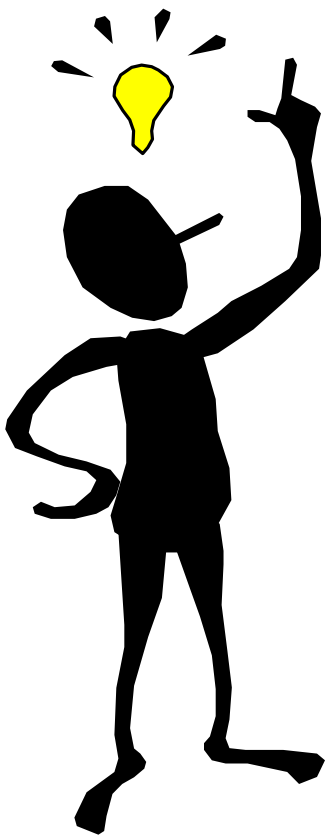
Distance entre clusters

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

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

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

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** à examiner est **réduit**

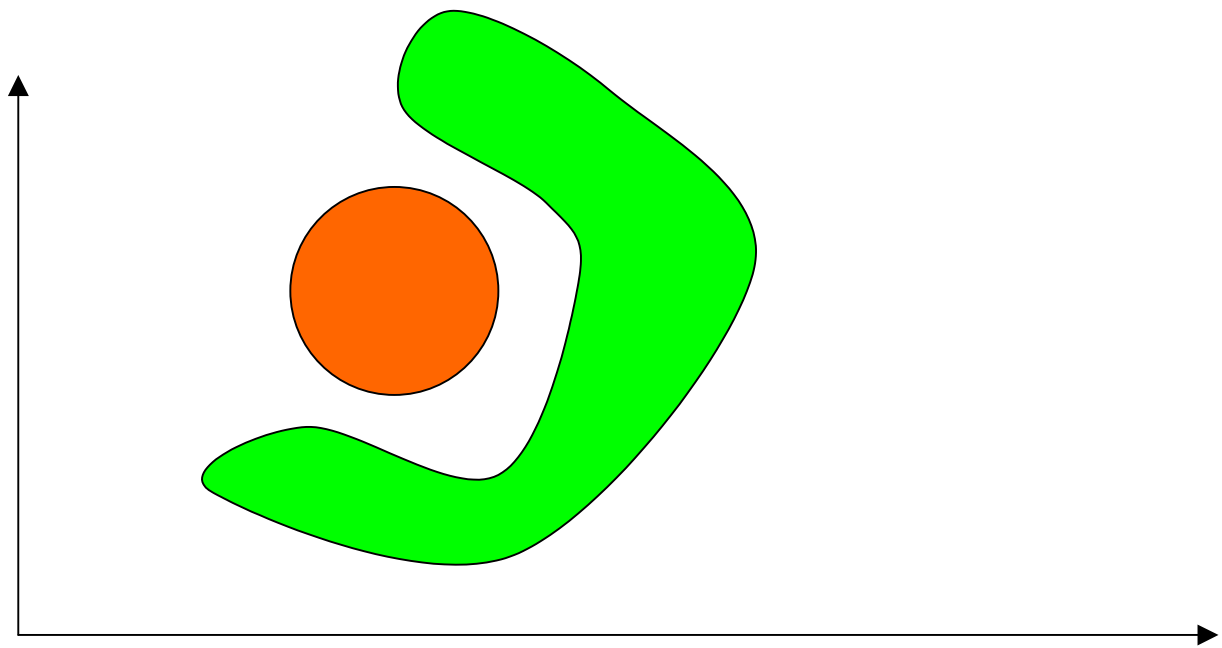
Méthodes hiérarchiques : Inconvénients

- **Groupement** de clusters est **définitif** => décisions erronnées sont **impossibles à modifier** ultérieurement

- Méthodes **non extensibles** pour des ensembles de données de grandes tailles

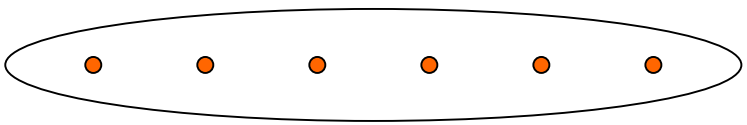
Méthodes basées sur la densité

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

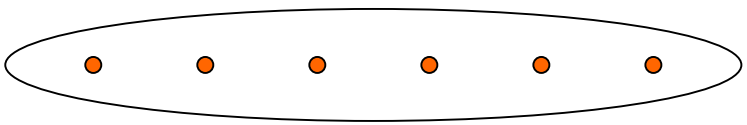


Méthodes basées sur la densité

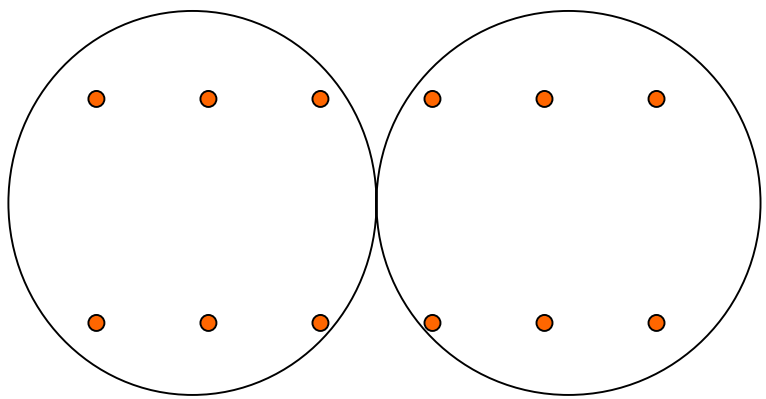
- Minimiser la distance inter-clusters n'est pas toujours un bon critère pour reconnaître des «formes » (applications géographiques, reconnaissance de formes – tumeurs, …).



Dist=18



Dist=15.3



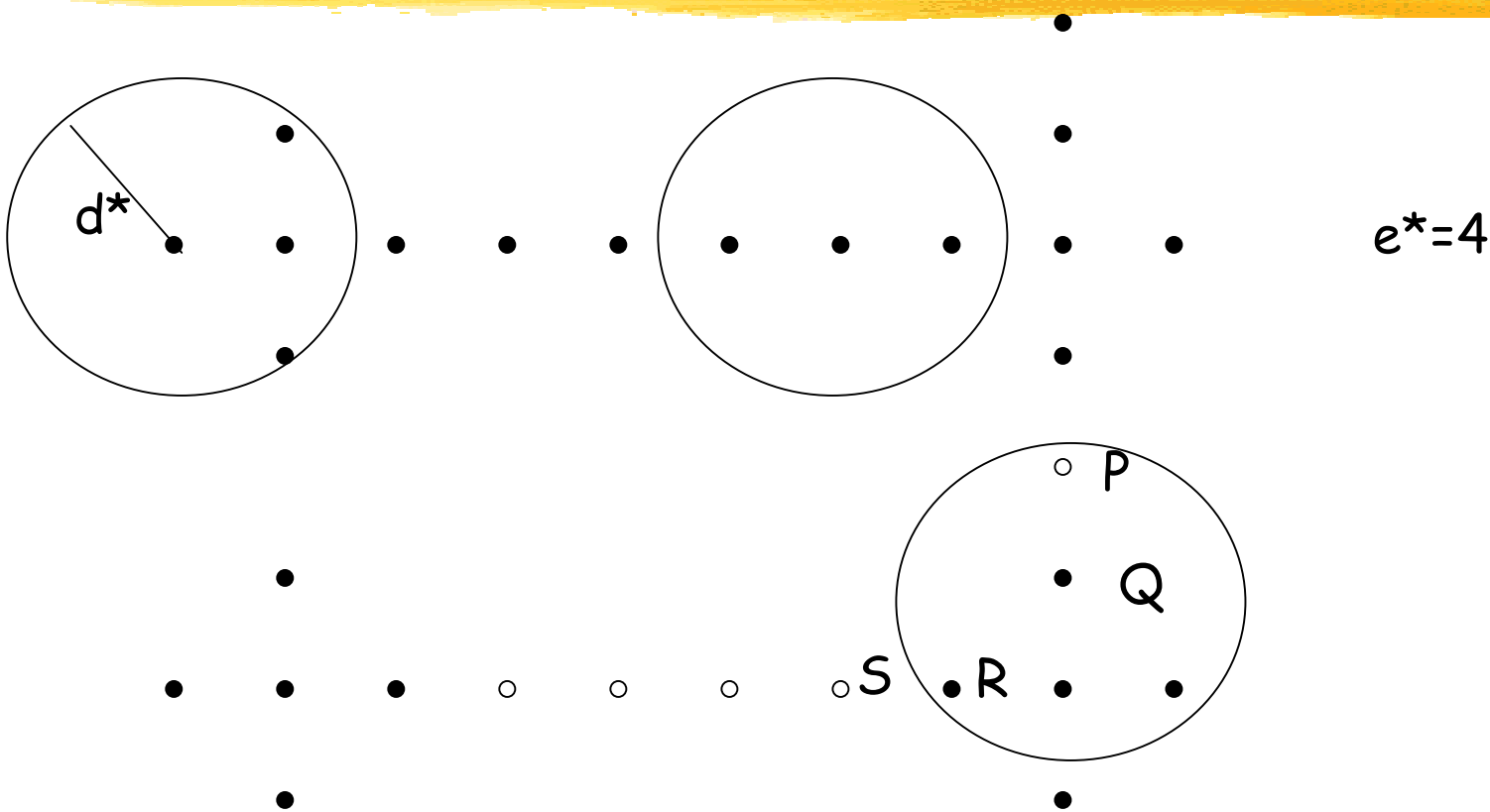
Méthodes basées sur la densité (1)

- Soit d* un nombre réel positif
- Si d(P,Q)<=d*, Alors P et Q appartiennent au même cluster
- Si P et Q appartiennent au même cluster, et d(Q,R)<=d*, Alors P et R appartiennent au même cluster

Méthodes basées sur la densité (2)

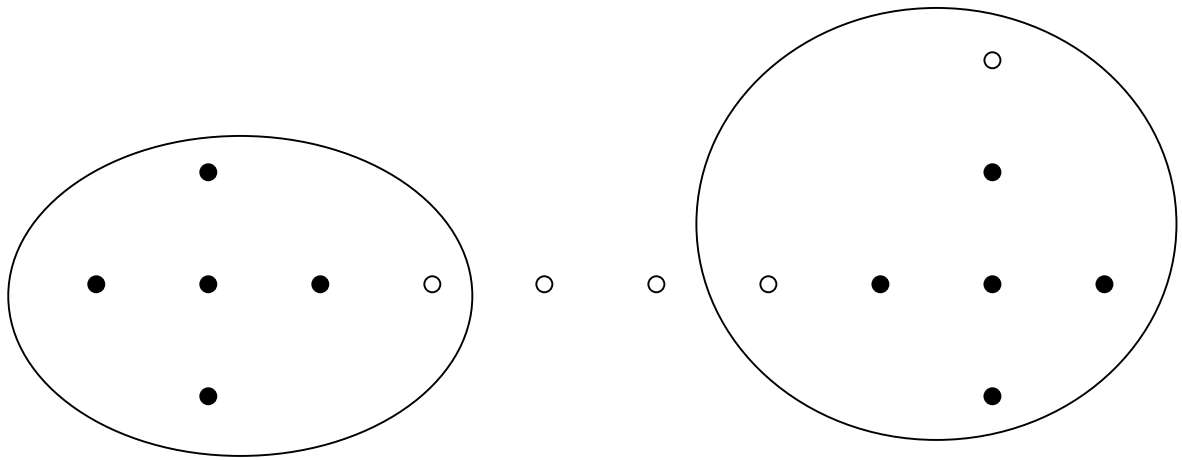
- Soit e* un nombre réel positif
- Un point P est dense ssi |{Q/d(P,Q)<=d*}|>=e*
- Si P et Q appartiennent au même cluster, et d(Q,R)<=d* et Q est dense, Alors P et R appartiennent au même cluster
- Les points non-denses sont appelés points de « bordure ».
- Les points en dehors des clusters sont appelés « bruits ».

Méthodes basées sur la densité



- Points noirs sont denses ; les autres ne sont pas denses
- Pour montrer que P et S appartiennent au même cluster, il suffit de montrer que P et R appartiennent au même cluster. Pour le montrer pour P et R, il suffit de le montrer pour P et Q …

Méthodes basées sur la densité



- Deux clusters sont trouvés
- Deux points sont des « bruits »
- Trois points sont des « bordures »

Etude de cas réel : **Génomique**

Sélection d'attributs + Clustering

LIFL : Equipe OPAC
I.B.L

USTL
LIFL Laboratoire d'Informatique Fondamentale de Lille
CNRS
OPAC

Le contexte

- Génopole de Lille : Aspect génétique des maladies multifactorielles
- Collaboration avec l'I.B.L. (Institut de Biologie de Lille) laboratoire des maladies multifactorielles (UPRES-A 8090) : diabète, obésité
- Génération de gros volumes de données : outil d'aide à l'interprétation des résultats

Etudes de l'IBL

- Etudes de type familial (parents, enfants) – Prélévement d'ADN
- Analyse de liaison : co-transmission d'un gène
- Comparaison de gènes entre paires d'individus d'une même famille

Objectif :

Localiser un ou plusieurs gènes de prédisposition pour la maladie

Problème posé

- Très grand nombre de données générées
  - (~ 1 000 points de comparaison, 200 familles)

- Méthodes statistiques limitées pour étudier la corrélation entre gènes

Besoin d'un outil d'extraction de connaissances : Data Mining

Contexte

Hypothèses de travail :

- un cas particulier de Data Mining
- les données fournies par l'IBL contiennent de nombreux attributs
- existence de données manquantes ou incertaines
- contexte d 'apprentissage non supervisé

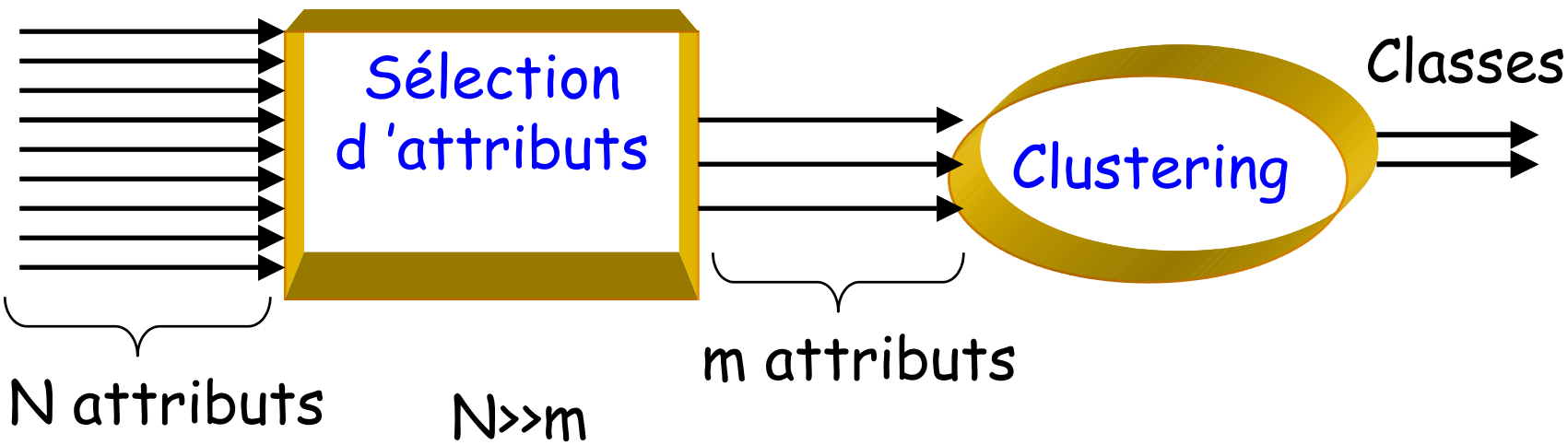
Objectif :

- connaître les classes d 'attributs provoquant la maladie
- connaître les corrélations entre les attributs

Méthodologie adoptée

Réalisation :

- d'une sélection d 'attributs : Réduire le nombre d 'attributs pour améliorer la classification
- d'un clustering

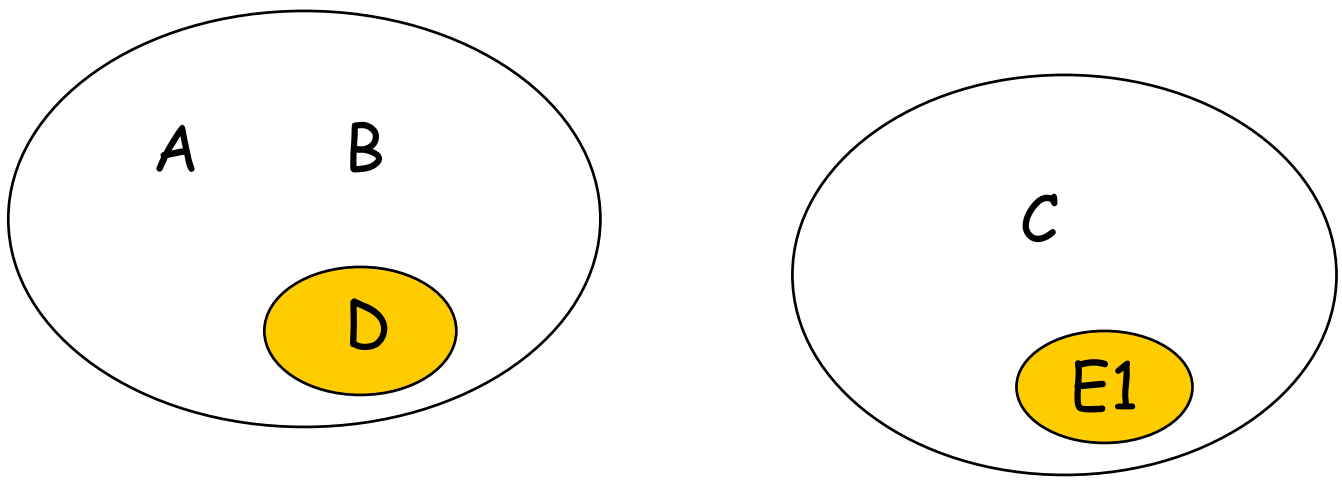


K-moyennes

- Sans sélection d 'attributs :
  - 400 attributs pour 200 objets,
  - temps de calcul > 7500 min. (>125 h.),
  - résultats inexploitables
- Avec sélection d 'attributs :
  - une dizaine d 'attributs pour 200 objets,
  - temps de calcul entre 3 minutes et 15 minutes,
  - résultats exploitables.

Workshop GAW11 de 1998

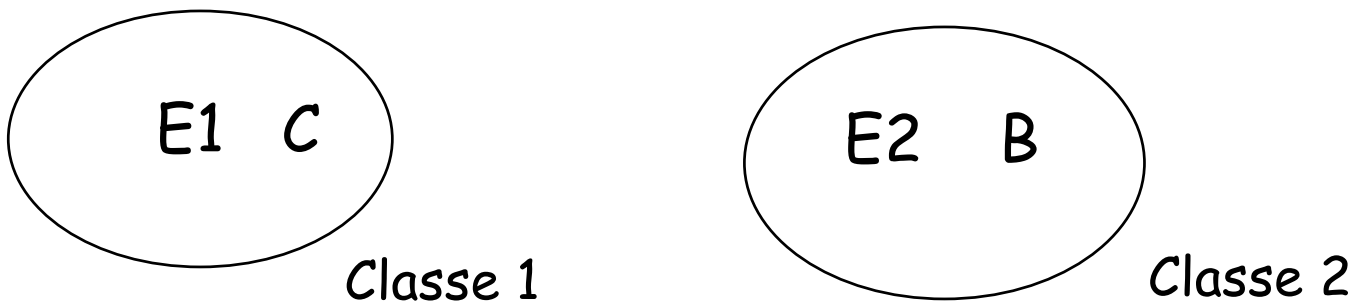
- Données simulées dont on connaît les résultats
- Résultats à trouver :



Résultats

Résultats obtenus sur le workshop GAW11 de 1998

- Exemple d 'ensembles d 'attributs sélectionnés (Support trouvé > 0.65) :
- 81 85, 402 407, 224 229 (Locus C) , 308 313, 190 195, 374 379 (Locus B)
- Exemple de clustering



Conclusion

- Bilan
  - Compréhension et modélisation d 'un problème complexe
  - Sélection d 'attributs : sélection de locus impliqués dans la maladie
  - Clustering : les ensembles finaux sont trouvés lorsqu 'il y a peu d 'erreurs dans le choix des attributs sélectionnés

Clustering – Résumé (1)

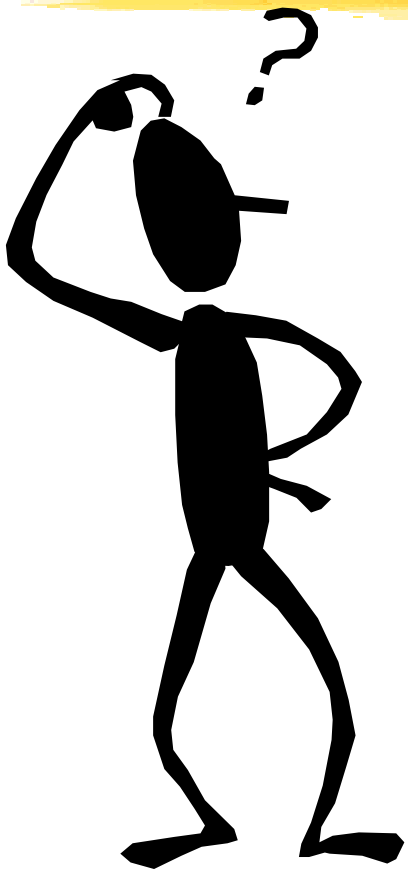
- Le clustering groupe des objets en se basant sur leurs similarités.
- Le clustering possède plusieurs applications.
- La mesure de similarité peut être calculée pour différents types de données.
- La sélection de la mesure de similarité dépend des données utilisées et le type de similarité recherchée.

Clustering – Résumé (2)

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

Références

- M. R. Anderberg. **Cluster Analysis for Applications**. Academic Press, 1973.
- P. Arabie, L. J. Hubert, and G. De Soete. **Clustering and Classification**. World Scientific, 1996
- A. K. Jain and R. C. Dubes. **Algorithms for Clustering Data**. Prentice Hall, 1988
- L. Kaufman and P. J. Rousseeuw. **Finding Groups in Data: an Introduction to Cluster Analysis**. John Wiley & Sons, 1990.

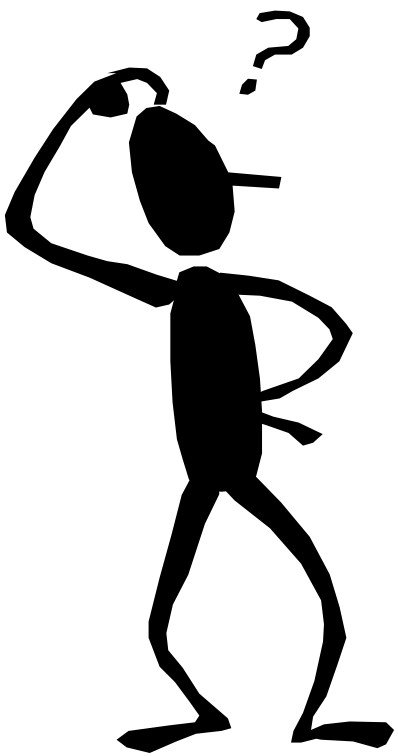


Classification

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

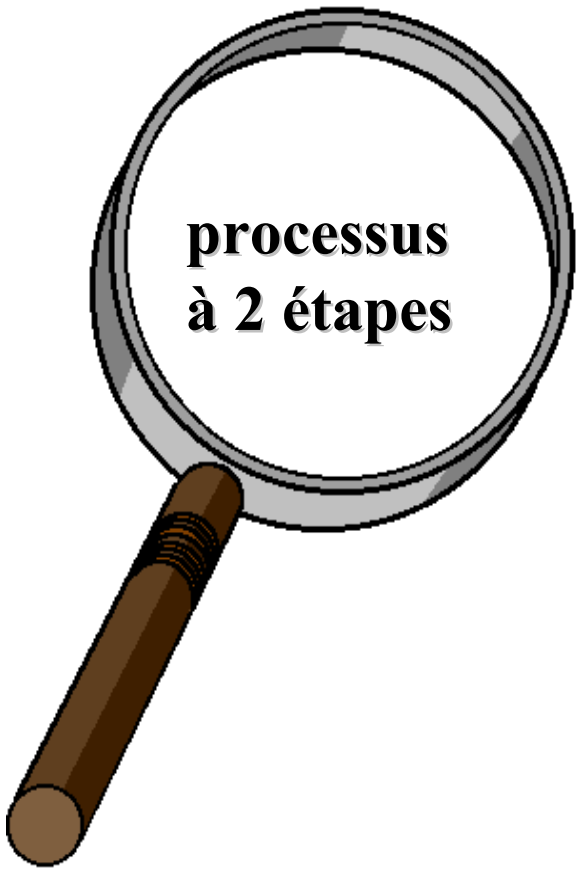


- Elle permet de **prédire** si un élément est membre d'un groupe ou d 'une catégorie donné.
- Classes
  - Identification de groupes avec des profils particuliers
  - Possibilité de décider de l'appartenance d'une entité à une classe
- Caractéristiques
  - Apprentissage supervisé : classes connues à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

Classification - Applications

- Accord de crédit
- Marketing ciblé
- Diagnostic médical
- Analyse de l'effet d'un traitement
- Détection de fraudes fiscales
- etc.

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 prédéfinie
- 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, ...

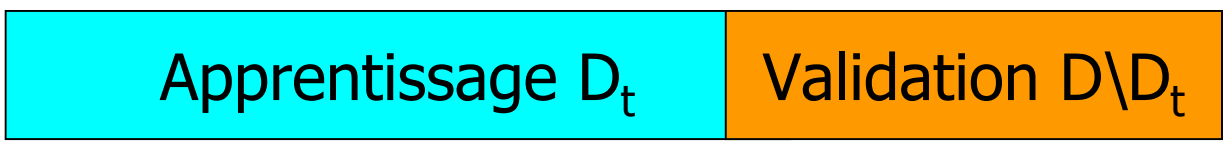
Utilisation du modèle

- **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)

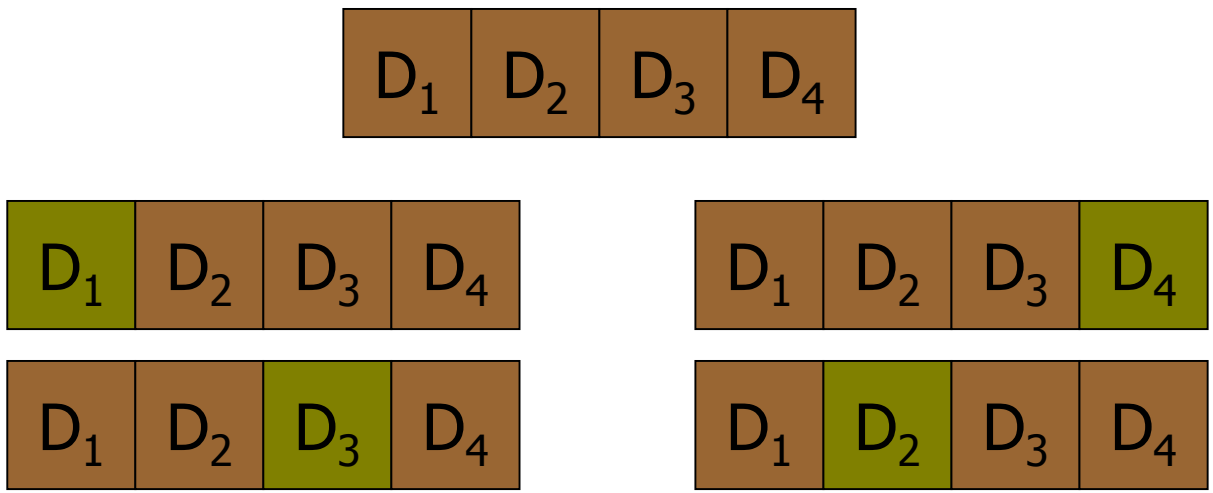
Estimation des taux d'erreurs :

- **Partitionnement** : apprentissage et test (ensemble de données important)
  - Utiliser 2 ensembles indépendents, e.g., ensemble d'apprentissage (2/3), ensemble test (1/3)



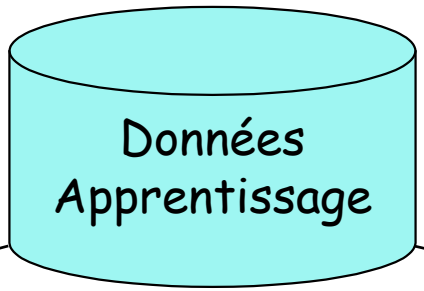
Validation de la Classification (accuracy)

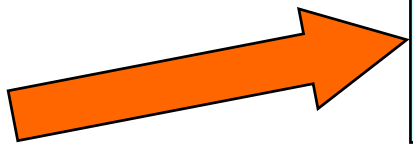
- **Validation croisée** (ensemble de données modéré)
  - Diviser les données en *k* sous-ensembles
  - Utiliser *k-1* sous-ensembles comme données d'apprentissage et un sous-ensemble comme données test



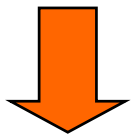
- **Bootstrapping** : n instances test aléatoires (ensemble de données réduit)

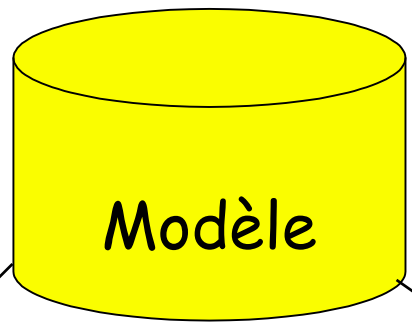
Exemple : Construction du modèle





Algorithmes
Classification

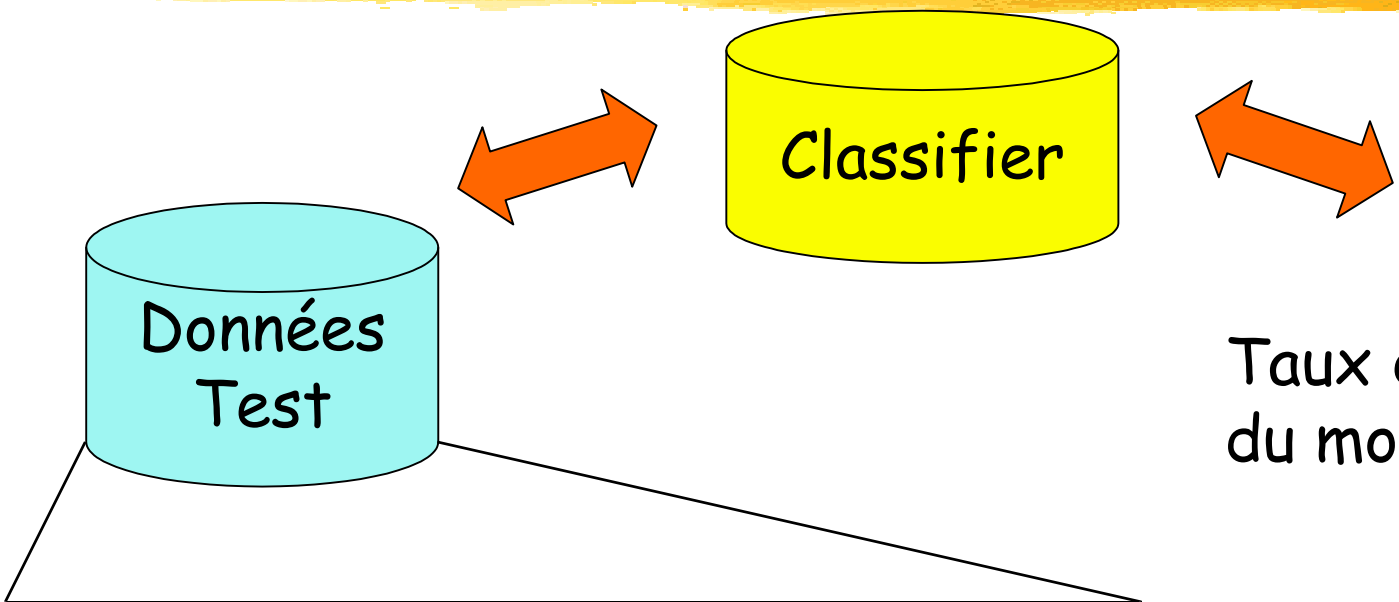




Nom	Rang	Année	Titulaire
Mary	Assistant Prof	3	non
James	Assistant Prof	7	oui
Bill	Professor	2	oui
John	Associate Prof	7	oui
Mark	Assistant Prof	6	non
Annie	Associate Prof	3	non

Si Rang = 'Professor'
Ou Année > 6
Alors Titulaire = Oui

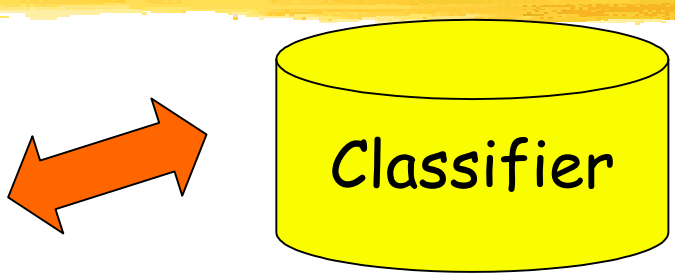
Exemple : Utilisation du modèle

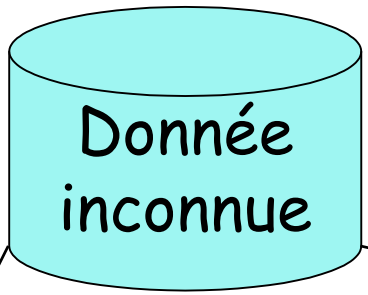


Taux d'erreur
du modèle ?

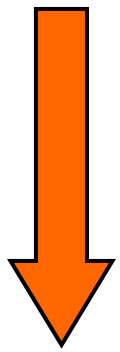
Nom	Rang	Année	Titulaire
Tom	Assistant Prof	2	non
Lisa	Associate Prof	7	non
Jack	Professor	5	oui
Ann	Assistant Prof	7	oui

Exemple : Utilisation du modèle





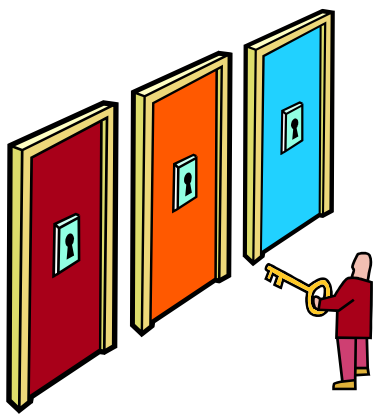
Titulaire ?



Nom	Rang	Année	Titulaire
Jeff	Professor	4	?
Paul	Associate Prof	7	?

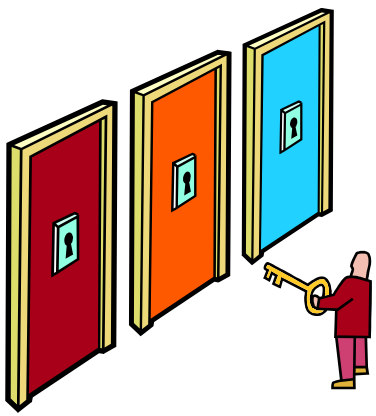
Oui
Oui

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 neighbor).
- 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)

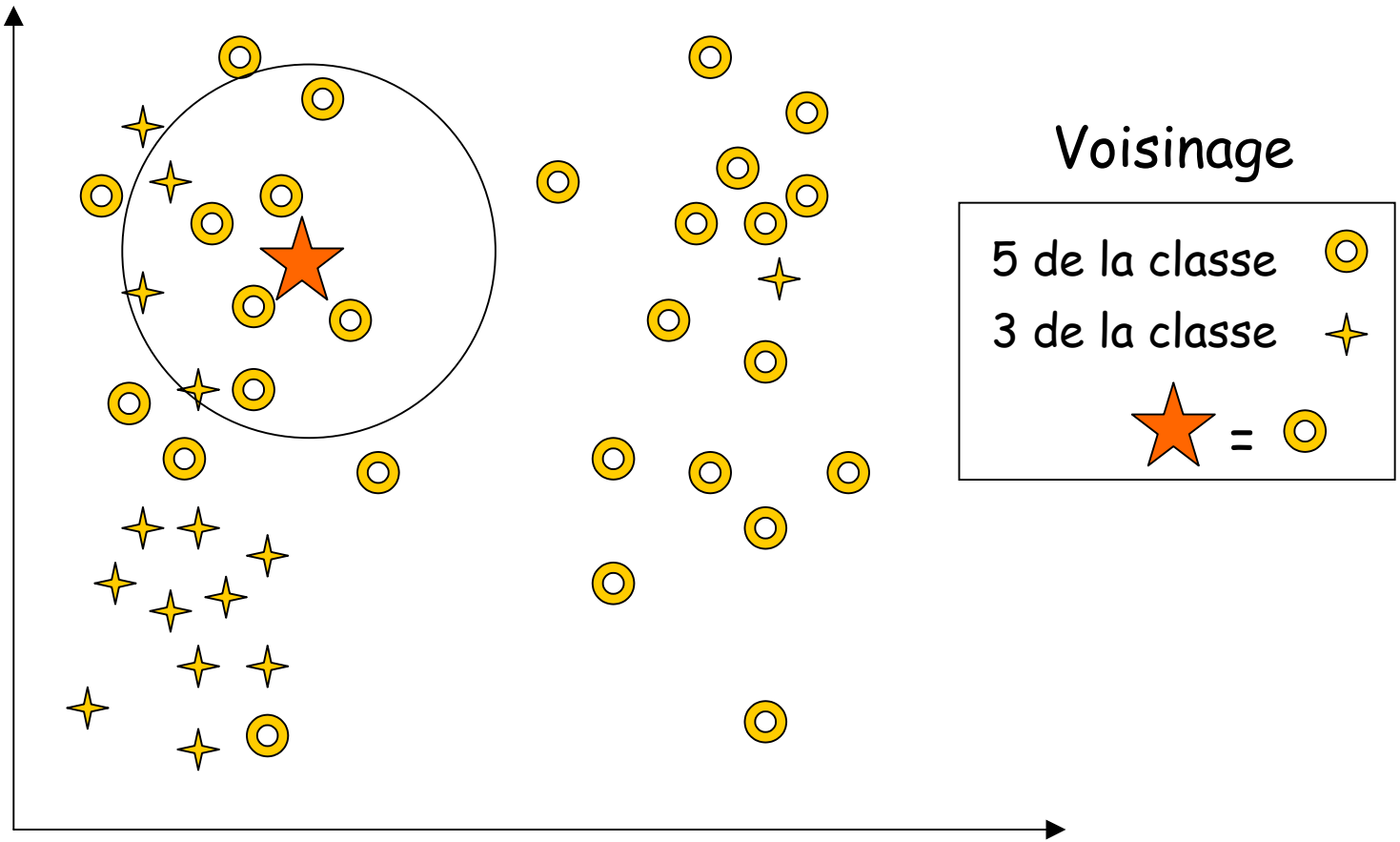
- Objectif : affecter une classe à une nouvelle instance
- donnée : un échantillon de m enregistrements classés (x, c(x))
- entrée : un enregistrement y
  - 1. Déterminer les k plus proches enregistrements de y
  - 2. combiner les classes de ces k exemples en une classe c
- sortie : la classe de y est 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 = nombre d 'attributs + 1
  - Vote majoritaire : prendre la classe majoritaire.
  - Vote majoritaire pondéré : chaque classe est pondérée. Le poids de c(xi) est inversement proportionnel à la distance d(y,xi).

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

Illustration



Algorithme kNN : critique

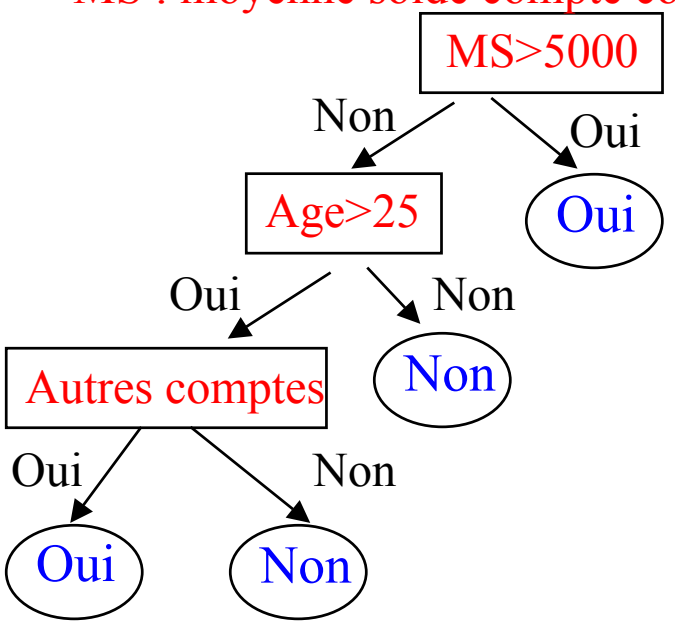
- Pas d'apprentissage : introduction de nouvelles données ne nécessite pas la reconstruction du modèle.
- Clarté des résultats
- Tout type de données
- Nombre d'attributs
- Temps de classification : -
- Stocker le modèle : -
- Distance et nombre de voisins : dépend de la distance, du nombre de voisins 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

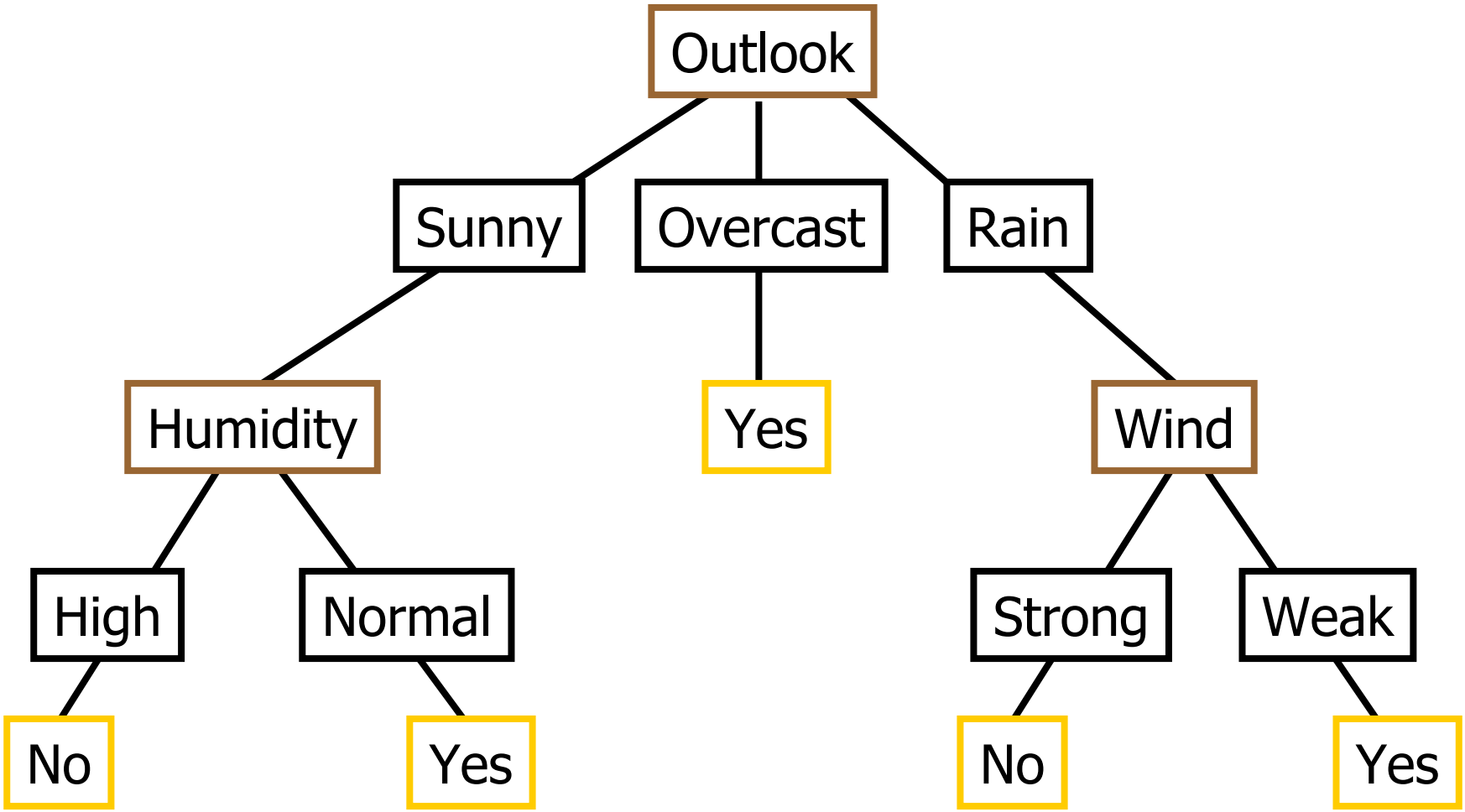
Arbre de décision - Exemple

Ensemble d'apprentissage

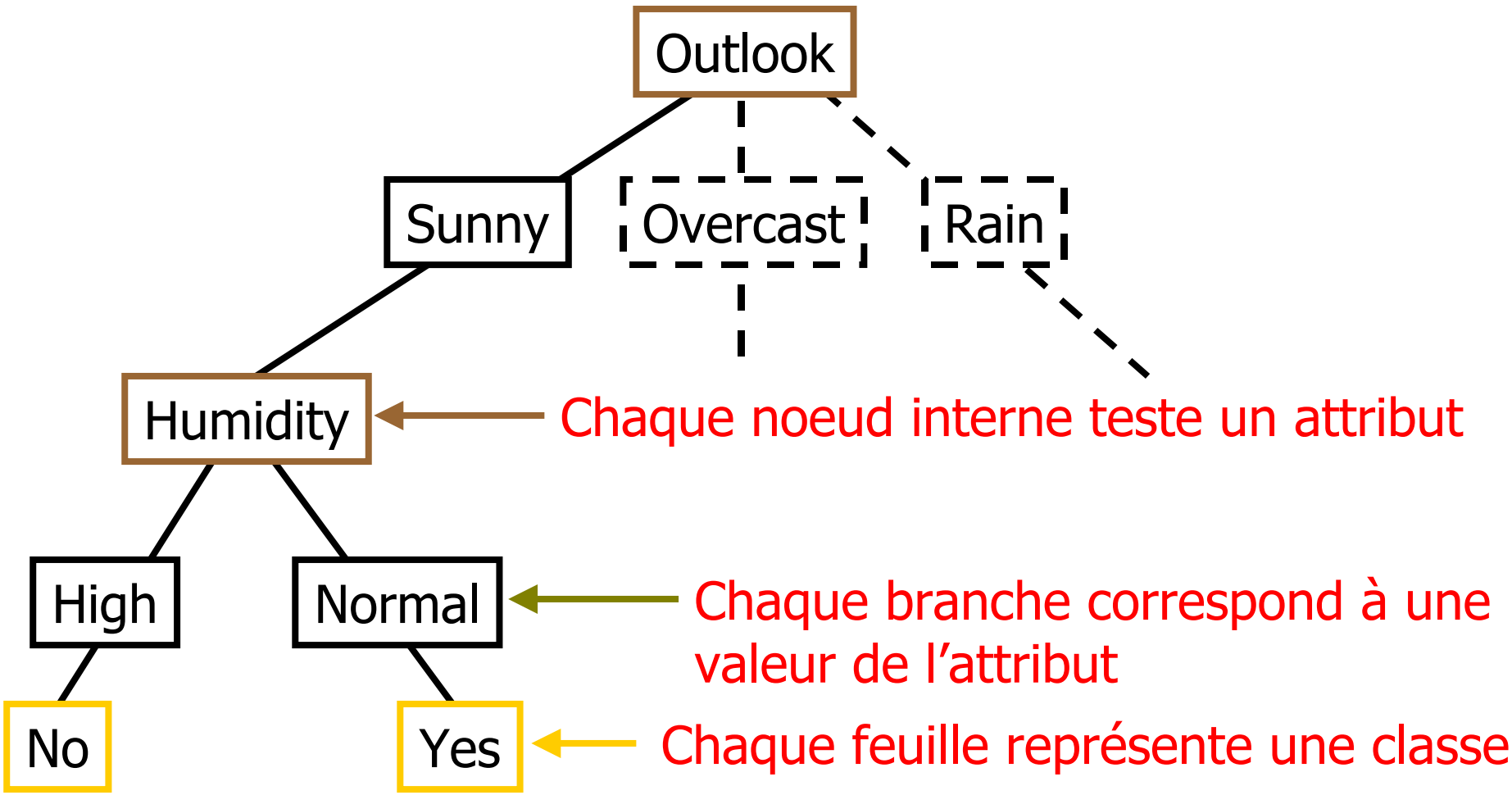
Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal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

Jouer au tennis ?

Arbre de décision - Exemple



Exemple - Jouer au tennis ?

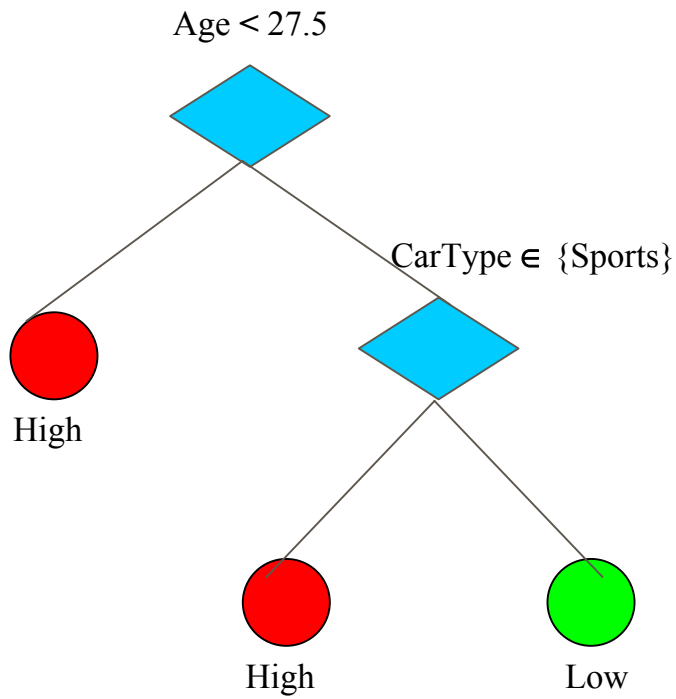


Arbres de décision - Exemple

Risque - Assurances

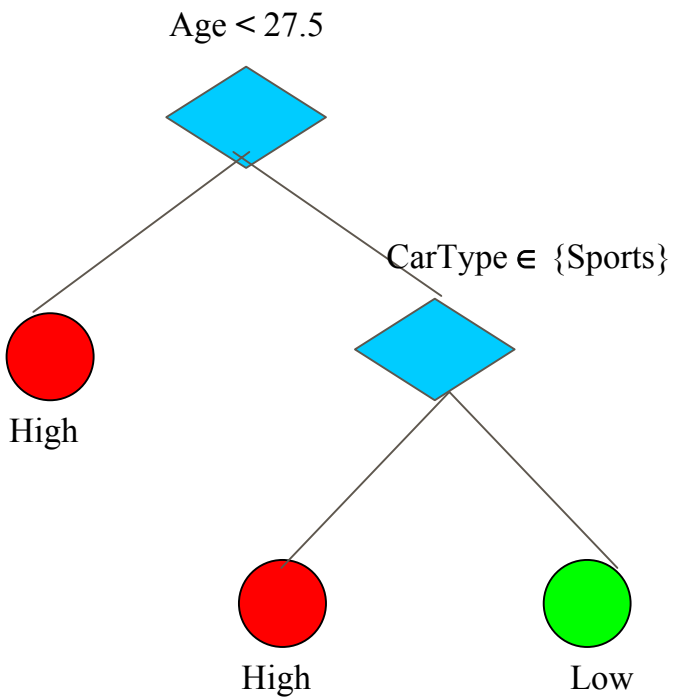
Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

Numérique Enumératif



Age=40, CarType=Family ⟹ Class=Low

Des arbres de décision aux règles



1) Age < 27.5 ⇒ High

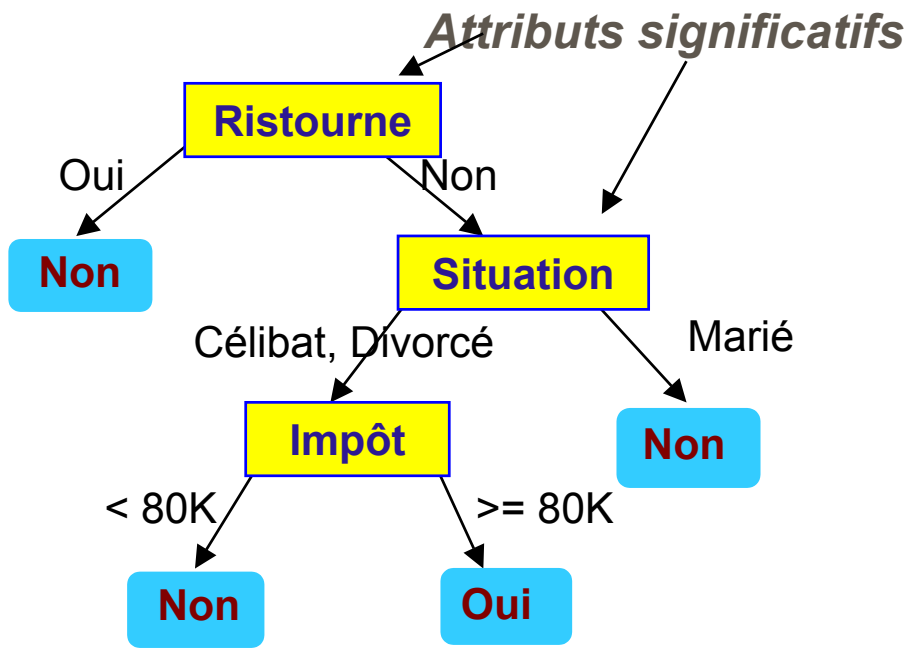
2) Age >= 27.5 and
CarType = Sports ⇒ High

3) Age >= 27.5 and
CarType ≠ Sports ⇒ High

Arbres de décision - Exemple
Détection de fraudes fiscales

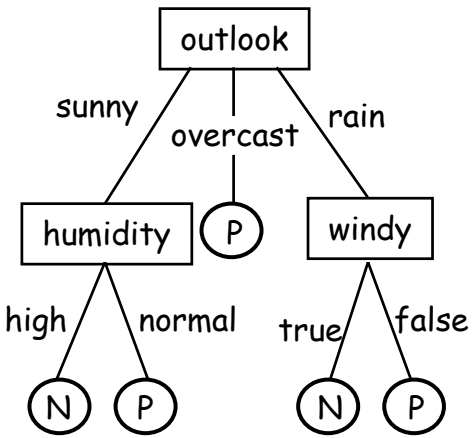
énumératif énumératif numérique classe

Id	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



- **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

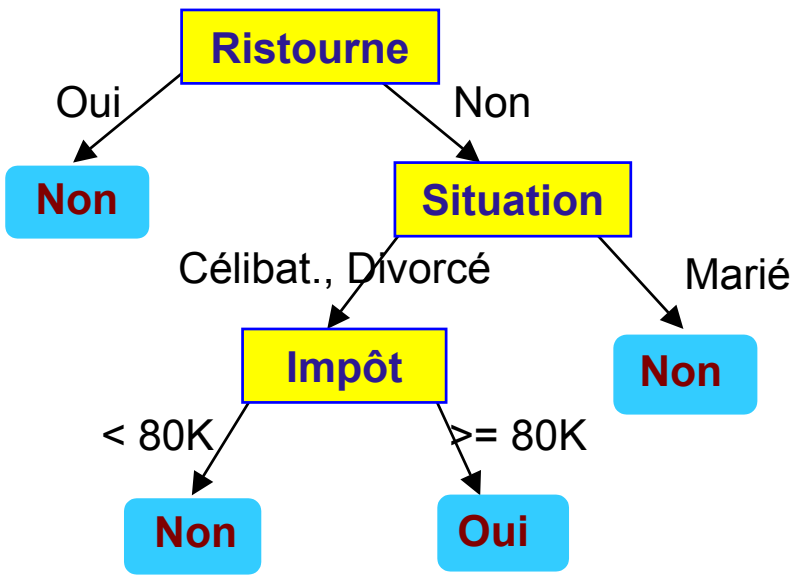


Si outlook=sunny
Et humidity=normal
Alors play tennis

- 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 noeud terminal représente la classe prédite
- Les règles sont généralement plus faciles à comprendre que les arbres

Des arbres de décision aux règles

Arbre de décision = Système de règles exhaustives et mutuellement exclusives



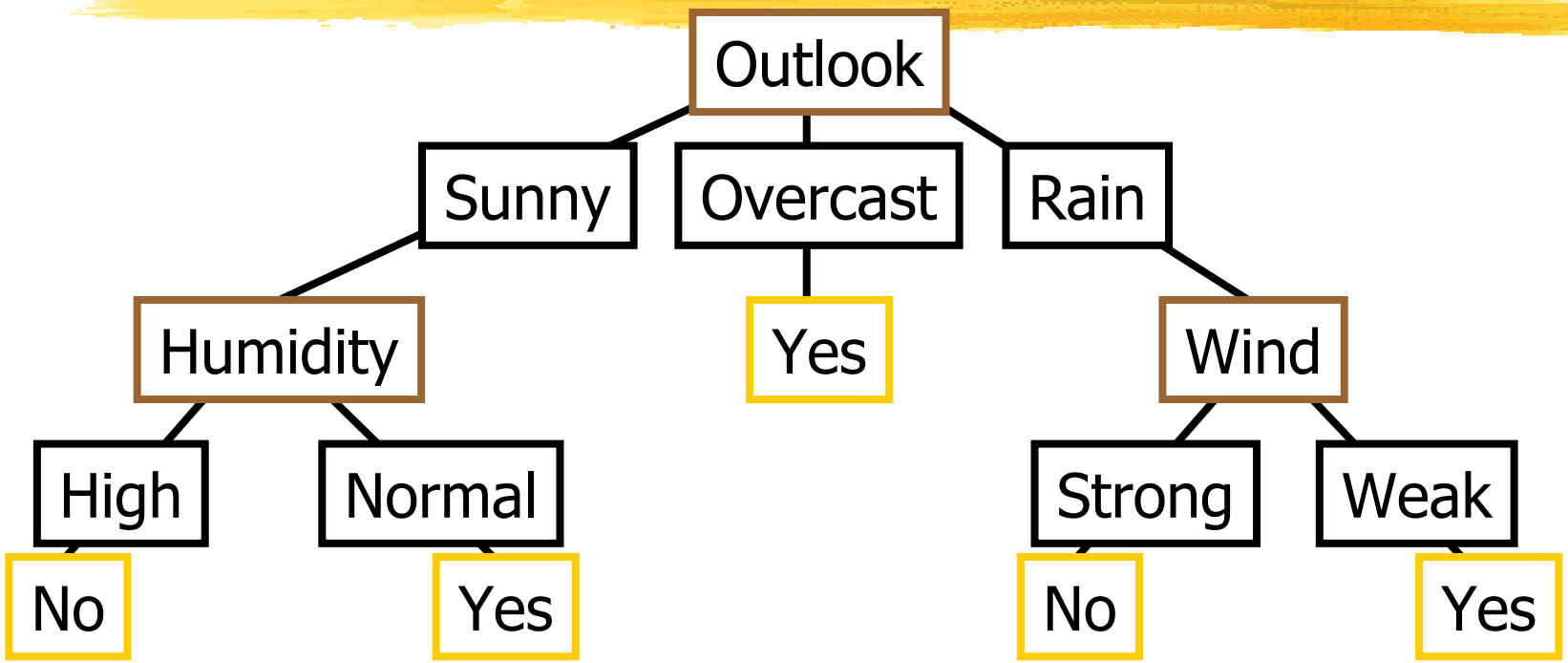
1) Ristourne = Oui ⇒ Non

2) Ristourne = Non et
Situation in {Célibat., Divorcé}
et Impôt < 80K ⇒ Non

3) Ristourne = Non et
Situation in {Célibat., Divorcé}
et Impôt >= 80K ⇒ Oui

4) Ristourne = Non et
Situation in {Marié} ⇒ Non

Des arbres de décision aux règles



R_1: If (Outlook=Sunny) $\wedge$ (Humidity=High) Then PlayTennis=No
R_2: If (Outlook=Sunny) $\wedge$ (Humidity=Normal) Then PlayTennis=Yes
R_3: If (Outlook=Overcast) Then PlayTennis=Yes
R_4: If (Outlook=Rain) $\wedge$ (Wind=Strong) Then PlayTennis=No
R_5: If (Outlook=Rain) $\wedge$ (Wind=Weak) Then PlayTennis=Yes

Génération de l'arbre de décision

Deux phases dans la génération de l'arbre :

- **Construction de l'arbre**
  - Arbre peut atteindre une taille élevée
- **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

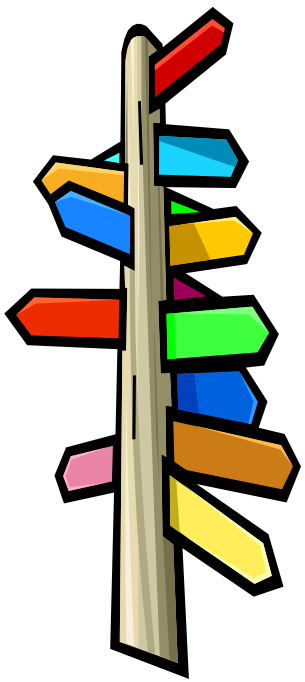
Algorithmes de classification

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

Algorithmes de classification

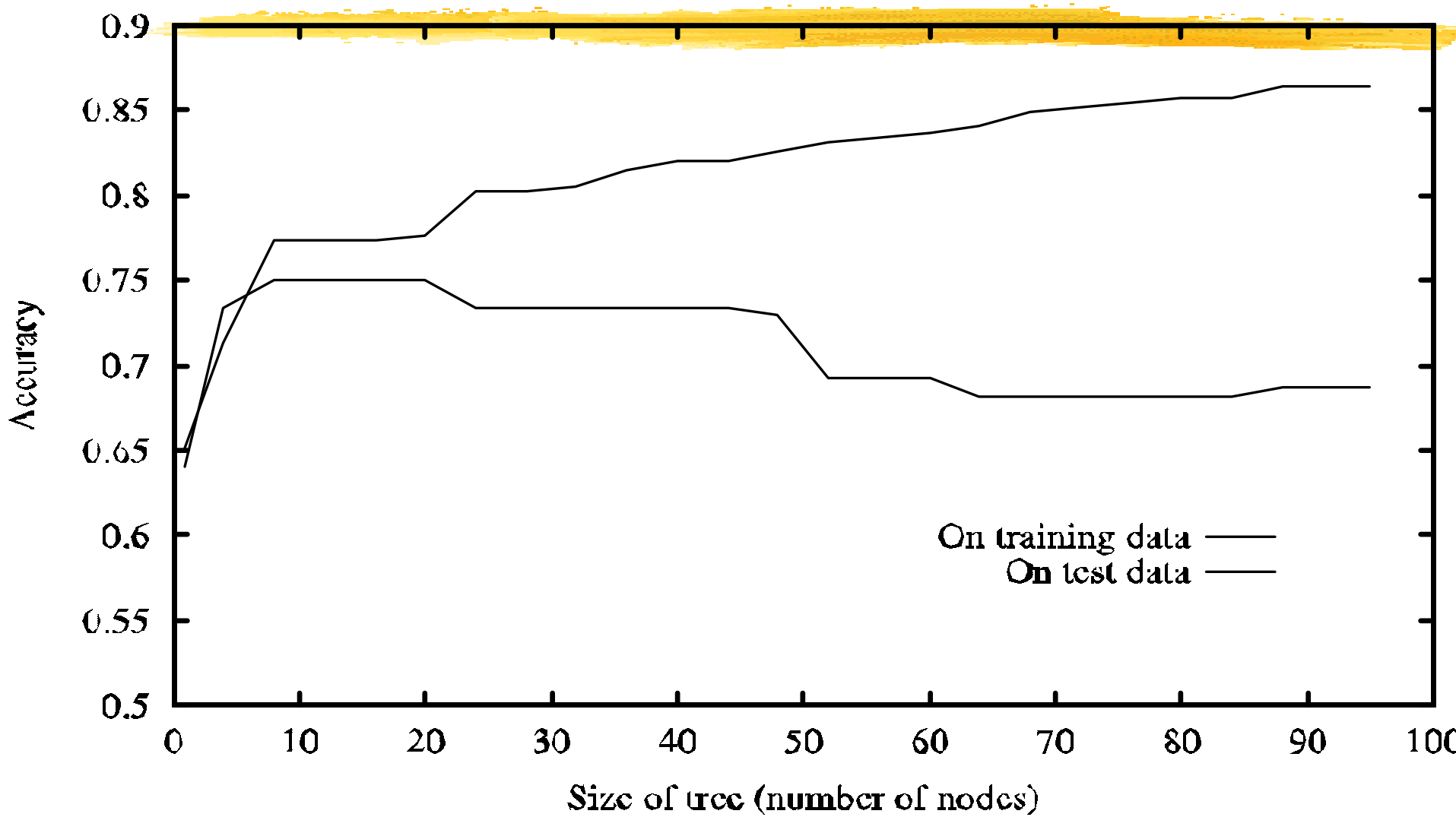
- Elaguer l'arbre obtenu (pruning)
  - Supprimer les sous-arbres qui n'améliorent pas l'erreur de la classification (accuracy) ➔ arbre ayant un meilleur pouvoir de généralisation, même si on augmente l'erreur sur l'ensemble d'apprentissage
  - Eviter le problème de sur-spécialisation (over-fitting),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

Sur-spécialisation - arbre de décision

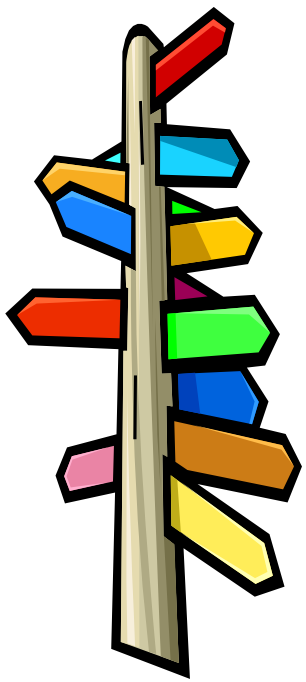


- **L'arbre généré peut sur-spécialiser l'ensemble d'apprentissage**
  - Plusieurs branches
  - Taux d'erreur important pour les instances inconnues
- **Raisons de la sur-spécialisation**
  - bruits et exceptions
  - Peu de donnée d'apprentissage
  - Maxima locaux dans la recherche gloutonne

Overfitting dans les arbres de décision



Comment éviter l'overfitting ?



- **Deux approches :**
- **Pré-élagage** : Arrêter de façon prématurée 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

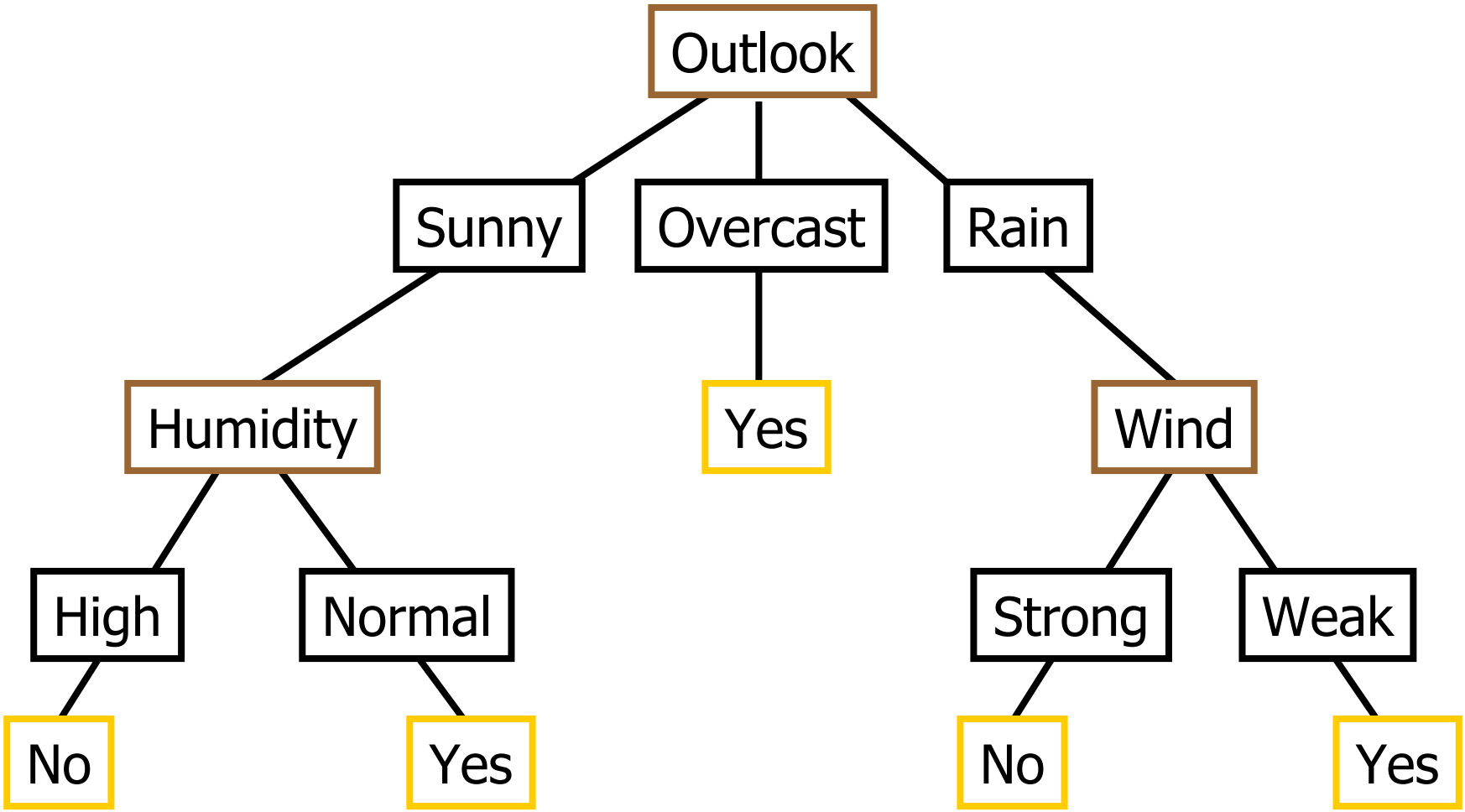
- Evaluation des différents branchements pour tous les attributs
- Sélection du "meilleur" branchement "et de l'attribut "gagnant"
- Partitionner les données entre les fils
- Construction en largeur (C4.5) ou en profondeur (SPLIT)
- Questions critiques :
  - Formulation des tests de branchement
  - Mesure de sélection des attributes

Exemple : Jouer au tennis ?

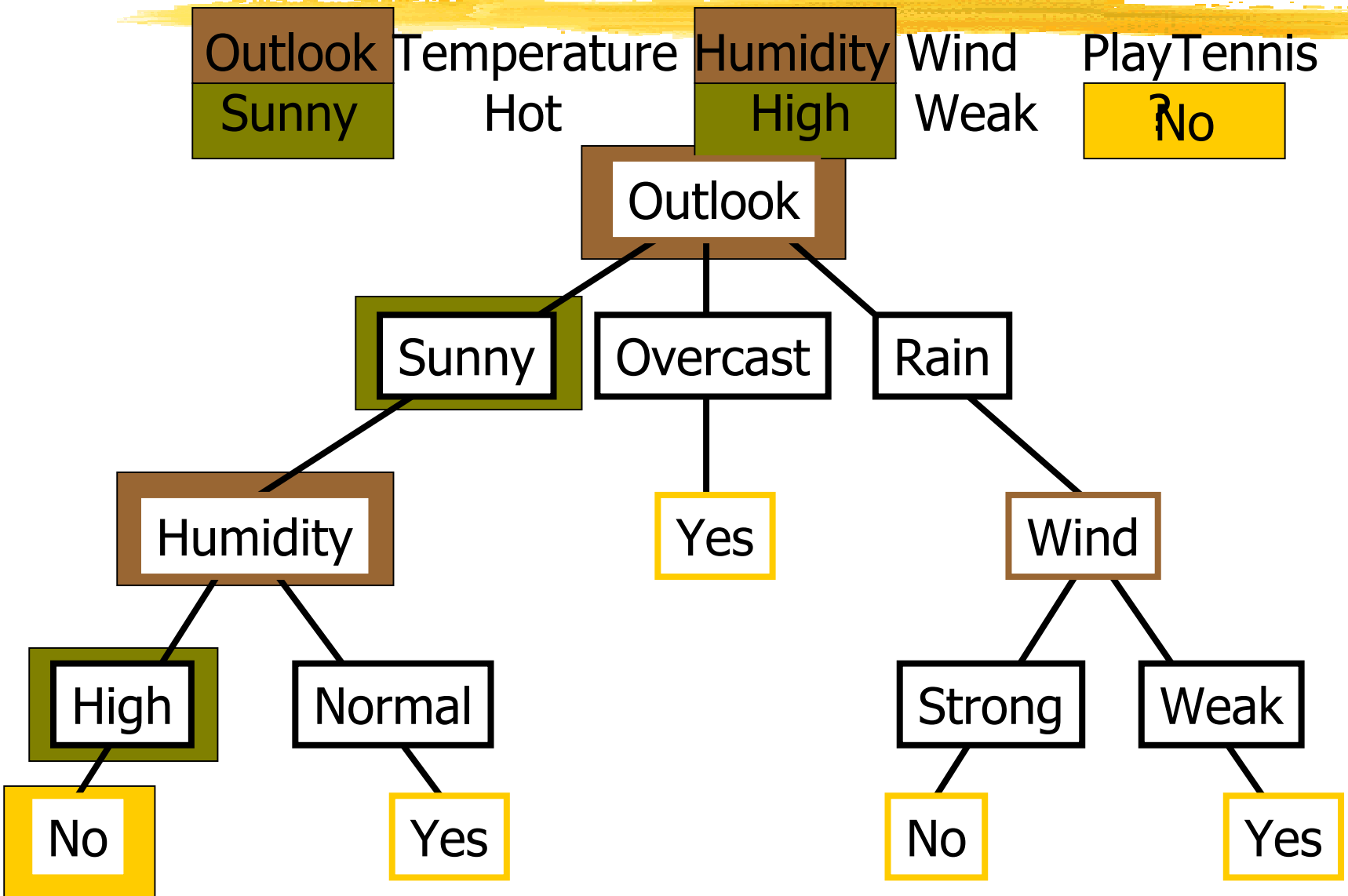
Ensemble d'apprentissage

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)

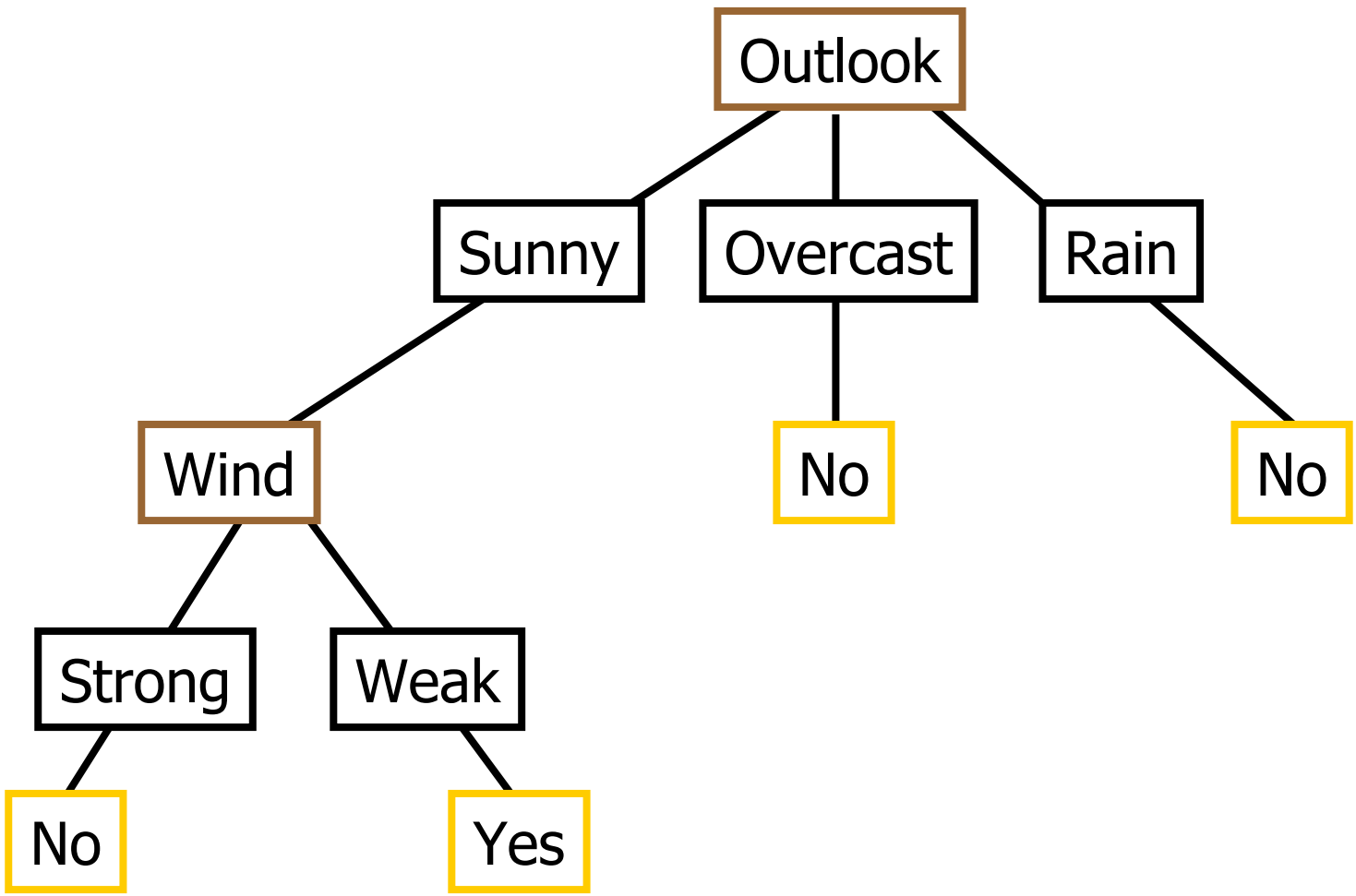


Arbre de décision obtenu avec ID3 (Quinlan 86)



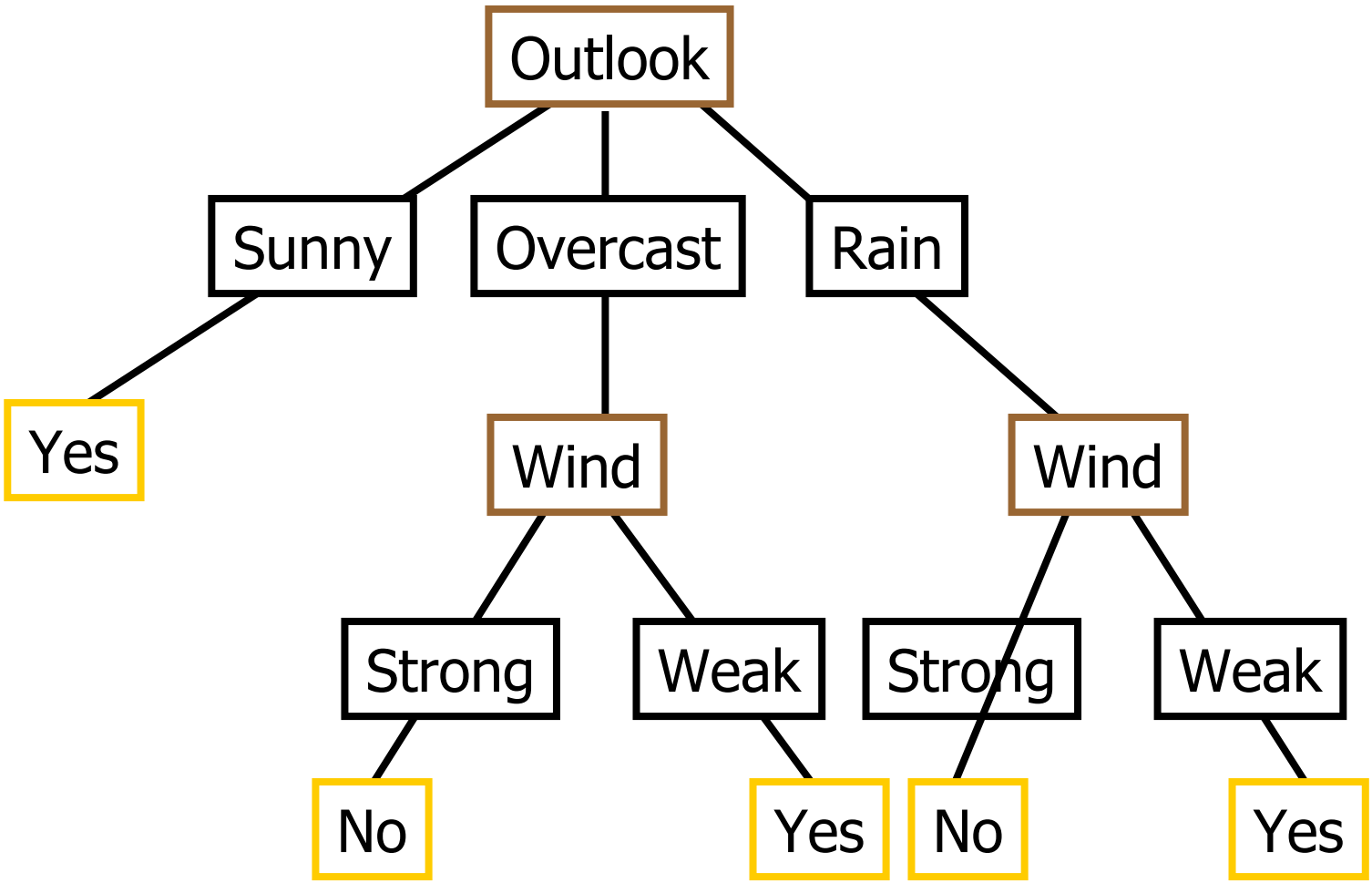
Arbre de décision et conjonction

Outlook=Sunny $\wedge$ Wind=Weak



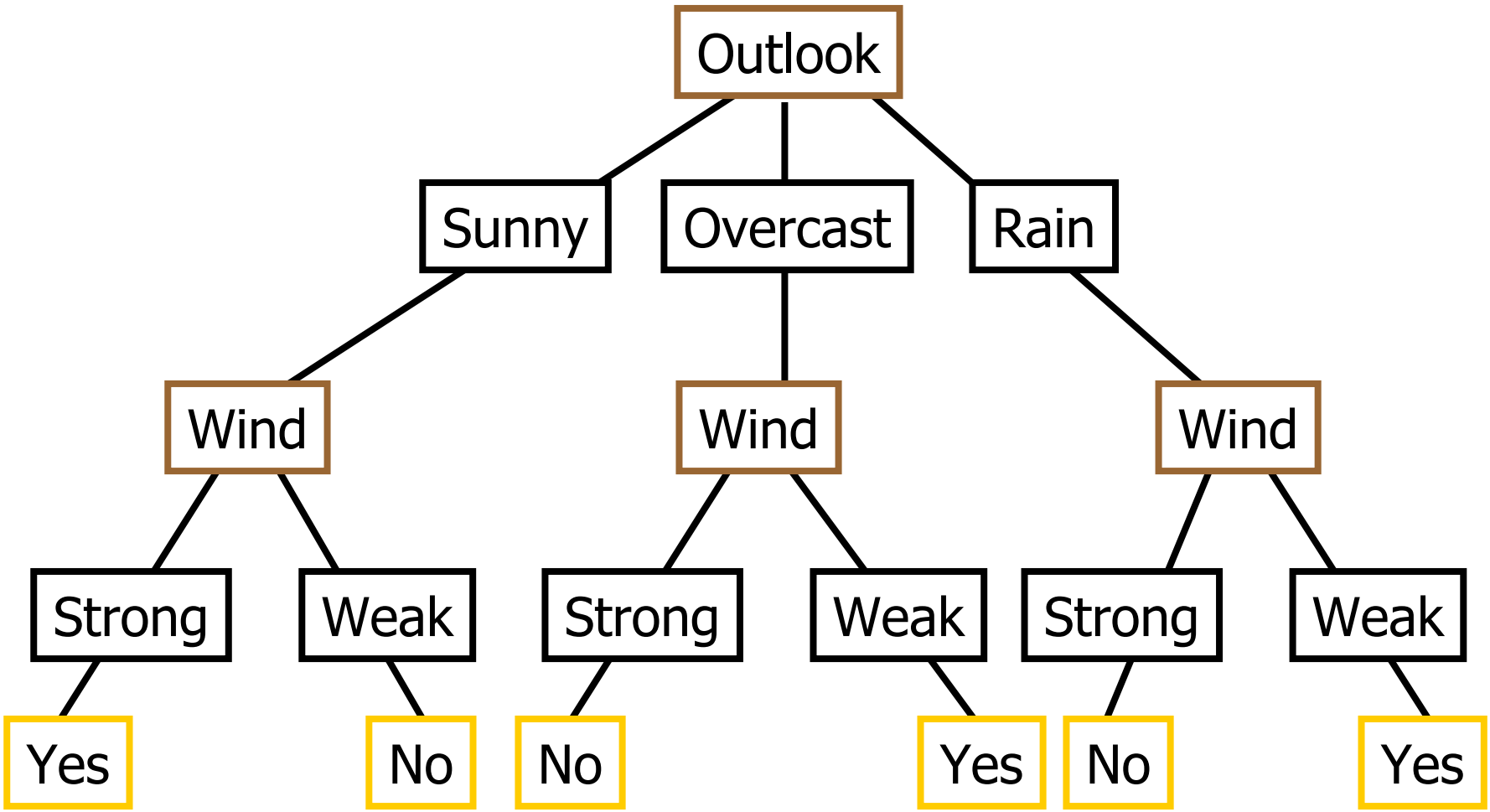
Arbre de décision et disjonction

Outlook=Sunny $\vee$ Wind=Weak



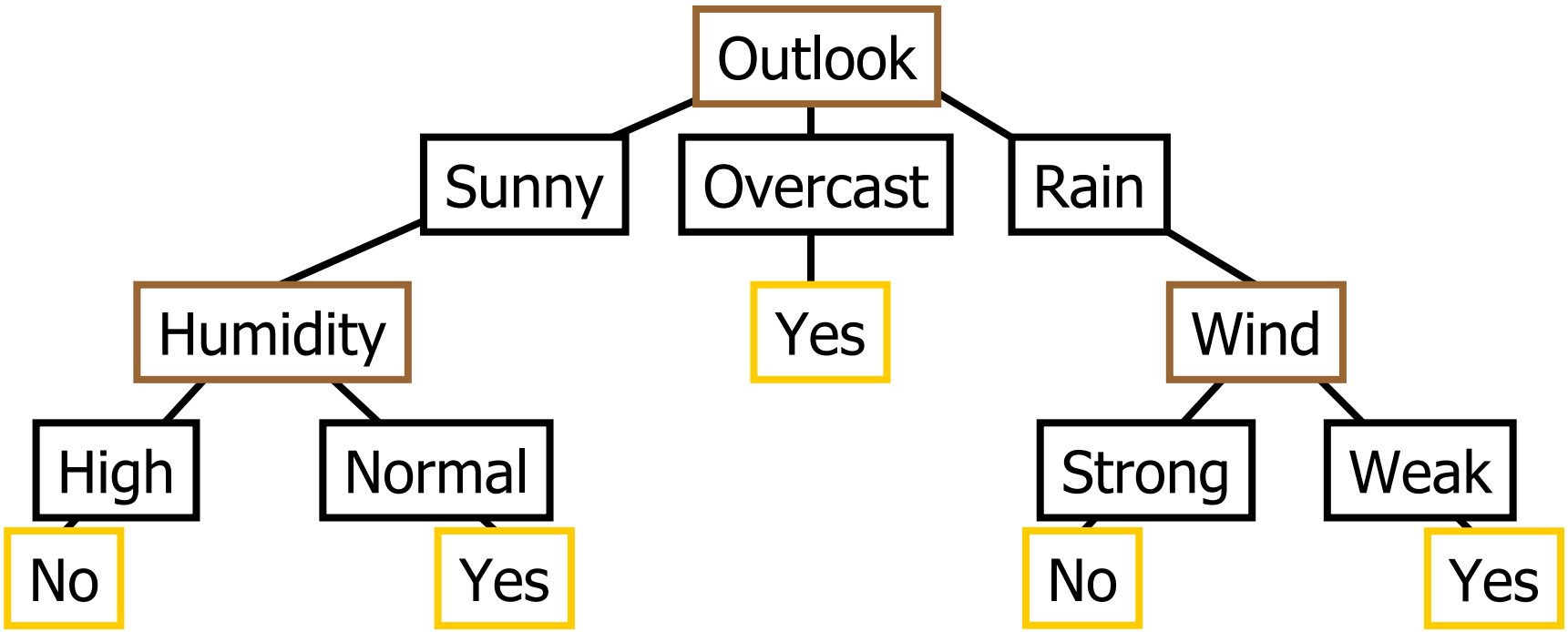
Arbre de décision et XOR

Outlook=Sunny XOR Wind=Weak



Arbre de décision et conjonction

- arbre de décision représente des disjonctions de conjonctions



(Outlook=Sunny $\wedge$ Humidity=Normal)
$\vee$ (Outlook=Overcast)
$\vee$ (Outlook=Rain $\wedge$ Wind=Weak)

Algorithmes pour les arbres de décision

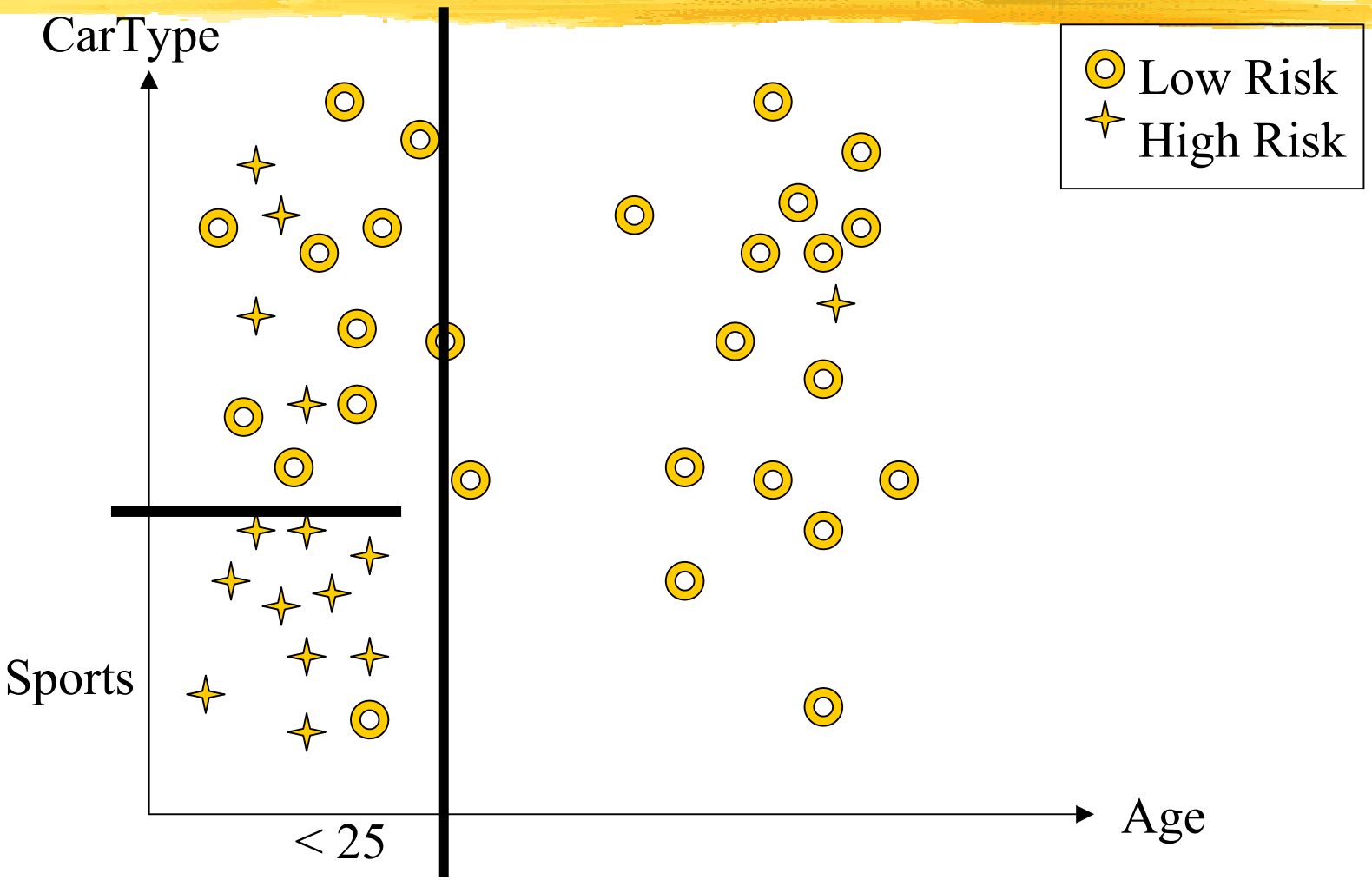
- **Algorithme de base**
  - Construction récursive d'un arbre de manière "diviser-pour-régner" descendante
  - Attributs considérés énumératifs
  - Glouton (piégé par les optima locaux)

- **Plusieurs variantes** : ID3, C4.5, CART, CHAID
  - Différence principale : mesure de sélection d'un attribut – critère de branchement (split)

Mesures de sélection d'attributs

- Gain d'Information (ID3, C4.5)
- Indice Gini (CART)
- Table de contingence statistique χ^2 (CHAID)
- G-statistic

Bonne sélection et branchement ?

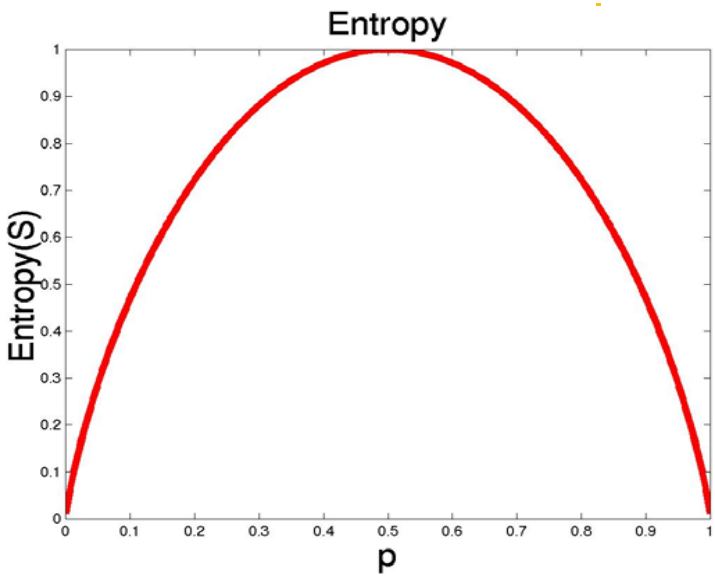


Gain d'information

- 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



- 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
  - Entropie(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$

Gain d'information

- Soient les ensembles $\{S_1, S_2, \ldots, S_v\}$ 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 :

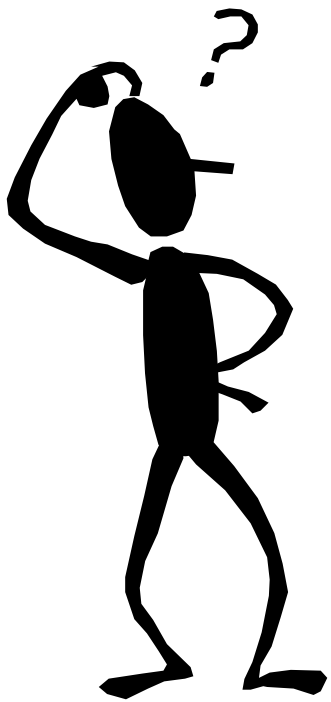
$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- 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 → besoin d'information minimal

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

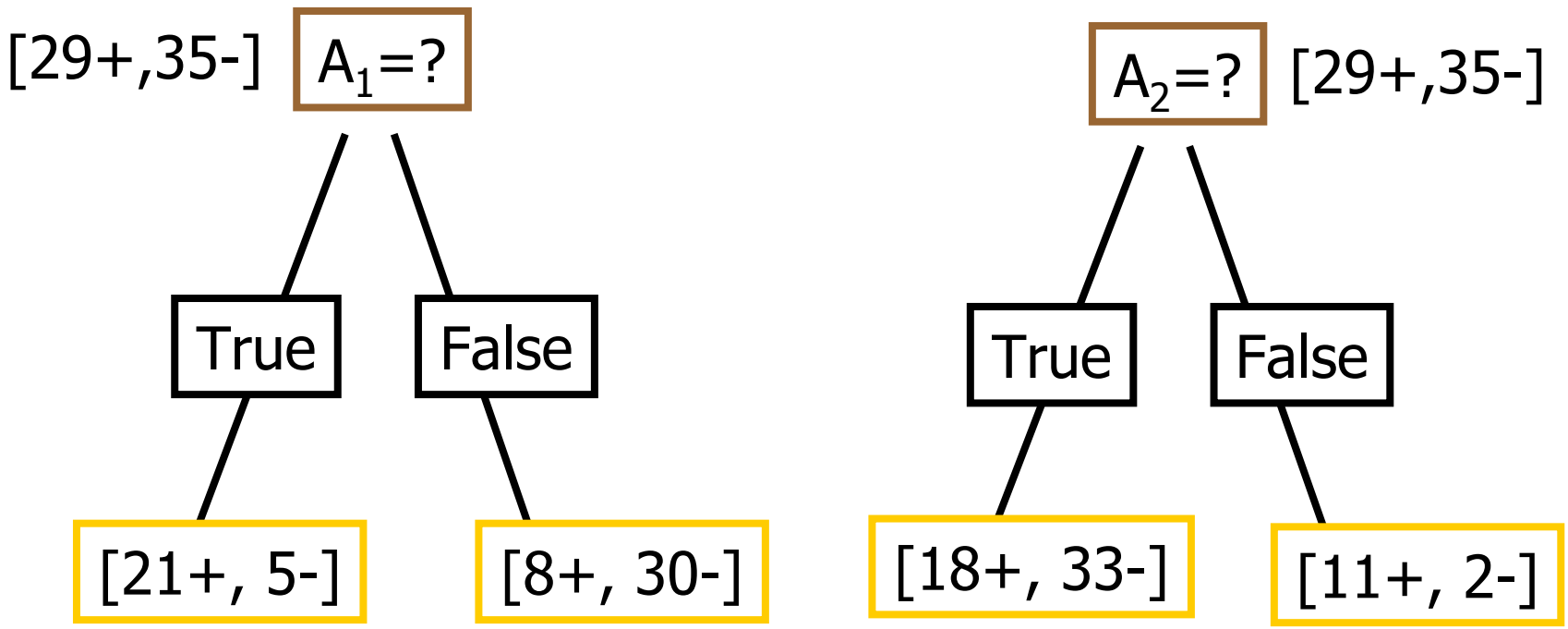
$$E(outlook) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

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" ?

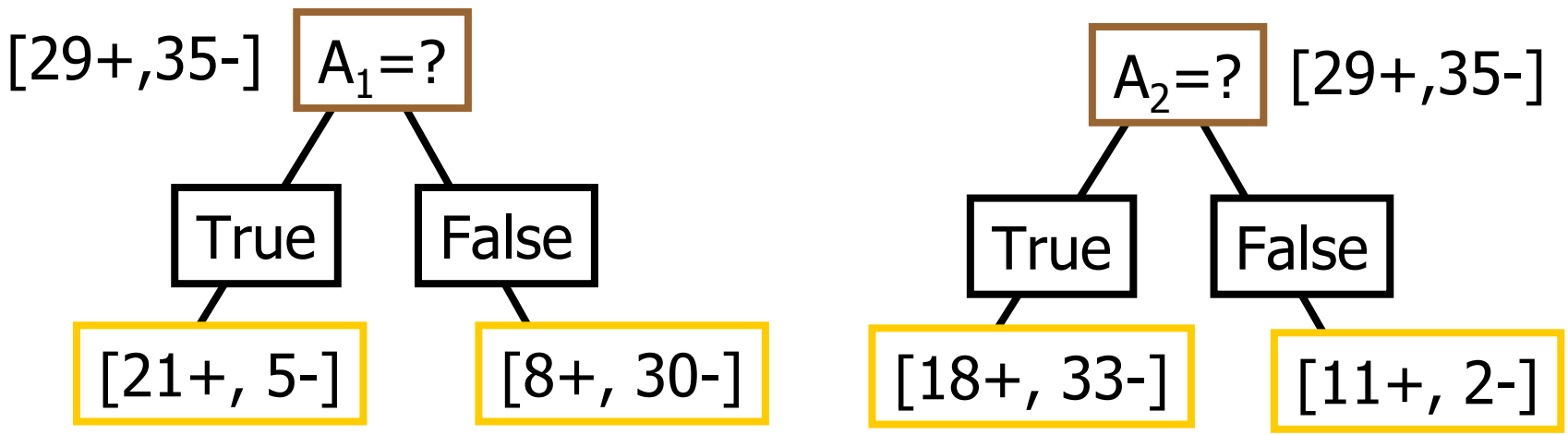


Gain d'information - Exemple

- Gain(S,A) : réduction attendue de l'entropie dûe au branchement de S sur l'attribut A

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

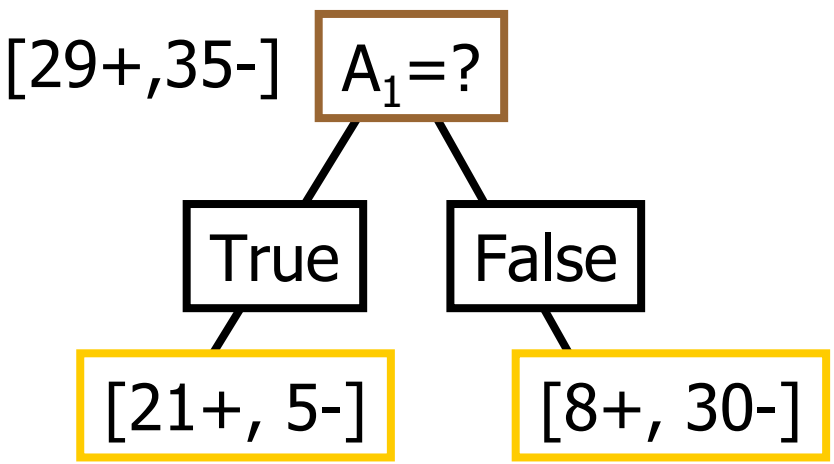
$$Entropie([29+,35-]) = -29/64 \log_2 29/64 - 35/64 \log_2 35/64 = 0.99$$

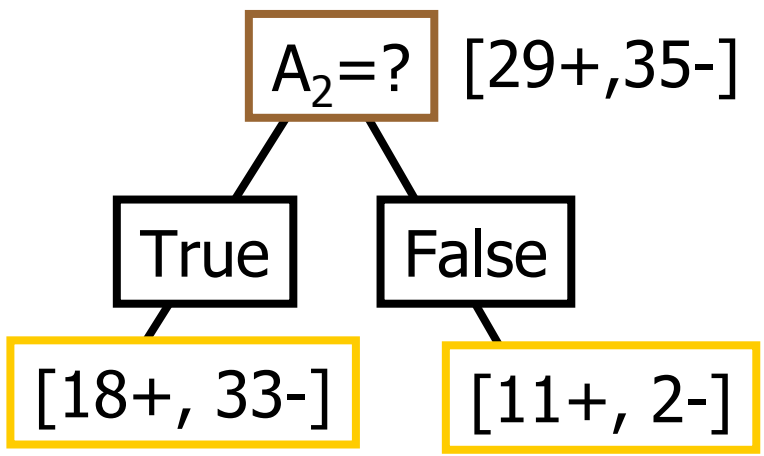


Gain d'information - Exemple

Entropie([21+,5-]) = 0.71
Entropie([8+,30-]) = 0.74
Gain(S,A_1)=Entropie(S)
-26/64*Entropie([21+,5-])
-38/64*Entropie([8+,30-])
=0.27

Entropie([18+,33-]) = 0.94
Entropie([8+,30-]) = 0.62
Gain(S,A_2)=Entropie(S)
-51/64*Entropie([18+,33-])
-13/64*Entropie([11+,2-])
=0.12

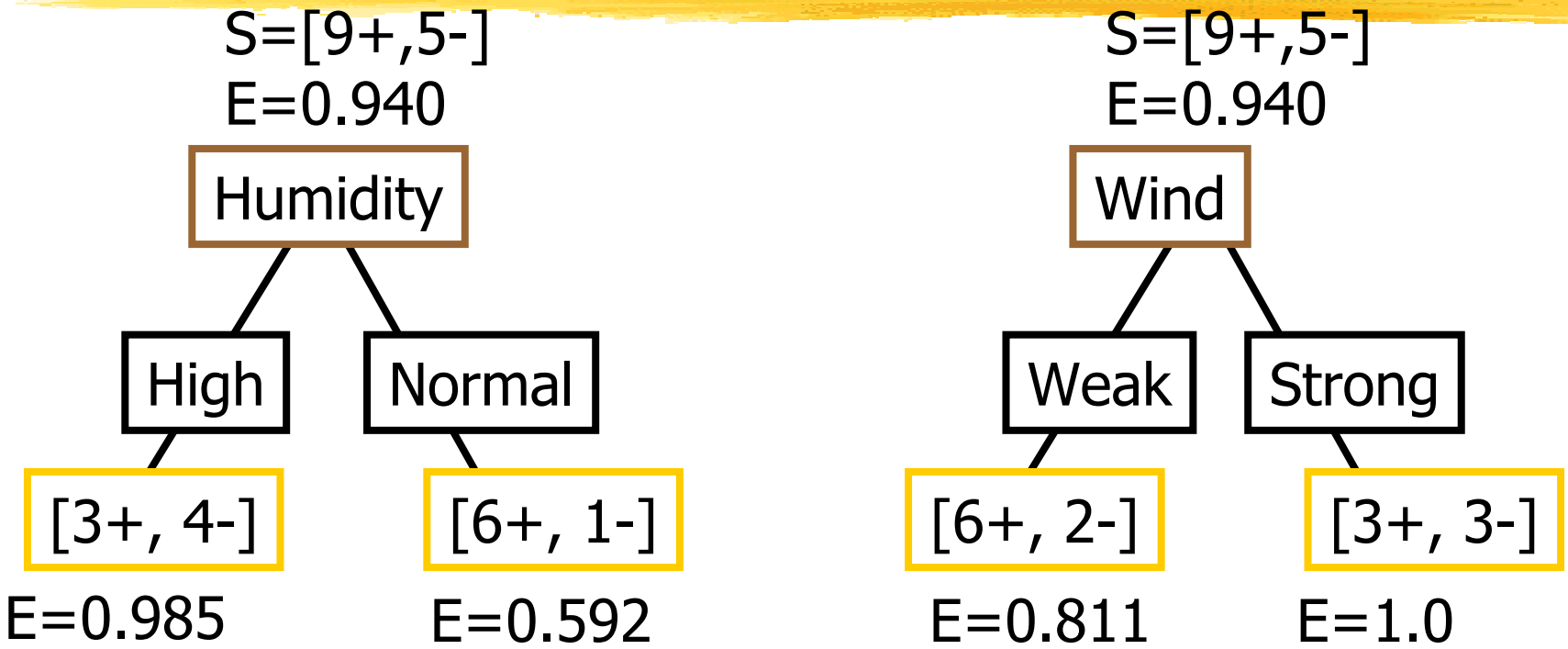




Exemple d'apprentissage

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

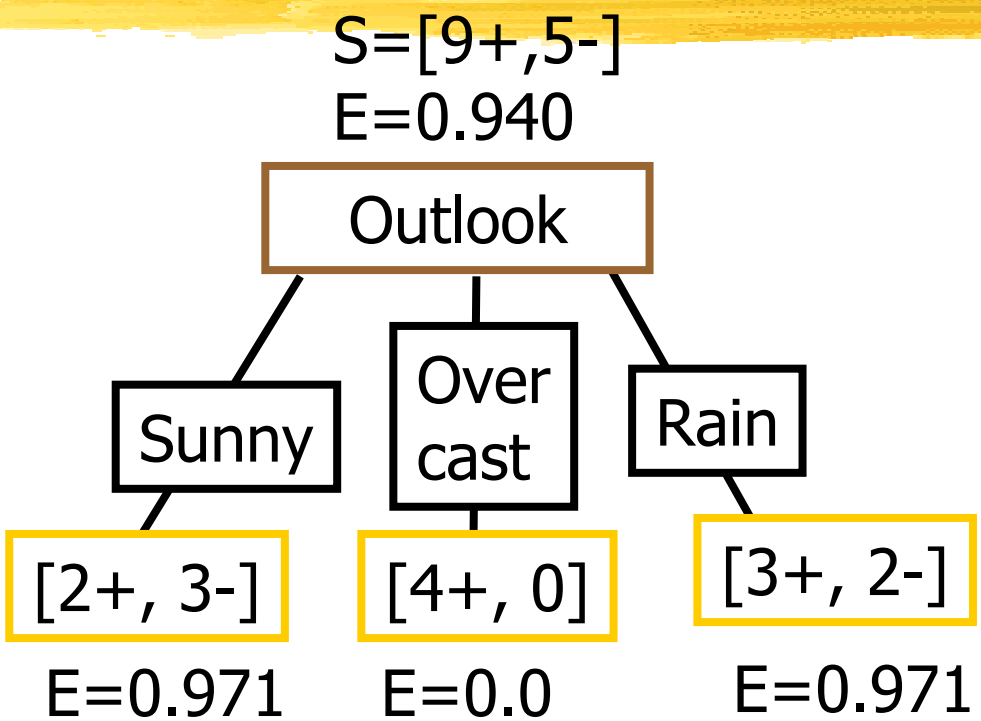
Sélection de l'attribut suivant



Gain(S,Humidity)
=0.940-(7/14)*0.985
– (7/14)*0.592
=0.151

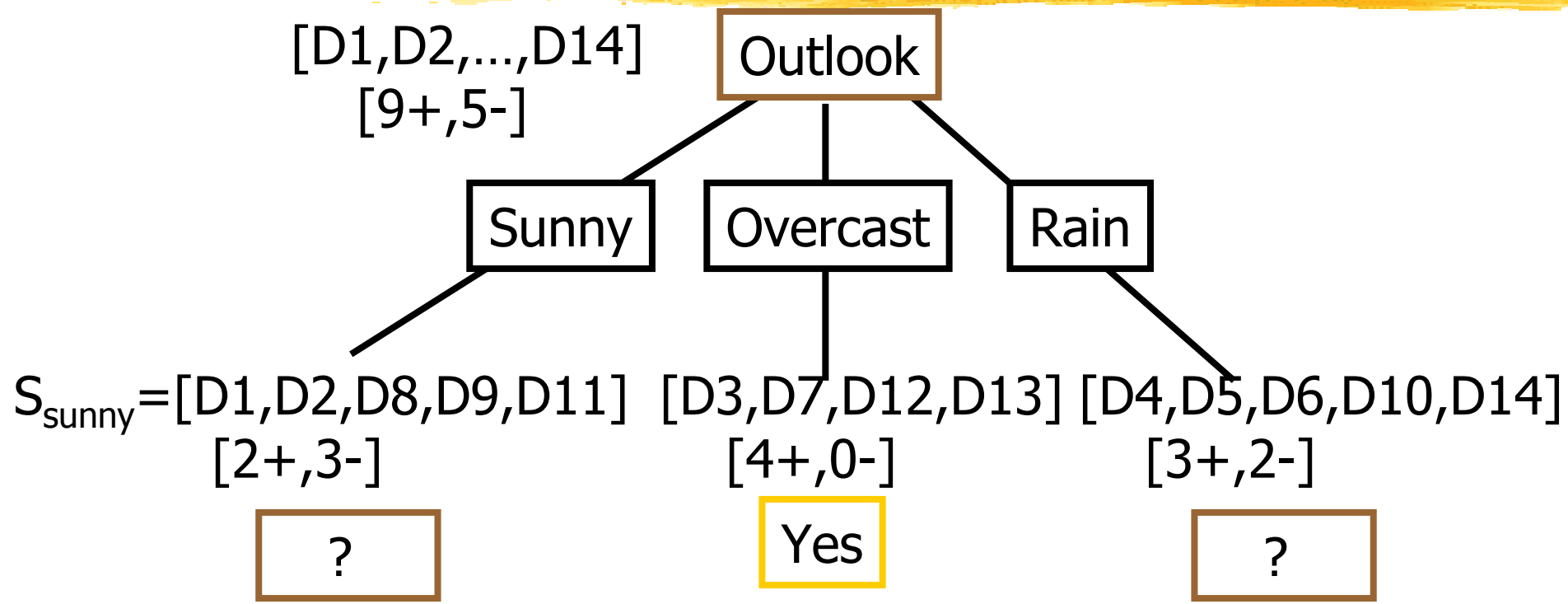
Gain(S,Wind)
=0.940-(8/14)*0.811
– (6/14)*1.0
=0.048

Sélection de l'attribut suivant



Gain(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247

Algorithme ID3

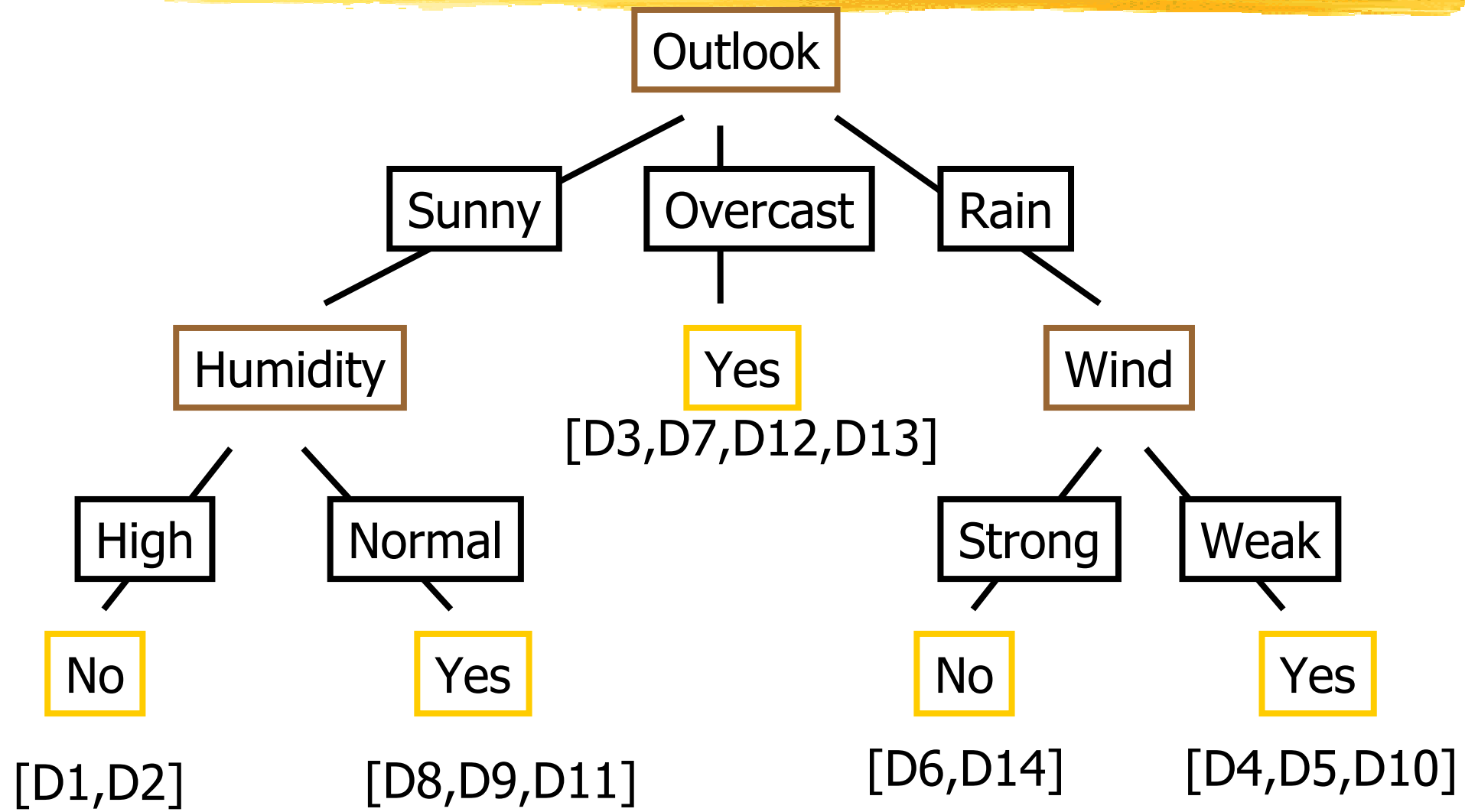


Gain(S_{sunny} , Humidity)=0.970-(3/5)0.0 – 2/5(0.0) = 0.970

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

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

Algorithme ID3



Indice Gini

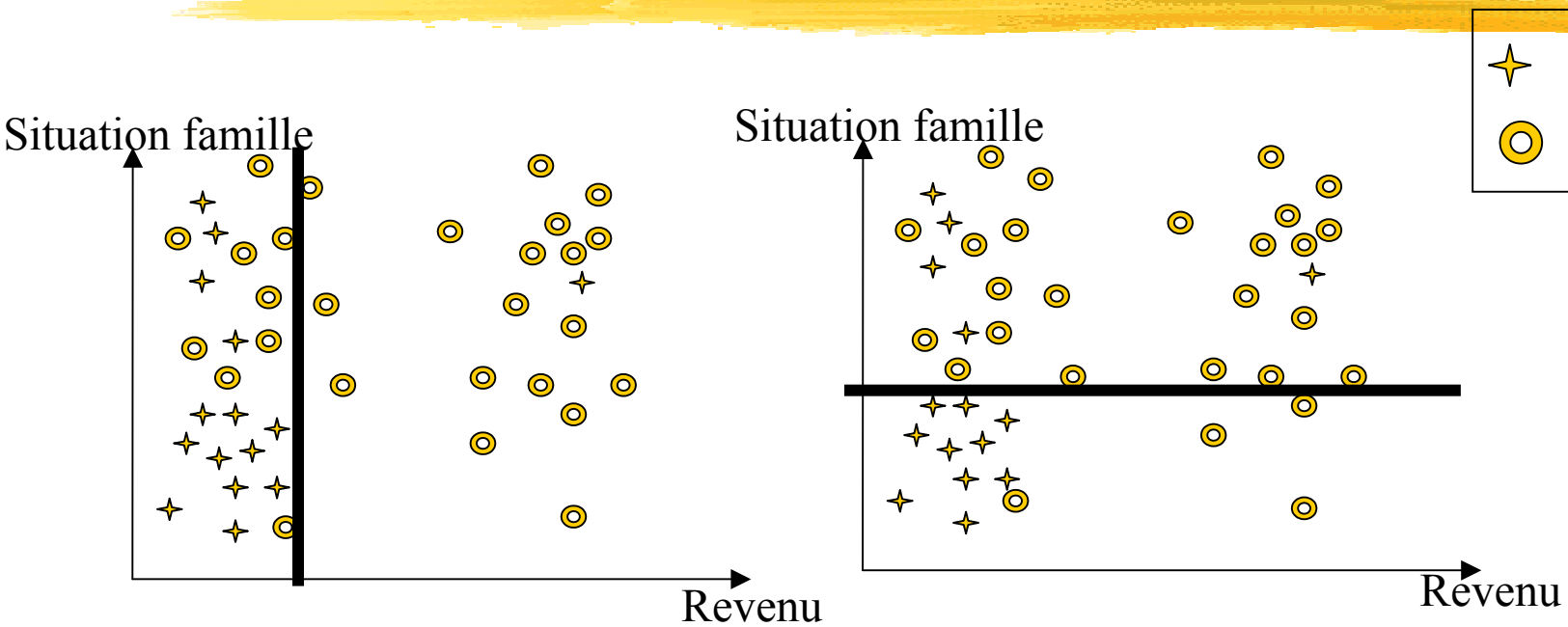
- Utiliser l'indice Gini pour un partitionnement pur

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

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

- 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



Calcul de Gini nécessite une Matrice de dénombrement

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

Gini(split) = **0.31**

	Non	Oui
M	5	23
F	10	4

Gini(split) = 0.34

Attributs énumératifs – indice GINI

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

Partage en plusieurs classes

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

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

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

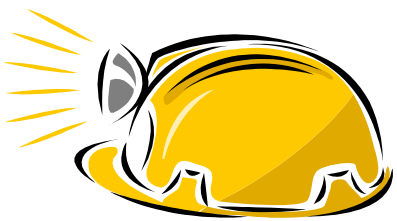
	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Attributs numériques – indice GINI

- 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

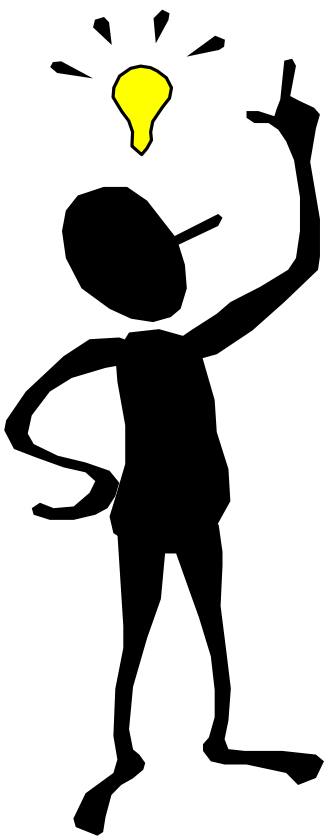
Fraude	No	No	No	Yes	Yes	Yes	No	No	No	No
	Revenu imposable									
Valeurs triées →	60	70	75	85	90	95	100	120	125	220
Positions Split →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3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

Méthodes à base d'arbres de décision



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



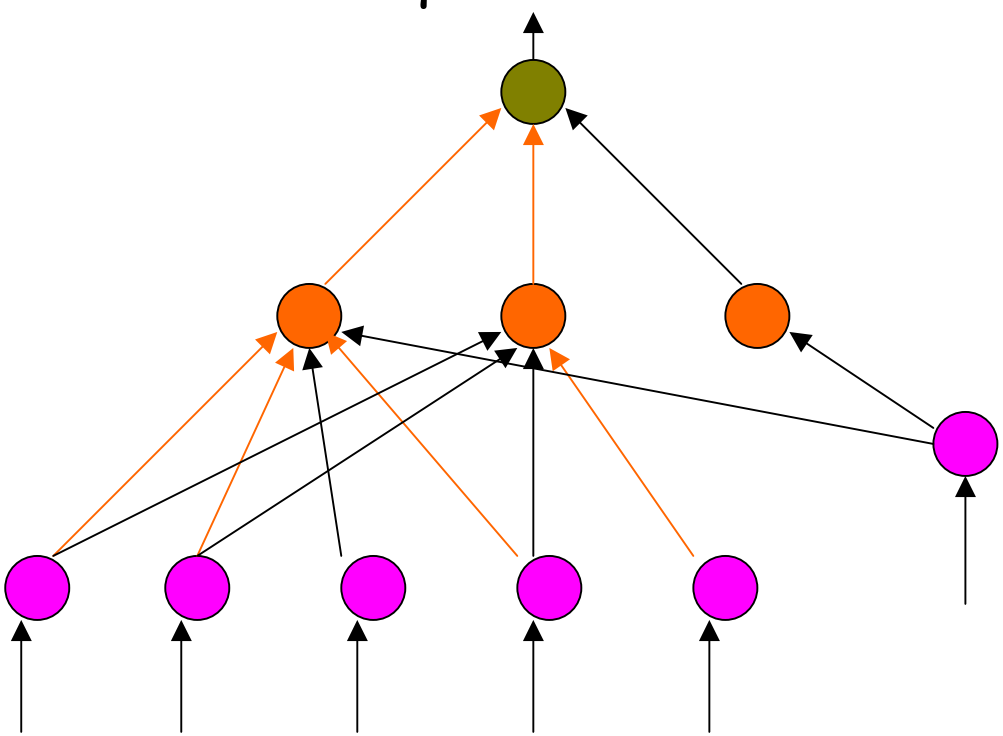
- Compréhensible pour tout utilisateur (lisibilité du résultat – règles - arbre)
- Justification de la classification d'une instance (racine → feuille)
- Tout type de données
- Robuste au bruit et aux valeurs manquantes
- Attributs apparaissent dans l'ordre de pertinence → tâche de pré-traitement (sélection 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 relance la phase d'apprentissage

Réseaux de neurones

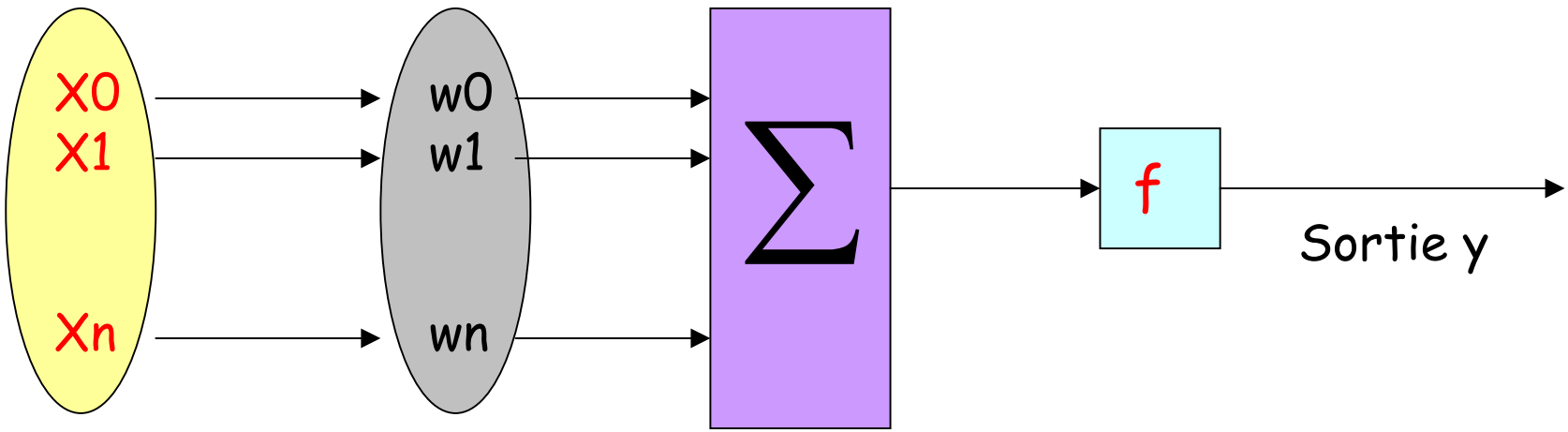
- Réseau neuronal : simule le système nerveux biologique
- Un réseau de neurones est composé de plusieurs neurones interconnectés. Un poids est associé à chaque arc. A chaque neurone on associe une valeur.



- Temps de "switch" d'un neurone > 10^{-3} secs
- Nombre de neurones (humain) ~10^{10}
- Connexions (synapses) par neurone : ~10^4–10^5

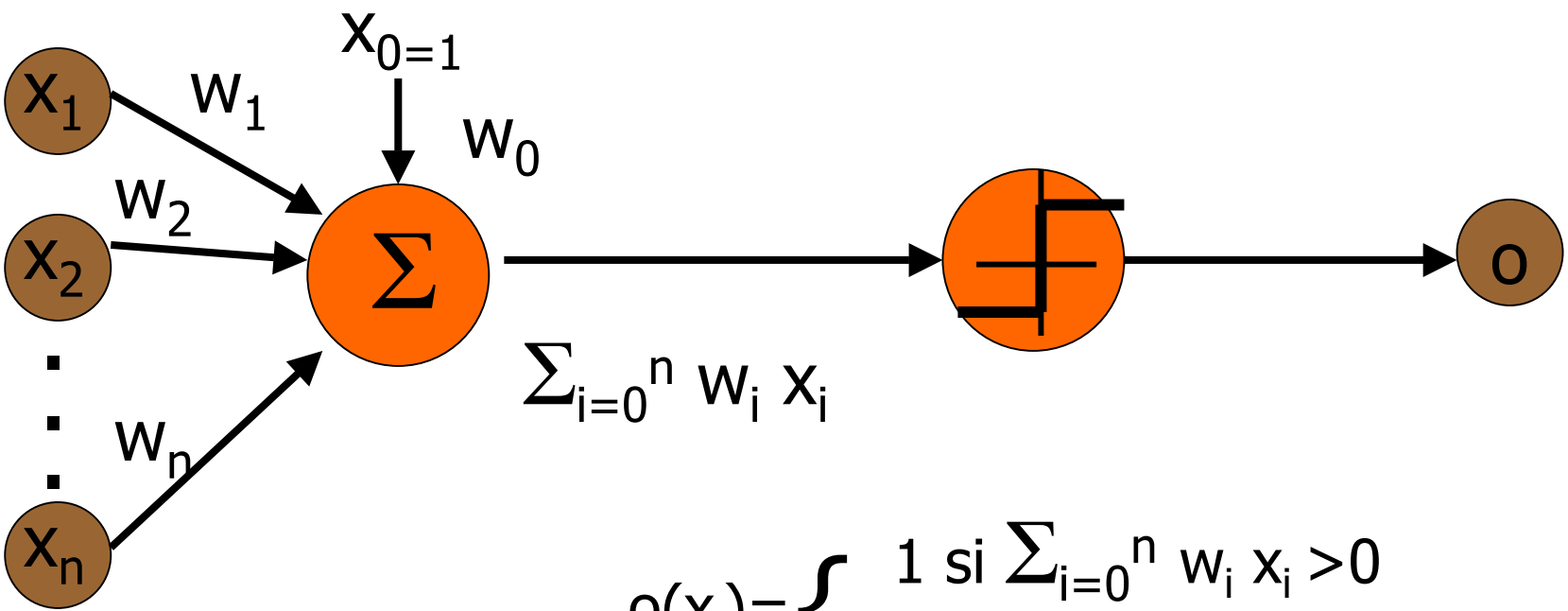
Neurone ou perceptron

- Neurone = Unité de calcul élémentaire
- Le vecteur d'entrée X est transformé en une variable de sortie y, par un produit scalaire et une fonction de transformation non linéaire



Neurone ou perceptron

Linear treshold unit (LTU)



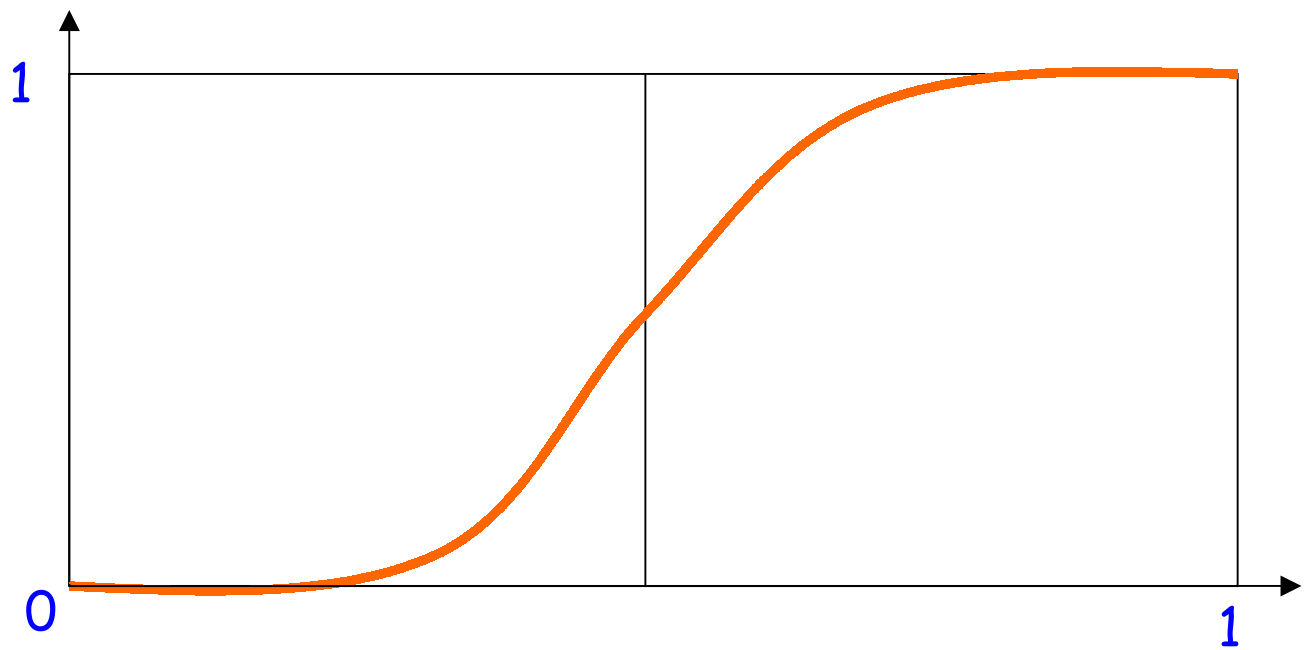
$$o(x_i) = \begin{cases} 1 \text{ si } \Sigma_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ sinon} \end{cases}$$

Neurone

- Fonction d'activation la plus utilisée est la fonction sigmoide

$$\sigma(x)=\frac{1}{1+e^{x}}$$

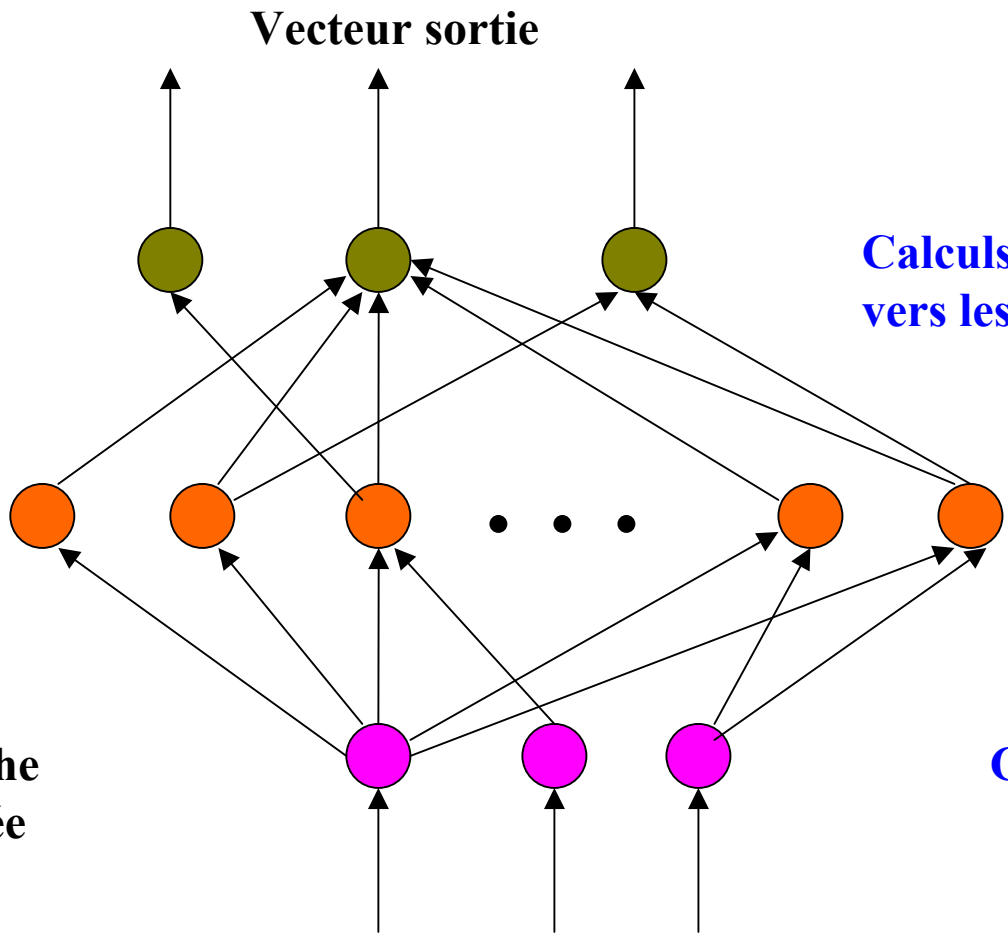
- Elle prend ses valeurs (entrée et sortie) dans l'intervalle [0,1]



Réseaux de neurones

- Capacité d 'apprentissage : apprendre et changer son comportement en fonction de toute nouvelle expérience.
- Permettent de découvrir automatiquement des modèles complexes.
- Plusieurs modèles de réseaux de neurones : PMC (Perceptron Multi-Couches), RBF (Radial Basis Function), Kohonen, ...

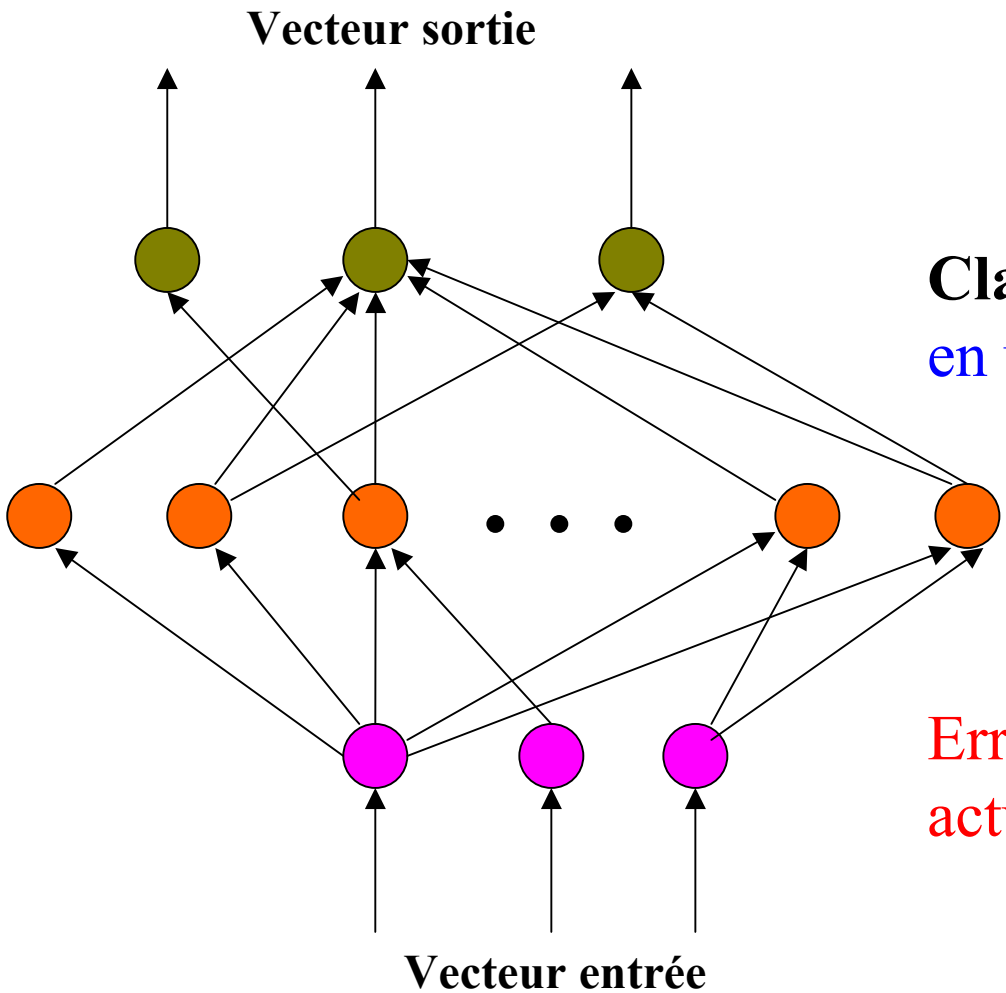
Perceptron Multi Couches (PMC)



Calculs effectués des entrées vers les sorties

Graphe complet

Paradigme d'apprentissage



Classification : Ajuster les poids en utilisant l'erreur

Erreur = Valeur désirée – Valeur actuelle

Algorithmes d'apprentissage

- Rétro-propagation du gradient (Back propagation)
- Kohonen
- RBF (Radial basis function)
- Réseaux de neurones probabilistes
- ART (Adaptive resonance theory)
- ...

Rétro-propagation du gradient

Principales étapes

- Construction du réseau
  - Représentation des entrées
  - Nombre de couches, nombre de noeuds dans chaque couche
- Apprentissage du réseau utilisant les données disponibles
- Elagage du réseau
- Interprétation des résultats

Construction du réseau

- Nombre de noeuds en entrée : correspond à la dimension des données du problème (attributs ou leurs codages).

Normaliser dans l'intervalle [0,1].

Exemple énumératif : Attribut *A* prenant ses valeurs {1,2,3,4,5}

- 5 entrées à valeurs binaires ; 3 = 00100
- 3 bits ; 3 = 010
- 1 entrée réelle ; 0, 0.25, 0.5, 0.75, 1

Construction du réseau

- Nombre de couches cachées : Ajuster pendant l'apprentissage.
- Nombre de nœuds par couche : Le nombre de nœuds par couche est au moins égal à deux et au plus égal au nombre de nœuds en entrée
- Nombre de nœuds en sortie : fonction du nombre de classes associées à l'application.

- Réseau riche → pouvoir d'expression grand (Ex. 4-2-1 est moins puissant que 4-4-1)
- Attention : Choisir une architecture riche mais pas trop – Problème de sur-spécialisation

Apprentissage du réseau

- Objectif principal : obtenir un ensemble de poids qui font que la plupart des instances de l'ensemble d'apprentissage sont correctement classées.
- Etapes :
  - Poids initiaux sont *générés* aléatoirement
  - Les vecteurs en entrée sont traités en séquentiel par le réseau
  - Calcul des valeurs d'activation des nœuds cachés
  - Calcul du vecteur de sortie
  - Calcul de l'erreur (sortie désirée – sortie actuelle).

$$e(PMC)=\frac{1}{2}\sum_{x\in S}\left(d(x)-a(x)\right)^2$$

  - d(x) : sortie désirée, a(x) : sortie actuelle

Apprentissage du réseau

- Les poids sont mis à jour en utilisant l'erreur. Le nombre d'instances qui sont passés dans le réseau avant la mise à jour des poids est un paramètre (entre 1 – convergence rapide et minimum local - et m – convergence lente -).

- Rétro propagation à l'aide de la méthode de gradient. Le paramètre taux d'apprentissage [0,1] influe sur la modification des poids.

  Valeur grande : modification forte ; Valeur petite : modification minime

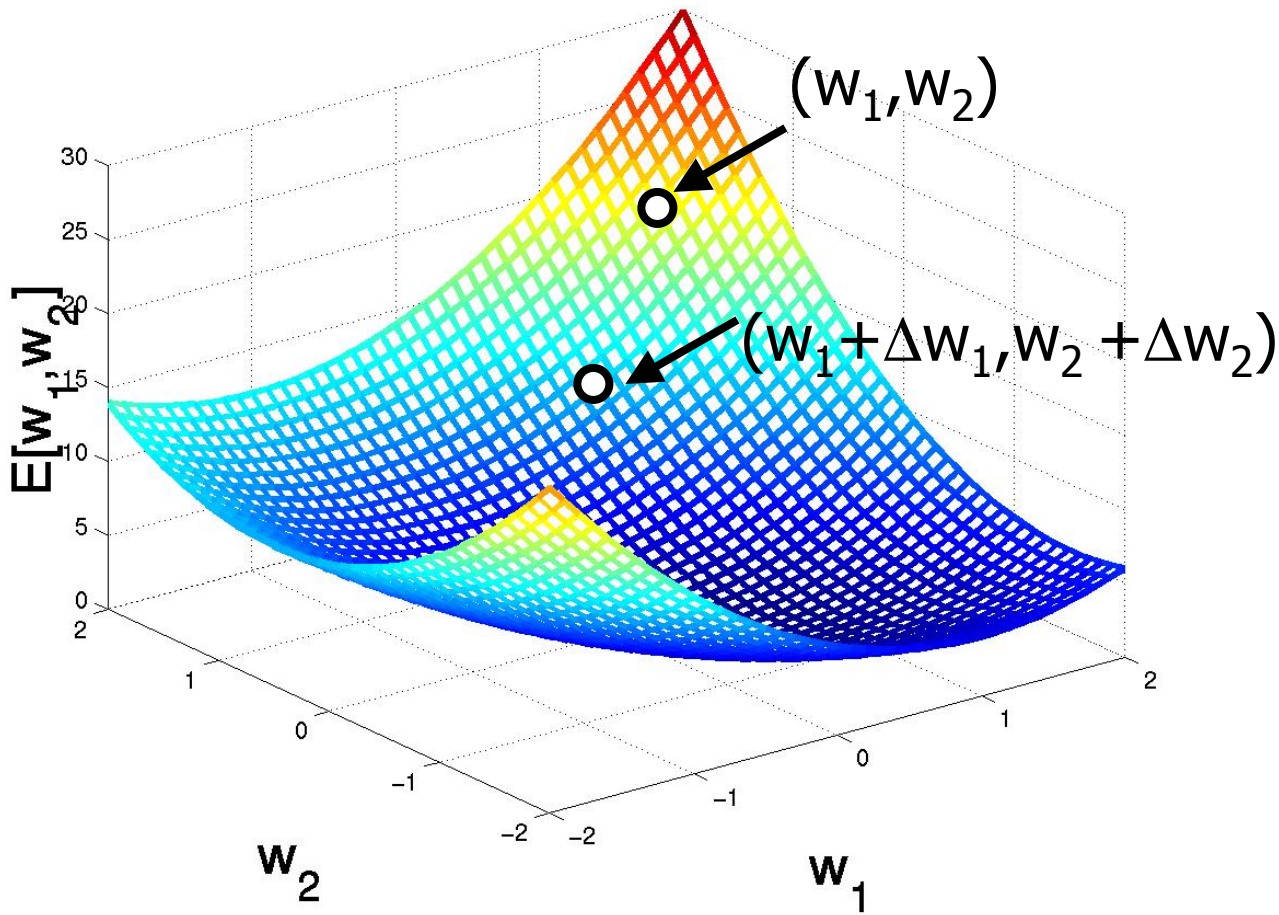
Apprentissage du réseau

$w_i = w_i + \Delta w_i$
$\Delta w_i = \eta\ (t - o)\ x_i$
t=c(x) est la valeur désirée
o est la sortie obtenue
η est le taux d'apprentissage (e.g 0.1)

- Critère d'arrêt : la tolérance définit l'erreur cible.

  et/ou Nombre d'instances bien classées (seuil)

Apprentissage du réseau

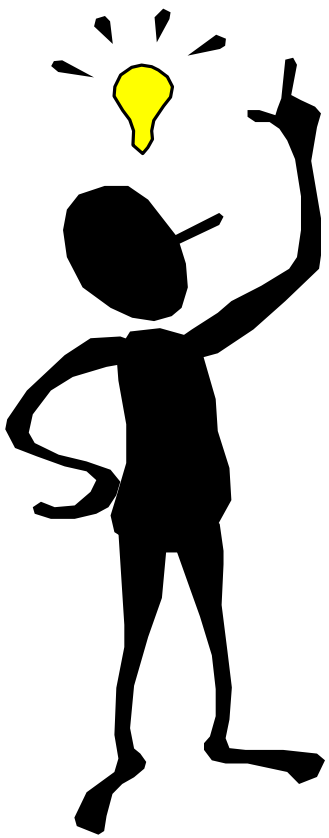


Elagage du réseau

- Réseau fortement connexe est difficile à articuler
- N nœuds en entrée, h couches cachées, et m nœuds en sortie → h(m+n) arcs (poids)
- Elagage : Supprimer les arcs et les nœuds qui n'affectent pas le taux d'erreur du réseau. Eviter le problème de sur-spécialisation (over-fitting).

  Ceci permet de générer des règles concises et compréhensibles.

Réseaux de neurones - Avantages



- Taux d'erreur généralement bon
- Outil disponible dans les environnements de data mining
- Robustesse (bruit) – reconnaissance de formes (son, images sur une rétine, …)
- Classification rapide (réseau étant construit)
- Combinaison avec d'autres méthodes (ex : arbre de décision pour sélection d'attributs)

Réseaux de neurones - Inconvénients

- Apprentissage très long
- Plusieurs paramètres (architecture, coefficients synaptiques, …)
- Pouvoir explicatif faible (boite noire)
- Pas facile d'incorporer les connaissances du domaine.
- Traitent facilement les attributs numériques et binaires
- Evolutivité dans le temps (phase d'apprentissage)

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) $X=<x_1,\ldots,x_k>$ est dans la classe *C*
- Par exemple
  - **P(**classe=***N*** **/** outlook=sunny,windy=true,…**)**
- **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 naive

- Hypothèse Naïve : indépendance des attributs
  - $P(x_1,\ldots,x_k/C) = P(x_1/C)\cdot\ldots\cdot P(x_k/C)$

  $P(x_i/C)$ 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(sunny | p) = 2/9	P(sunny | n) = 3/5
P(overcast | p) = 4/9	P(overcast | n) = 0
P(rain | p) = 3/9	P(rain | n) = 2/5
Temperature	
P(hot | p) = 2/9	P(hot | n) = 2/5
P(mild | p) = 4/9	P(mild | n) = 2/5
P(cool | p) = 3/9	P(cool | n) = 1/5

Humidity	
P(high | p) = 3/9	P(high | n) = 4/5
P(normal | p) = 6/9	P(normal | n) = 1/5
Windy	
P(true | p) = 3/9	P(true | n) = 3/5
P(false | p) = 6/9	P(false | n) = 2/5

Classification bayésienne – Exemple (1)

- Classification de *X* :
  - Une instance inconnue *X* = <*rain, hot, high, false*>
  - **P**(*X*/*p*)·**P**(*p*) =
    P(*rain*/*p*)·**P**(*hot*/*p*)·**P**(*high*/*p*)·**P**(*false*/*p*)·**P**(*p*) =
    3/9·2/9·3/9·6/9·9/14 = 0.010582
  - **P**(*X*/*n*)·**P**(*n*) =
    P(*rain*/*n*)·**P**(*hot*/*n*)·**P**(*high*/*n*)·**P**(*false*/*n*)·**P**(*n*) =
    2/5·2/5·4/5·2/5·5/14 = **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

Etude de cas

Prédiction de structure de la protéine

Les protéines

- Une protéine = séquence d'acides aminés définie par un gêne et ayant une fonction spécifique dans la cellule

« Building block of life »

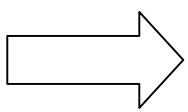
- Les protéines sont partout :
  - Protéines enzymatiques (catalyse)
  - Protéines de transport : hémoglobine (oxygène), albumine (corps gras) …
  - Protéine messager : insuline …
  - Protéines récepteur
  - Protéines sériques : anticorps
  - Protéines structurelles : collagène dans la peau, kératine dans les cheveux, …
  - …

Les protéines

- 20 acides aminés distincts, chaque acide aminé étant constitué de (jusqu'à) 18 atomes
- Une séquence protéique est constituée de 50 à 2000 acides aminés
- 3000 à 4000 protéines dans une cellule
- Une protéine se replie « en pelote », adoptant une configuration spatiale caractéristique de sa fonction

Les 20 Acides Aminés

- A Ala Alanine
- C Cys Cysteine
- D Asp Aspartic
- E Glu Glutamic
- F Phe Phenylalanine
- G Gly Glycine
- H His Histidine
- I Ile Isoleucine
- K Lys Lysine
- L Leu Leucine
- M Met Methionine
- N Asn Asparagine
- P Pro Proline
- Q Gln Glutamine
- R Arg Arginine
- S Ser Serine
- T Thr Threonine
- V Val Valine
- W Trp Tryptophan
- Y Tyr Tyrosine



20 Lettres de l'alphabet

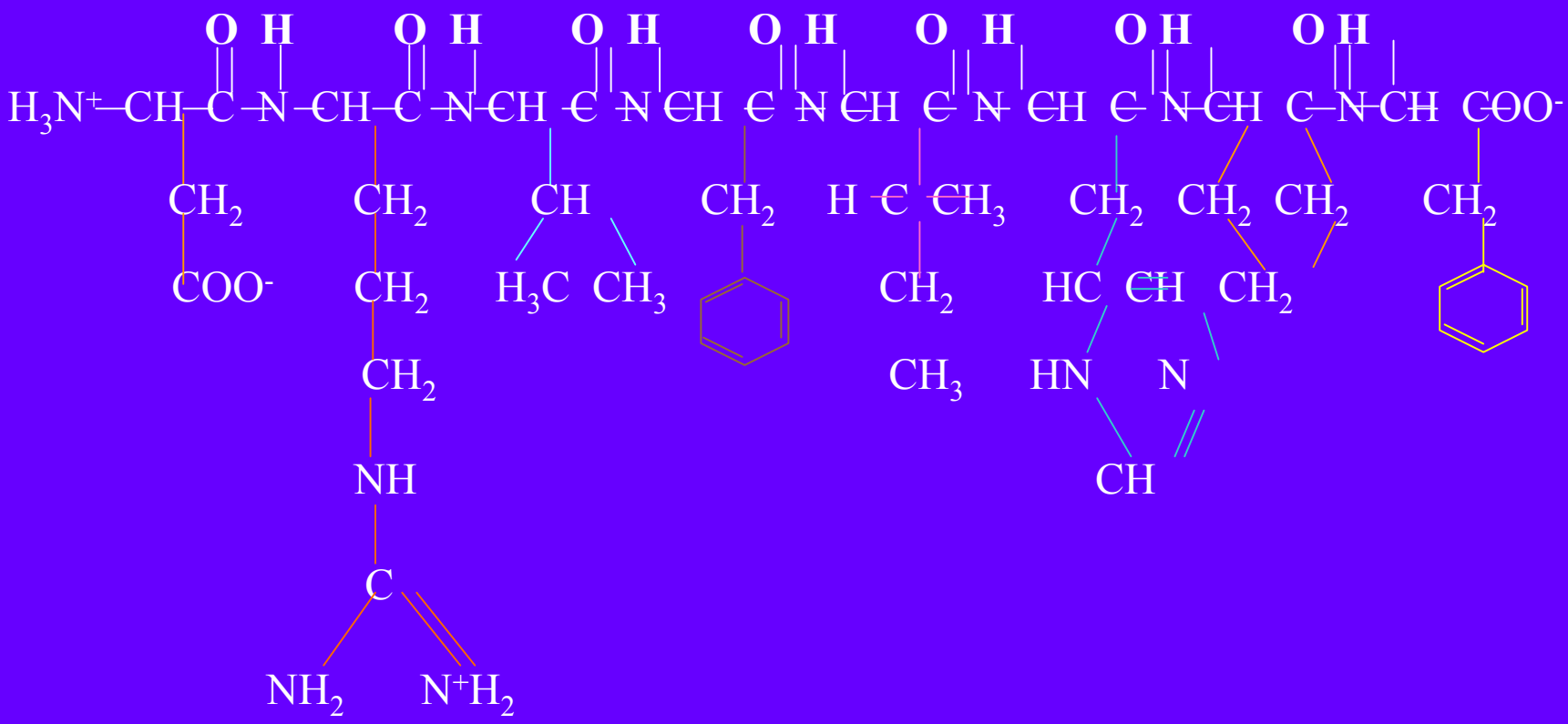
Les structures

- Structure primaire = ordre dans lequel sont enchaînés les acides aminés dans la molécule

- Structure secondaire = rotation des atomes de la chaîne peptidique les uns par rapport aux autres au cours de la synthèse de la chaîne

- Structure tertiaire = résultat de liaisons diverses (hydrogène, hydrophobes, électrostatiques, covalentes,...) entre des acides aminés de la même chaîne peptidique mais non voisins dans la structure primaire

Structure primaire



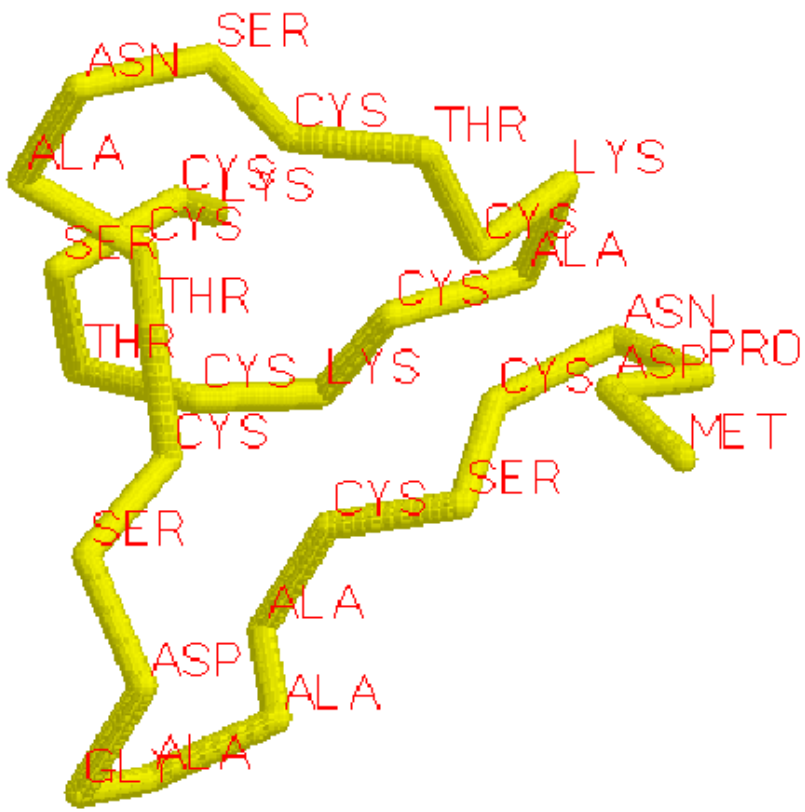
Asp Arg Val Tyr Ile His Pro Phe

D R V Y I H P F

Séquence de la protéine : DRVYIHPF

Protein Folding Problem

Etant donné une séquence primaire de la protéine, ex.,
MDPNCSCAAAGDSCTCANSCTCLACKCTSCK,
prédire la structure secondaire et 3D.



Base de données

Structures prédites (connues) :

Protein Data Bank (PDB) (centaine de structures non redondantes) [www.rcsb.org/pdb/]

Base de données de séquences de protéines :

Genbank (milliers de séquences)
[www.ncbi.nlm.nih.gov/Genbank/GenbankSearch.html]

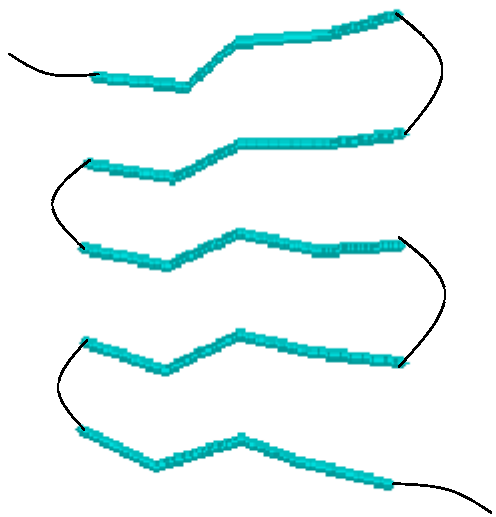
SWISSPROT

[www.ebi.ac.uk/swissprot]

Structure secondaire

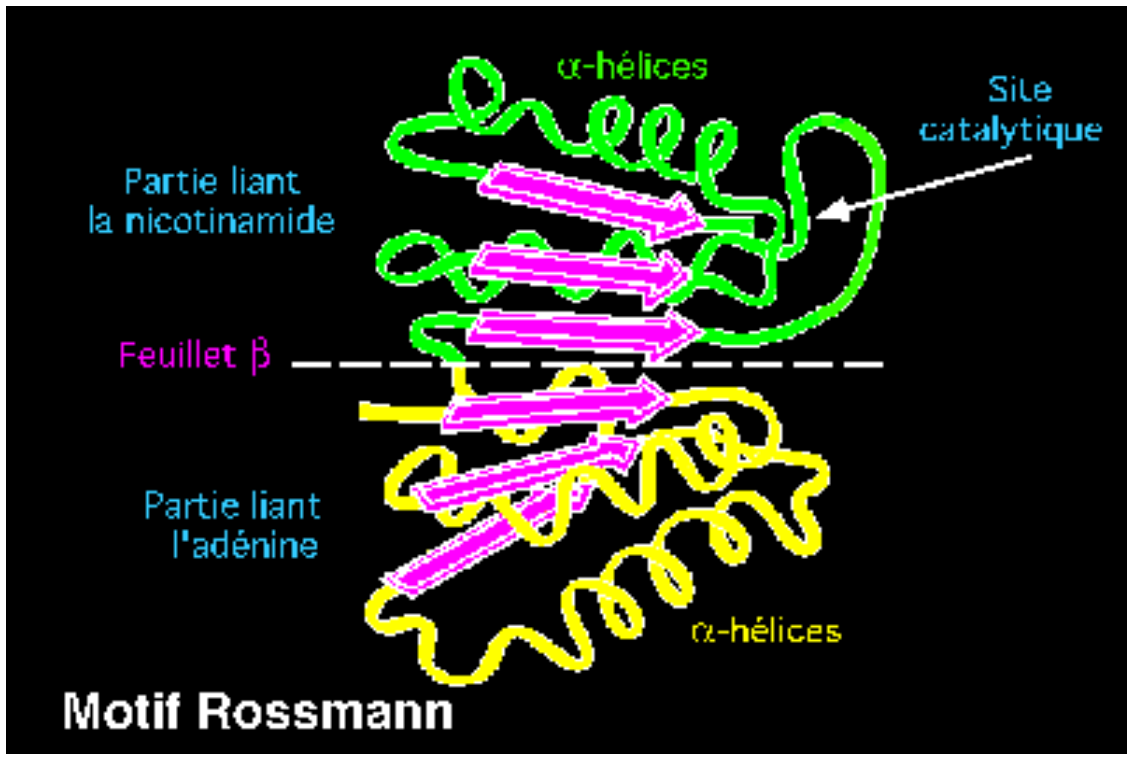
- Hélice α

- Feuillet ß parallèle : tous les segments ont la même orientation
- Feuillet ß anti-parallèle
- Feuillet ß mixte



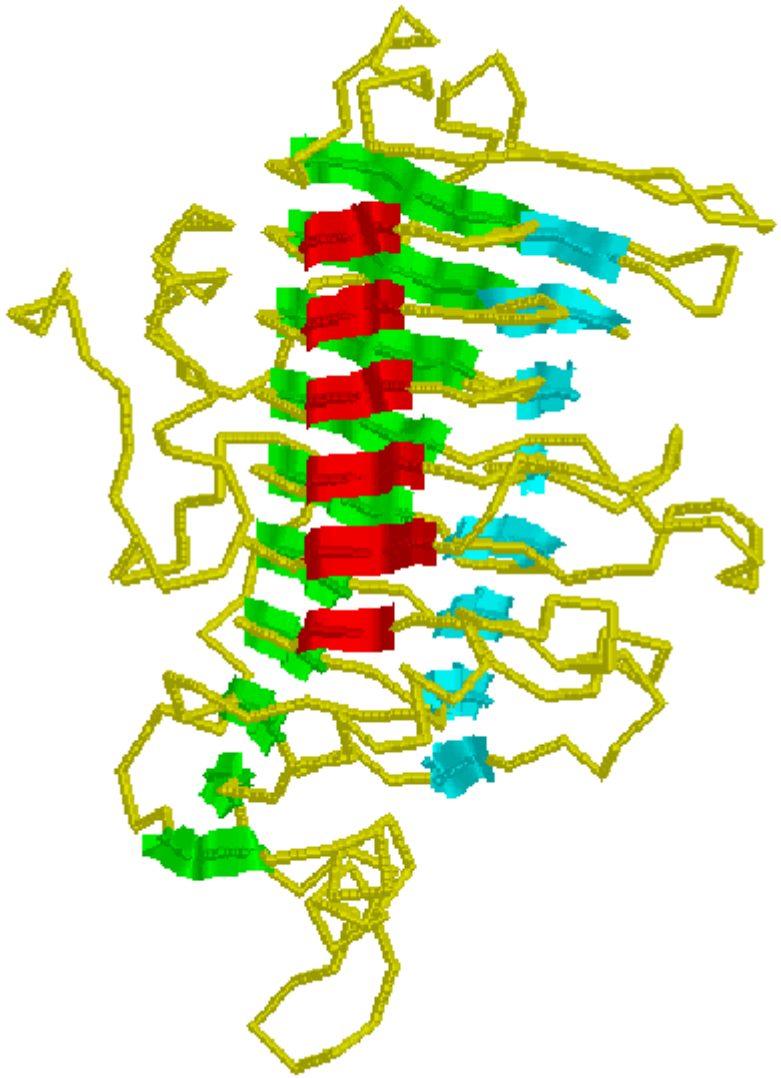
Structure secondaire

- Hélice α
- Feuillet ß parallèle : tous les segments ont la même orientation
- Feuillet ß anti-parallèle
- Feuillet ß mixte



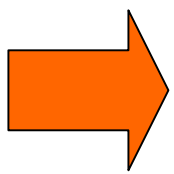
Structure secondaire

- Beta Hélice



Structure 3D

- Permet de comprendre le mode d'action d'une protéine : activité enzymatique, interaction avec d'autres protéines (ligands, substrats, récepteur, épitope, *etc.*).



Structure primaire

Structure secondaire / tertiaire

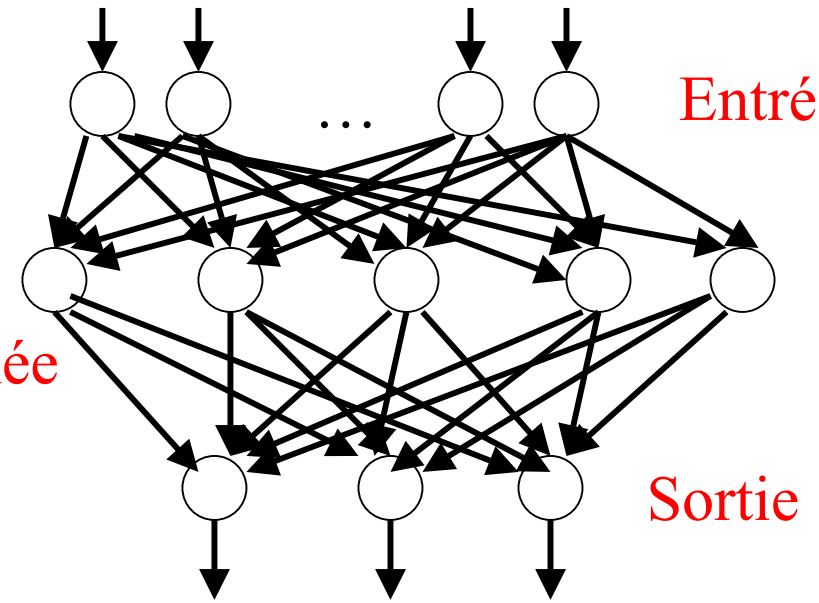
Réseaux de neurones

- Le processus neuronal de base traite des signaux d'entrée d'un ou plusieurs neurones et envoie un signal de sortie à un ou plusieurs (un 0 ou un 1)
- Le signal de sortie à chaque neurone récepteur est pondéré – ces poids sont ajustés par entraînement du modèle avec des séquences de structures connues
- Le programme donne une évaluation de fiabilité de chaque prévision basée sur la force des signaux d'une hélice alpha, d'un feuillet bêta et d'une boucle

Référence : Rost B, Sander C (1994) Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins*, **19**, 55-72

Réseaux de neurones

- Entrée : structure primaire
- Sortie : indication sur la structure secondaire



Efficacité > 70%

Plus proches voisins

- Une liste de fragments courts de séquence est faite en glissant une fenêtre de longueur n le long d'un ensemble d'approximativement 100-400 séquence d'entraînement de structure connue mais de similitude minimale
- La structure secondaire de l'acide aminé central dans chaque fenêtre d'entraînement est enregistrée
- Une fenêtre coulissante de même taille est alors choisi parmi la séquence de requête

Plus proches voisins

- La séquence dans la fenêtre à chaque position de la séquence demandée est comparée à chacun des fragments d'entraînement et les 50 meilleurs fragments appariés sont identifiés → Nécessité d'une notion de distance
- Les fréquences de la structure secondaire connue de l'acide aminé du milieu dans chacun de ces fragments appariés (H, B et C) sont alors employés pour prévoir la structure secondaire de l'acide aminé du milieu de la fenêtre de requête
- Des règles ou un NN sont utilisées pour faire la prédiction finale pour chaque AA.

Liens Web - Logiciels

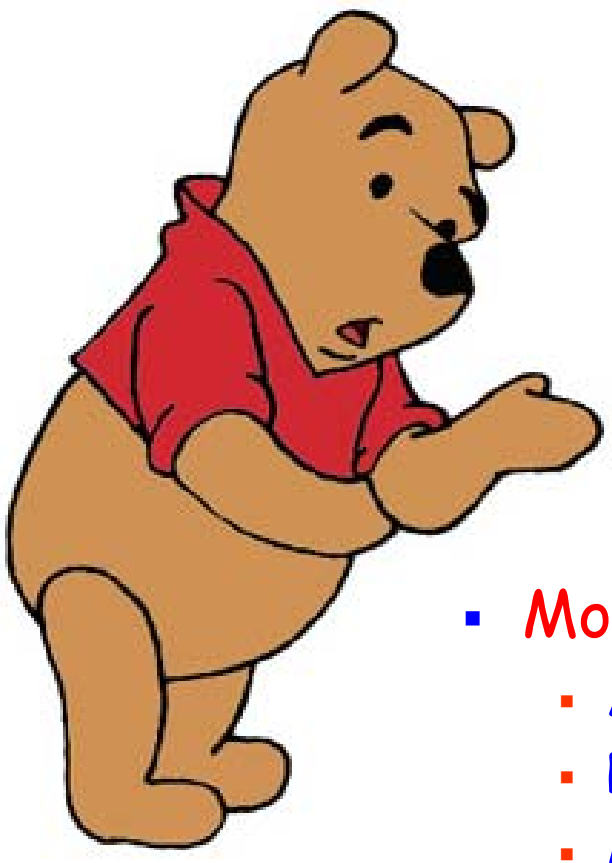
- http://dot.imgen.bcm.tmc.edu:9331/seq-search/struc-predict.html
- http://jura.ebi.ac.uk:8888/jnet/
- http://www.embl-heidelberg.de/predictprotein/
- http://cubic.bioc.columbia.edu/predictprotein

  - *(B Rost: PHD: predicting one-dimensional protein structure by profile based neural networks. Methods in Enzymology, 266, 525-539, 1996)*

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-)

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

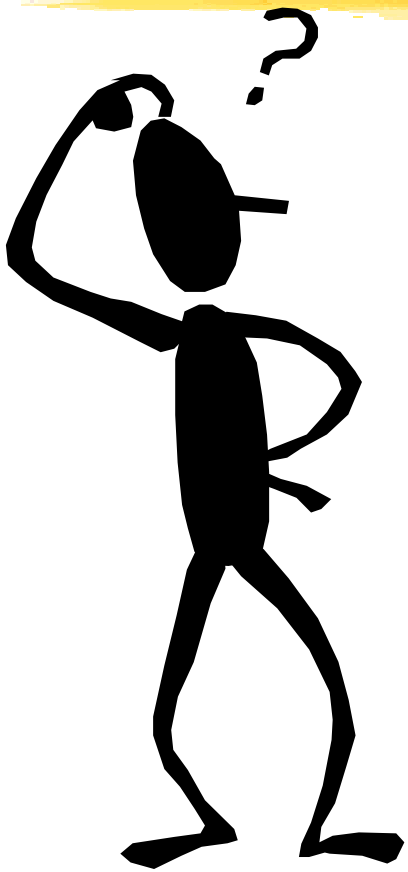
Difficile à comprendre

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



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, Sucre, Pain

Client 1

Lait, Oeufs, Céréale, Lait

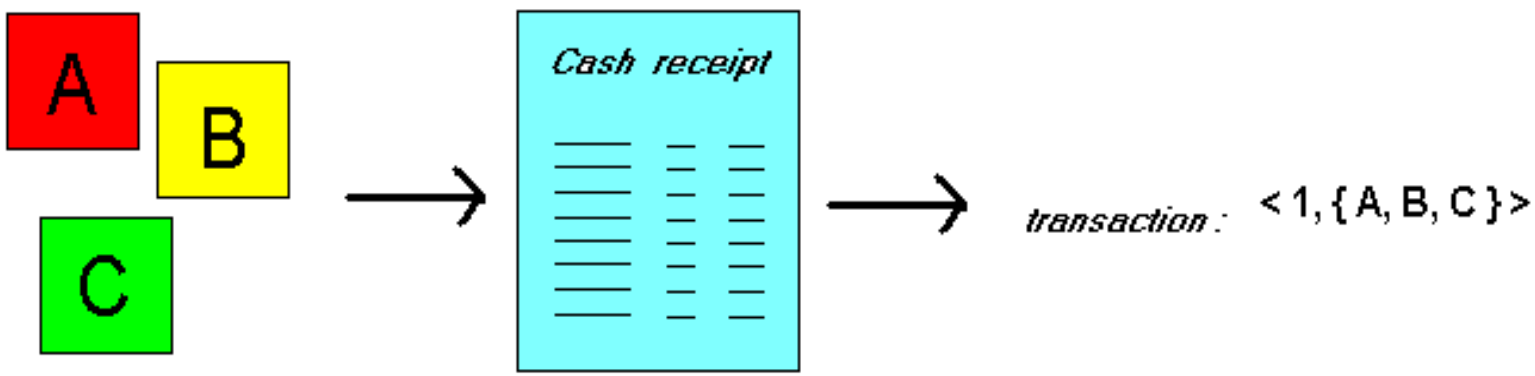
Client 2

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, …

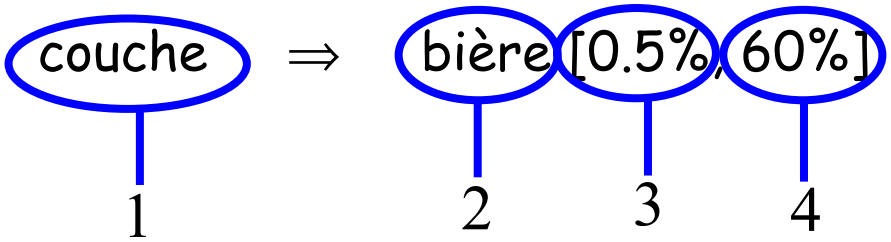
Règles d'associations

- **Recherche de règles d'association :**
  - Découvrir des patterns, corrélations, associations fréquentes, à partir d'ensembles d'items contenus dans des base de données.
- **Compréhensibles** : Facile à comprendre
- **Utiles** : Aide à la décision
- **Efficaces** : Algorithmes de recherche
- **Applications** :
  - Analyse des achats de clients, Marketing, Accés Web, Design de catalogue, Génomique, etc.

Règles d'associations

- **Formats de représentation des règles d'association :**
  - couches ⇒ bière [0.5%, 60%]
  - achète:couches ⇒ 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") ⇒ 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"

1. **Condition**, partie gauche de la règle
2. **Conséquence**, partie droite de la règle
3. **Support**, fréquence ("partie gauche **et** droite sont présentes ensemble dans la base")
4. **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.

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

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

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

Exemple

TID	Items
1	Pain, Lait
2	Bière, Couches, Pain, Oeufs
3	Bière, Coca, Couches, Lait
4	Bière, Pain, Couches, Lait
5	Coca, Pain, Couches, Lait

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

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

Règle : $X \Rightarrow_{s,\alpha} y$

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

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

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

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 :** σ = 2 - 10 %, γ = 70 - 90 %

Règles d'associations

- **Etant donné** : (1) un base de données de transactions, (2) chaque transaction est un ensemble d'articles (items) achetés

Transaction ID	Items achetés
100	A,B,C
200	A,C
400	A,D
500	B,E,F

Itemset fréquent	Support
{A}	3 ou 75%
{B} et {C}	2 ou 50%
{D}, {E} et {F}	1 ou 25%
{A,C}	2 ou 50%
Autres paires d'items	max 25%

- **Trouver** : **toutes** les règles avec un support et une confiance minimum donnés
  - Si support min. 50% et confiance min. 50%, alors

$$A \Rightarrow C \ [50\%, 66.6\%], \ C \Rightarrow A \ [50\%, 100\%]$$

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	1
Produit B	1	2	1	1	0
Produit C	1	1	1	0	0
Produit D	2	1	0	3	1
Produit E	1	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 :
  - Support(1)=20% ; Support(2)=Support(3)=40%
- Confiances :
  - Confiance(2) = 50% ; 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
fréquence	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

Description de la méthode

- Amélioration = confiance / fréq(résultat)
- Comparer le résultat de la prédiction en utilisant la règle avec la prédiction sans la règle
- Règle intéressante si Amélioration > 1

Règle	Confiance	F(résultat)	Amélioration
Si A et B alors C	0.20	40%	0.50
Si A et C alors B	0.25	42.5%	0.59
Si B et C alors A	0.33	45%	0.74

- Règle : Si A alors B ; support=25% ; confiance=55% ; Amélioration = 1.31  Meilleure règle

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 : $O(\mathbf{m}.2^{\mathbf{m}-1})$
  - Complexité de calcul : $O(\mathbf{n}.\mathbf{m}.2^{\mathbf{m}})$

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	5.10^7	$1.7\ 10^{11}$	$4.2\ 10^{14}$

- Conclusion de la règle restreinte à un sous-ensemble de l 'ensemble des articles vendus.
  - Exemple : articles nouvellement vendues.
- Création 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)

Support Minimum = 3

Triplets (3-itemsets)

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

Si tout sous-ensemble est considéré,

$C^6_1 + C^6_2 + C^6_3 = 41$

En considérant un seuil support min,

$6 + 6 + 2 = 14$

L'algorithme *Apriori* [Agrawal93]

- Deux étapes
  - Recherche des k-itemsets fréquents (support≥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≥ MINCONF
    - Exemple : ABCD fréquent
    - AB → CD est retenue si sa confiance ≥ MINCONF

Recherche des k-itemsets fréquents (1)

- Exemple
  - I = {A, B, C, D, E, F}
  - T = {AB, ABCD, ABD, ABDF, ACDE, BCDF}
  - MINSUP = 1/2
- Calcul de L1 (ensemble des 1-itemsets)
  - C1 = I = {A,B,C,D,E,F} // C1 : 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
  - L1 = {A, B, C, D}
- Calcul de L2 (ensemble des 2-itemsets)
  - C2 = L1xL1 = {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
  - L2 = {AB,AD, BD, CD}

Recherche des k-itemsets fréquents (2)

- Calcul de L3 (ensemble des 3-itemsets)
  - C3 = {ABD} (ABC $\notin$ C3 car AC $\notin$ L2)
  - s(ABD) = 3/6
  - L3 = {ABD}
- Calcul de L4 (ensemble des 4-itemsets)
  - C4 = ϕ
  - L4 = ϕ
- Calcul de L (ensembles des itemsets fréquents)
  - L = $\cup$Li = {A, B, C, D, AB, AD, BD, CD, ABD}

L'algorithme Apriori

L_1 = {1-itemsets fréquents};

for (k=2; $L_{k-1} \neq \phi$; k++) **do**

  C_k = apriori_gen(L_{k-1});

  forall instances $t \in T$ **do**

    C_t = subset(C_k,t);

    forall candidats $c \in C_t$ **do**

      c.count++;

  L_k = { $c \in C_k$ / c.count $\geq$ MINSUP }

$L = \cup_i L_i$;

La procédure *Apriori_gen*

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

insert into C_k;

select $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$

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

where $p.item_1=q.item_1, \ldots, p.item_{k-2}=q.item_{k-2}$

$, p.item_{k-1} < q.item_{k-1}$

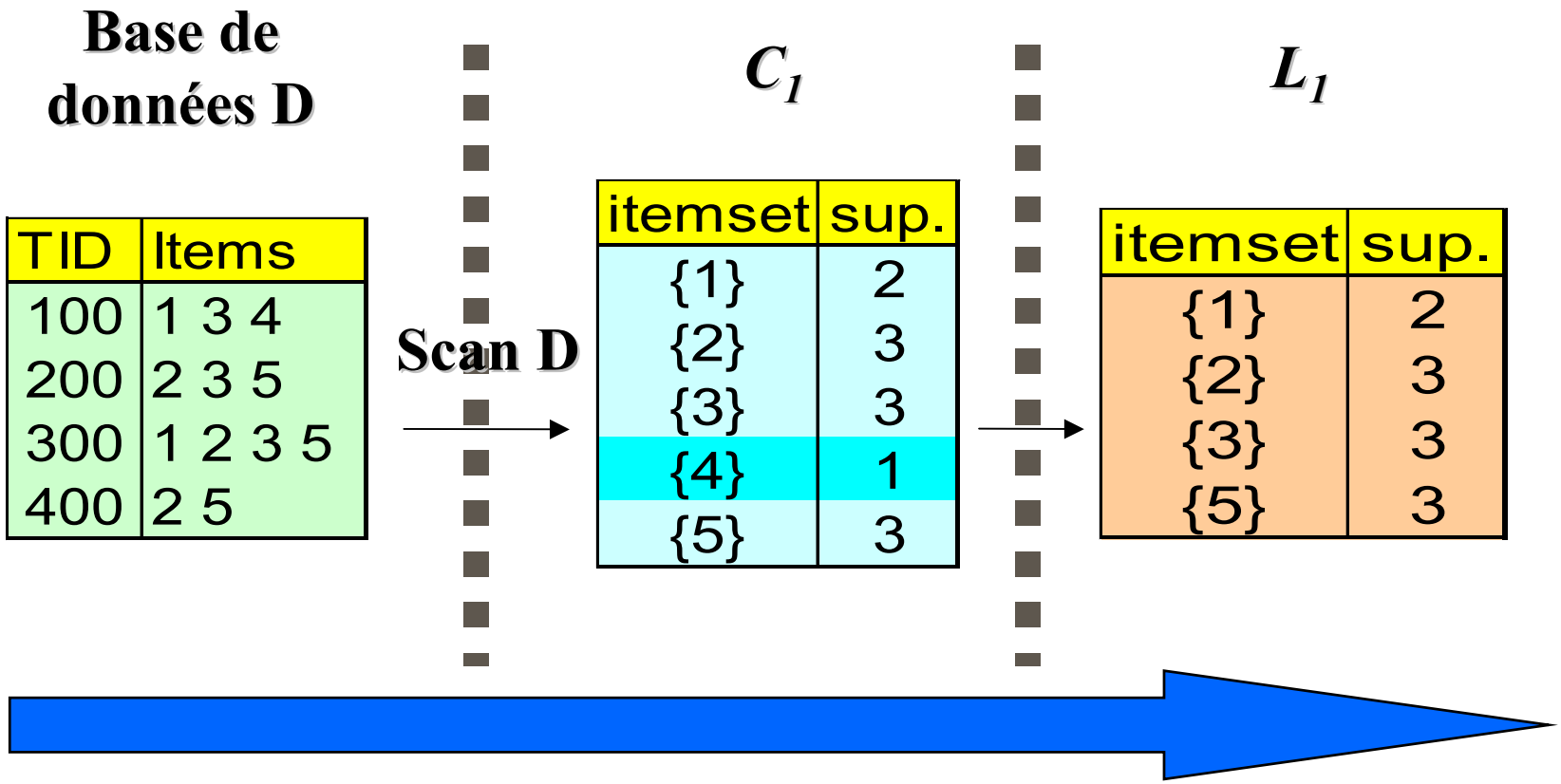
forall itemsets $c \in C_k$ **do**

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

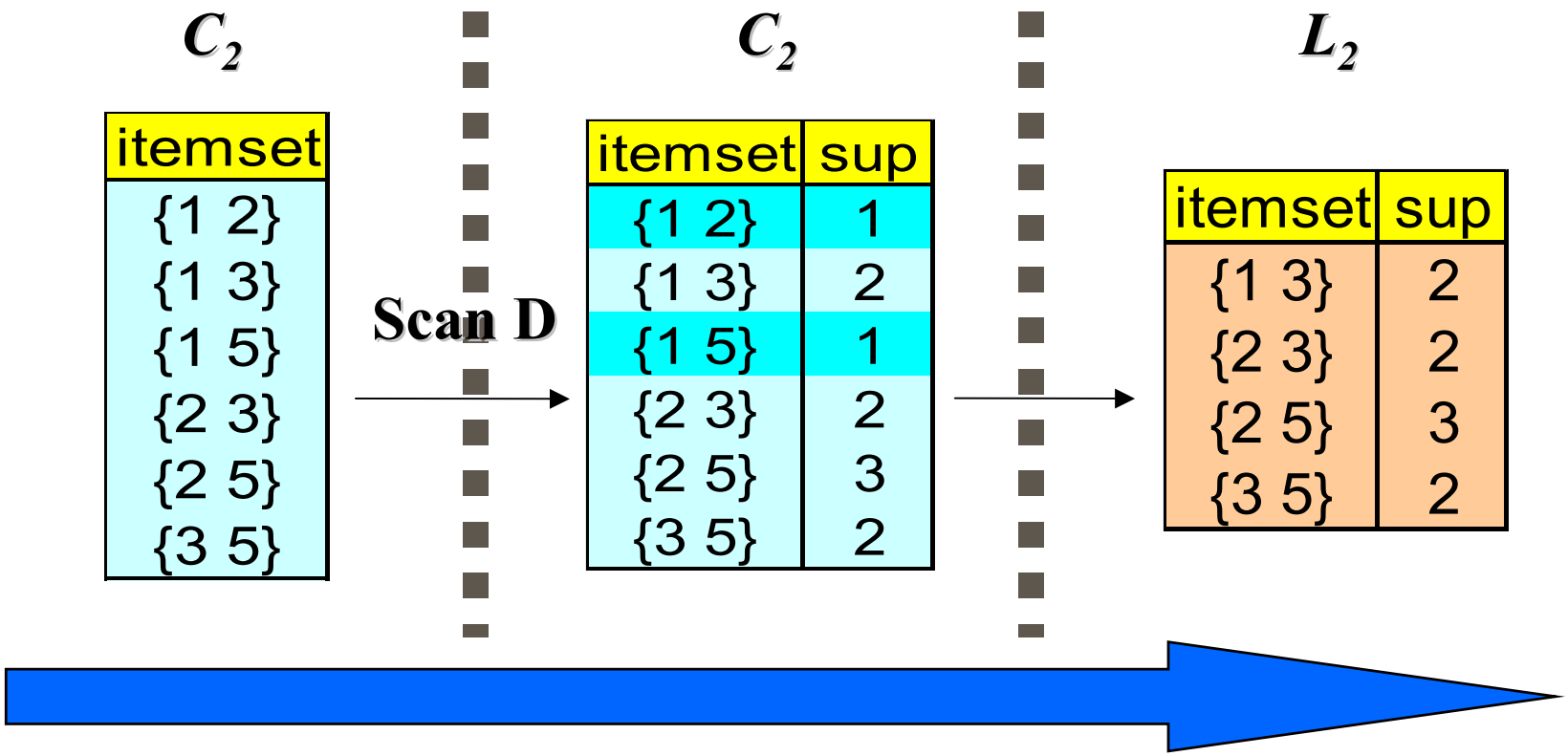
if $s \notin L_{k-1}$ **then**

delete c **from** C_k;

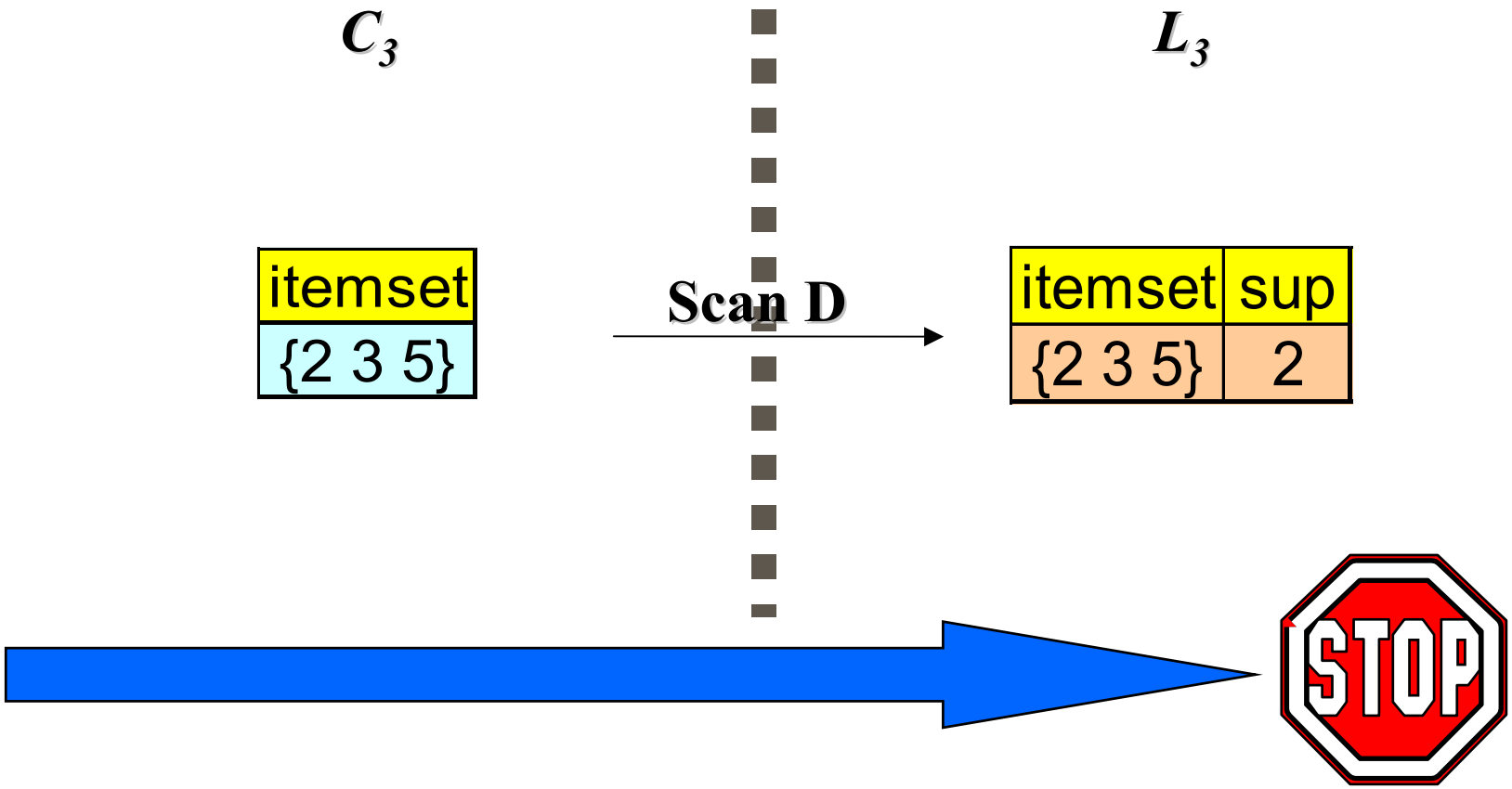
Apriori - Exemple



Apriori - Exemple

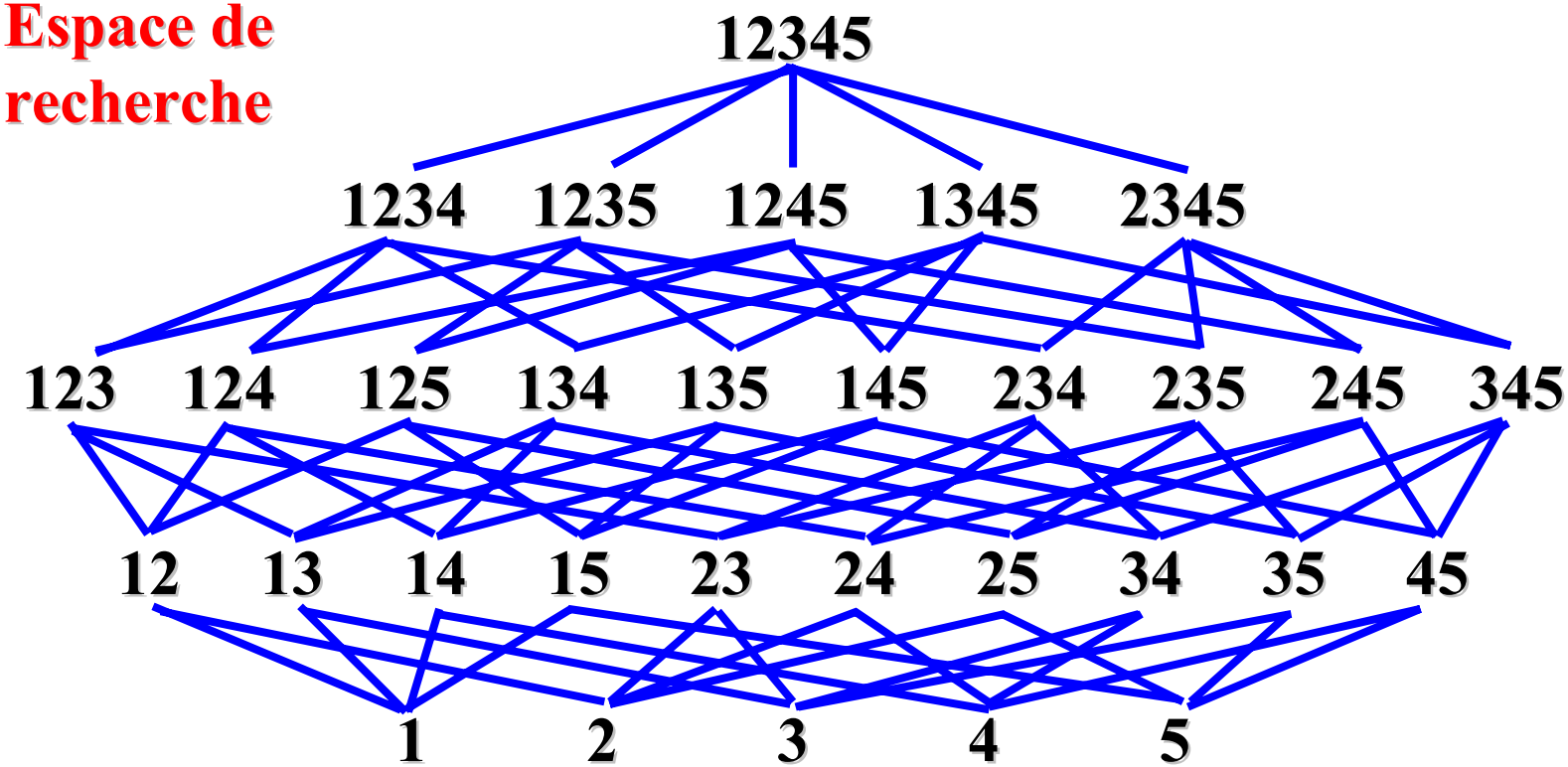


Apriori - Exemple



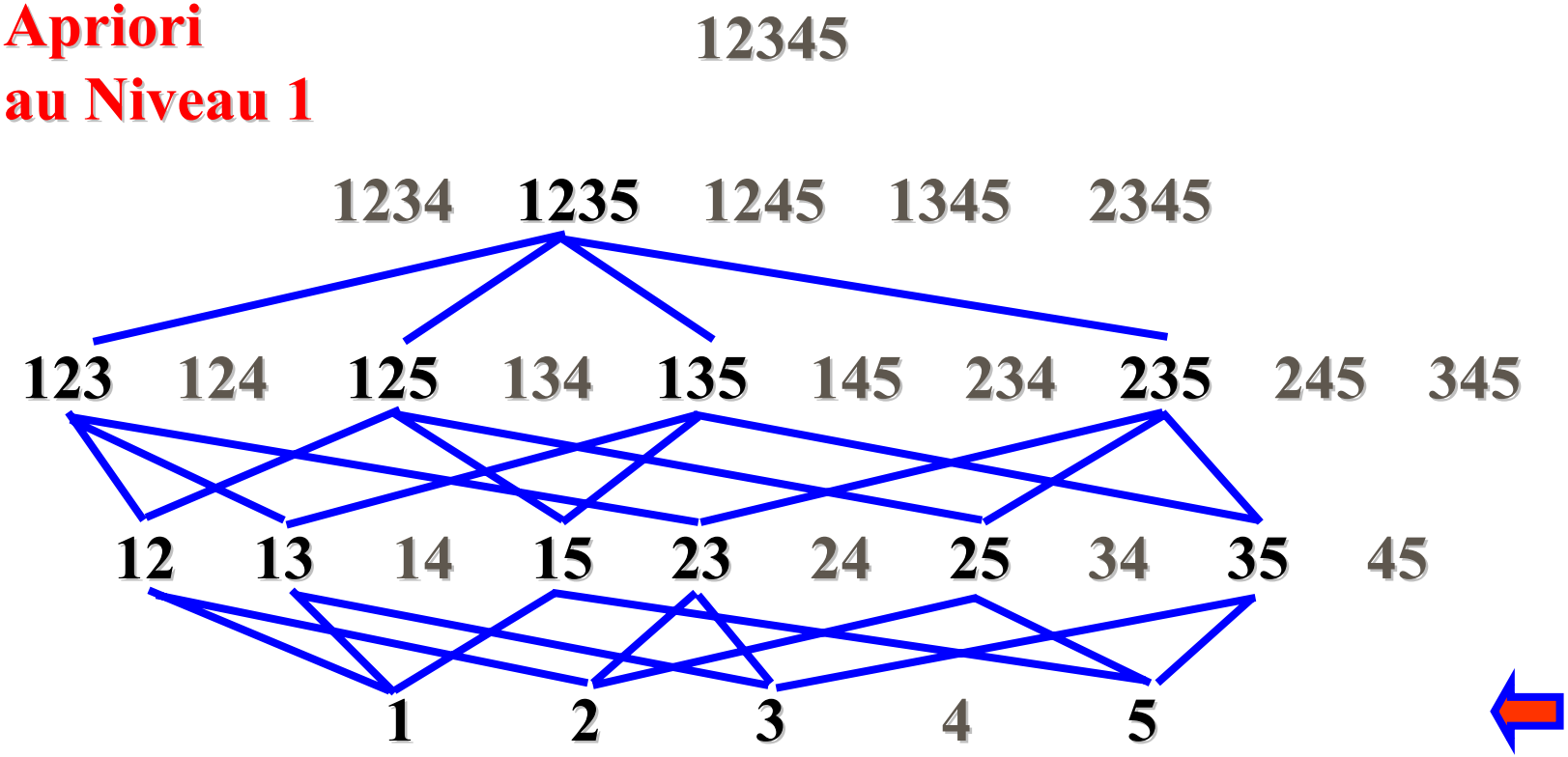
Apriori - Exemple

Espace de recherche



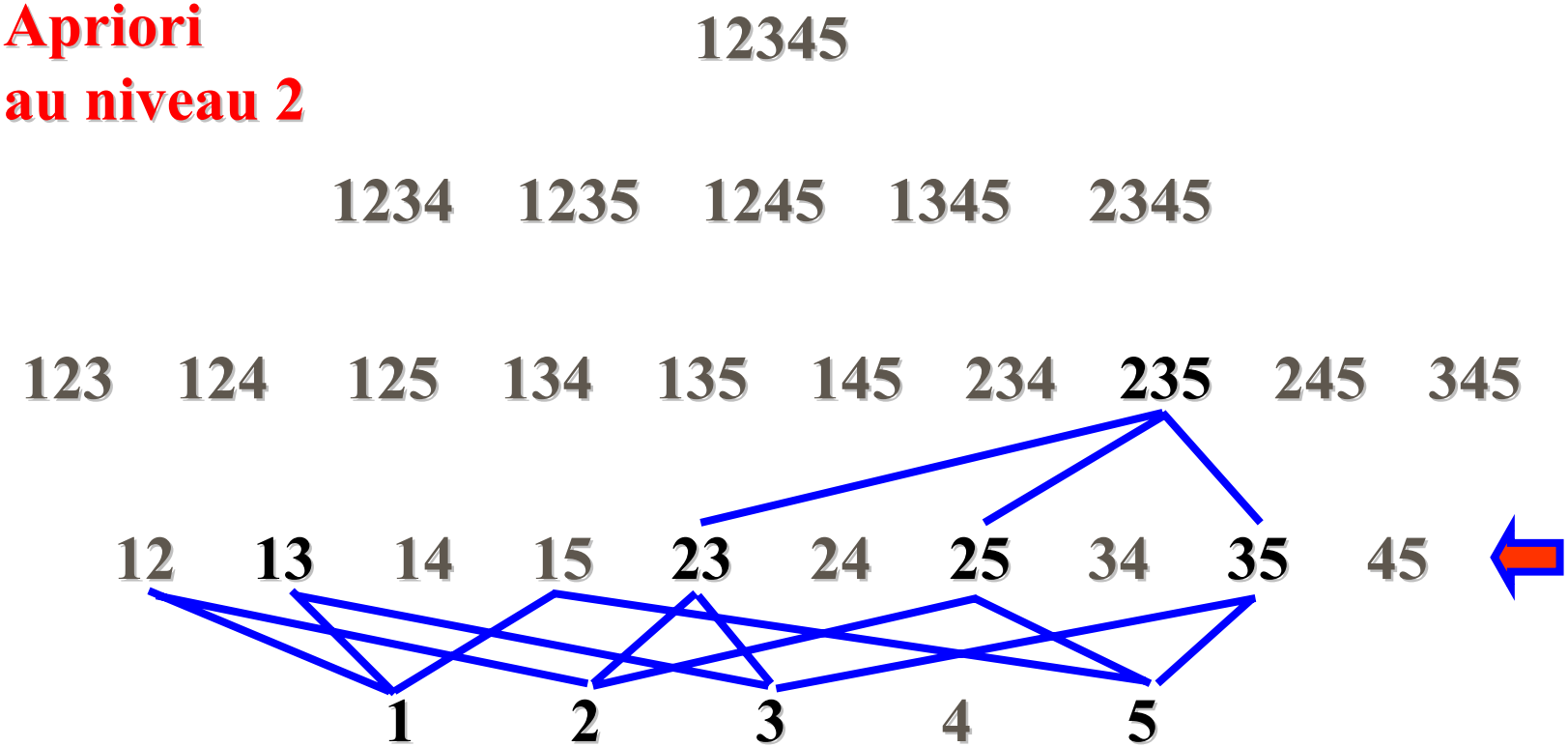
Apriori - Exemple

Apriori
au Niveau 1



Apriori - Exemple

Apriori
au niveau 2



Génération des règles à partir des itemsets

- **Pseudo-code :**
  - **pour** chaque itemset fréquent *l*
    générer tous les sous-itemsets non vides *s* de *l*
  - **pour** chaque sous-itemset non vide *s* de *l*
    produire la règle "$s \Rightarrow (l-s)$" si *support(l)/support(s)* ≥ *min_conf*", où *min_conf* est la confiance minimale

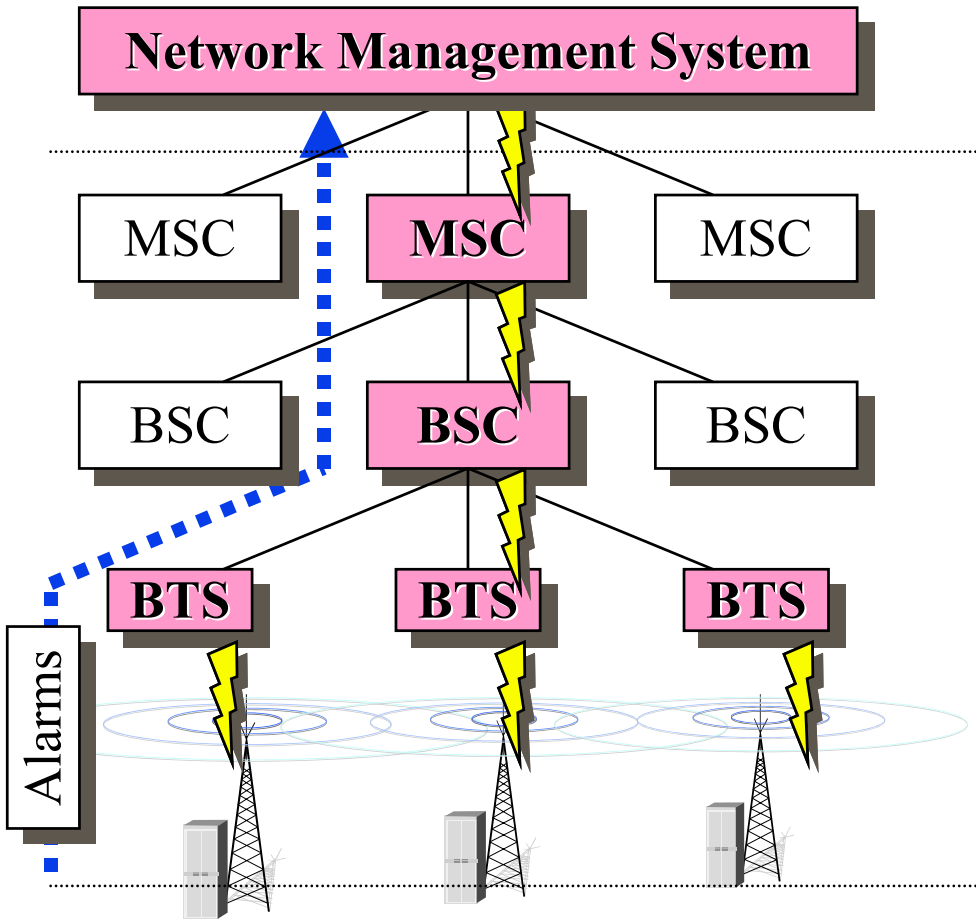
  - **Exemple : itemset fréquent l = {abc},**
  - **Sous-itemsets s = {a, b, c, ab, ac, bc)**

    - a ⇒ bc, b ⇒ ac, c ⇒ ab
    - ab ⇒ c, ac ⇒ b, bc ⇒ 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)

Exemple de performances



Réseau switché

Réseau d'accés

Réseau cellulaire

MSC Mobile station controller

BSC Base station controller

BTS Base station transceiver

Exemple de performances

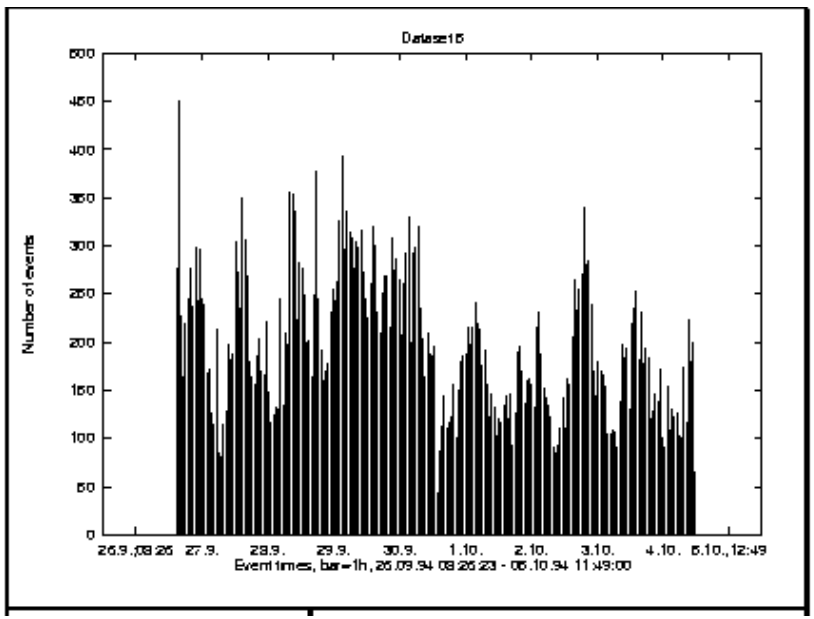
- **Données télécom contenant des alarmes :**
- `1234 EL1 PCM 940926082623 A1 ALARMTEXT..`

  Alarm number | Alarming network element | Alarm type | Date, time | Alarm severity class

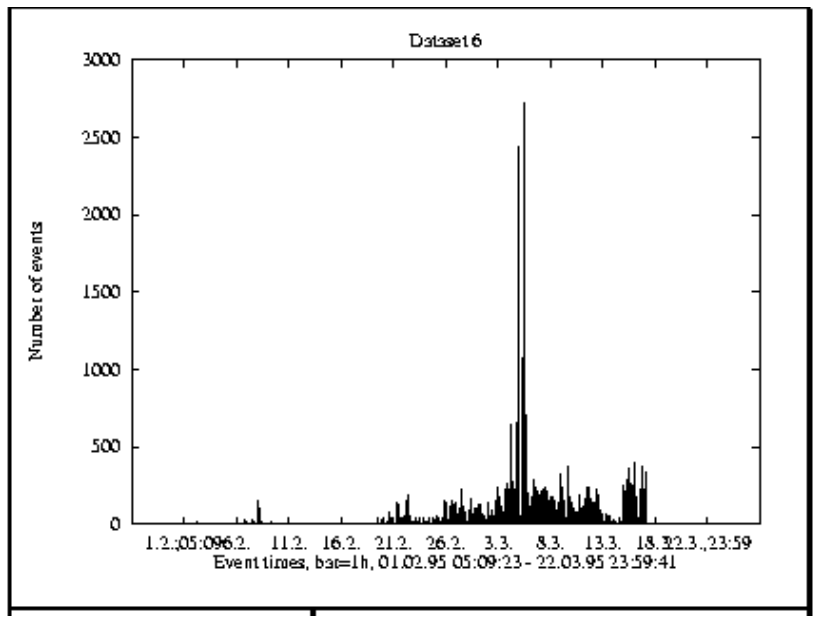
- **Exemple de données 1 :**
  - 43 478 alarmes (26.9.94 - 5.10.94; ~ 10 jours)
  - 2 234 différent types d'alarmes, 23 attributs, 5503 différentes valeurs
- **Exemple de données 2 :**
  - 73 679 alarmes (1.2.95 - 22.3.95; ~ 7 semaines)
  - 287 différent types d'alarmes, 19 attributs, 3411 différentes valeurs

Exemple de performances

Ensemble données 1 (~10 jours)



Ensemble données 2 (~7 semaines)



Exemple de règles :

alarm_number=1234, alarm_type=PCM ⇒ alarm_severity=A1 [2%,45%]

Exemple de performances

- **Exemple de résultats pour les données 1 :**
  - Seuil de fréquence : 0.1
  - Itemsets candidats : 109 719 Temps: 12.02 s
  - Itemsets fréquents : 79 311 Temps: 64 855.73 s
  - Règles : 3 750 000 Temps: 860.60 s

- **Exemple de résultats pour les données 2 :**
  - Seuil de fréquence : 0.1
  - Itemsets candidats : 43 600 Temps: 1.70 s
  - Itemsets fréquents : 13 321 Temps: 10 478.93 s
  - Règles : 509 075 Temps: 143.35 s

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., $\{a_1, a_2, \ldots, a_{100}\}$, 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 ...**

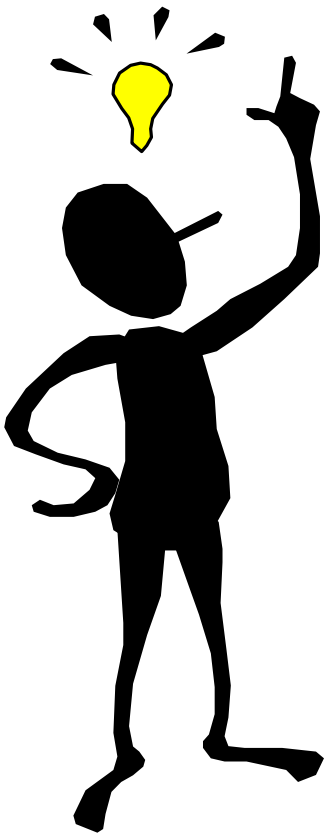
Apriori – Réduction de la complexité

- **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, décroitre 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 = terme$_1$ & terme$_2$ & … & terme$_n$
    - P = terme$_{n+1}$
    - terme$_i$ = <attribut$_j$, *op*, valeur> ou <attribut$_j$, *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.

Classification généralisée par Algorithmes Génétiques

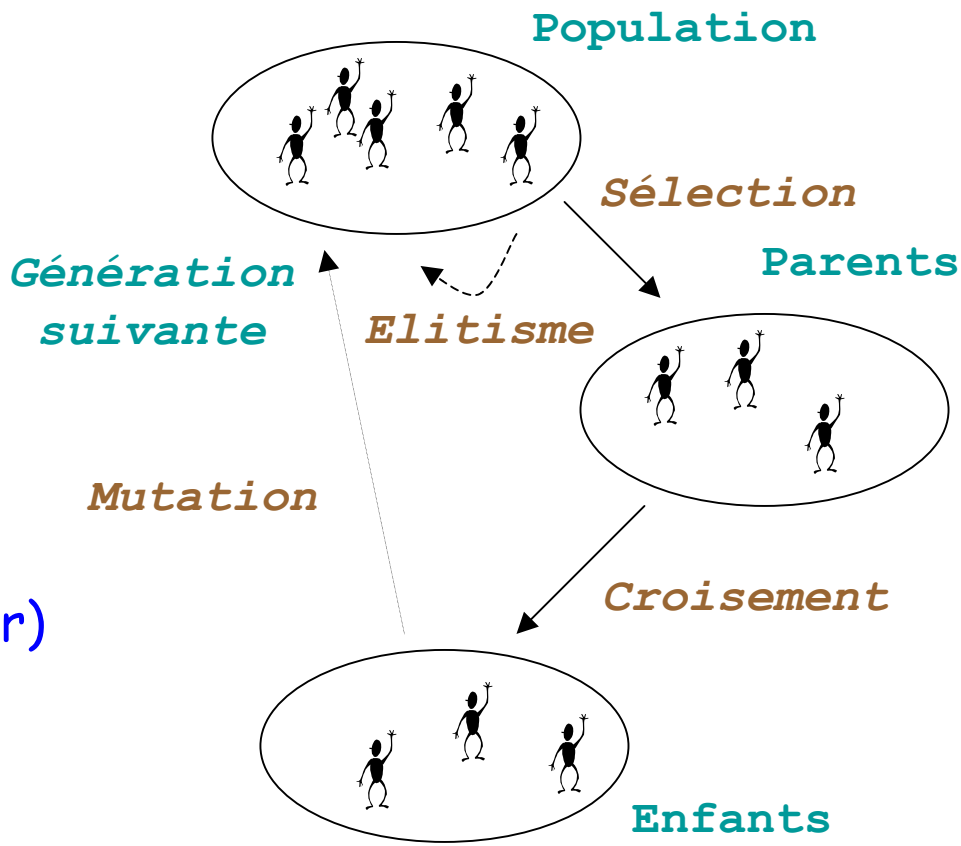
Problématique

- Découvrir dans une large BD **quelques** petites règles **intéressantes** «Si C Alors P»
  - C = $terme_1$ & $terme_2$ ... & $terme_n$ ($n \leq MAXTERM$)
    - $terme_{i=1..n}$ ≡ <attribut=valeur> / valeur est énumératif
  - P = terme ≡ <attribut but=valeur>
    - attribut but ∈ GoalsSet (défini par l'utilisateur)

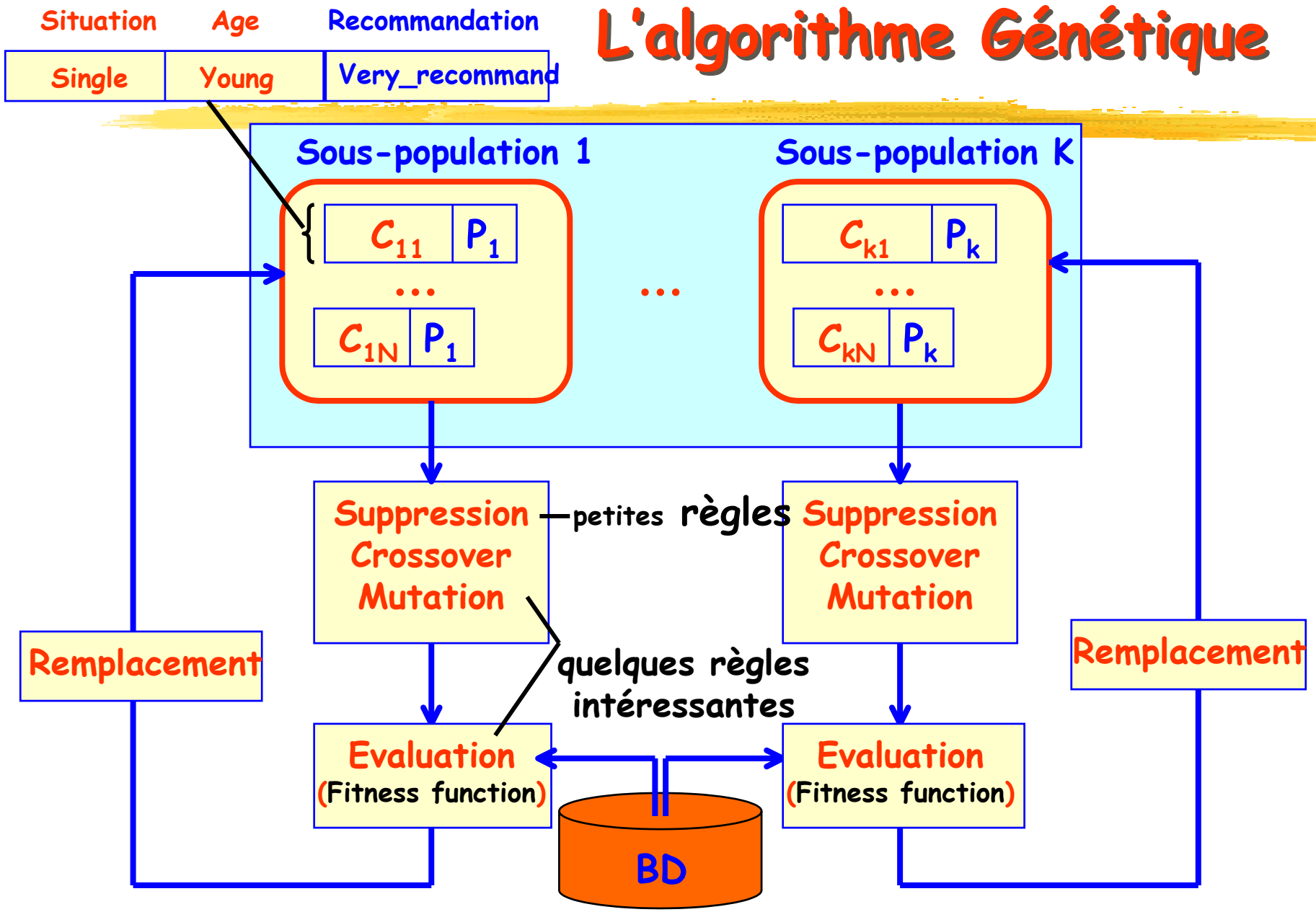
- Exemple : SI (Situation=Single) and (Age=Young) THEN (Recommandation=Very_recommand)

Les algorithmes génétiques

- J. Holland (1975)
- Principes
  - Codage des solutions
  - Opérateurs
    - Sélection
    - Croisement (Crossover)
    - Mutation



L'algorithme Génétique



Fitness (Intérêt d'une règle)

$$G(Rule) = \frac{|C|}{N} b.\log(\frac{b}{a})$$ [Wang et al. 98]

$$a = \frac{|P|}{N}, b = \frac{|C \ \& \ P|}{|C|}$$

$$F(Rule) = \frac{\omega_1 . G(Rule) + \omega_2 . \frac{\eta_{pu}}{\eta_t}}{\omega_1 + \omega_2}$$ [Freitas 99]

Opérateurs génétiques : Crossover (1)

- Deux parents P_1 et P_2 ont un ou plusieurs attributs commun(s) dans leurs parties *C*
  - Sélection aléatoire d'un terme
  - Permutation de leurs valeurs
- Exemple :
  - P1 : (*Marital_status=married)* $\wedge$ *(Gender=male)*
  - P2 : (*Marital_status= single)* $\wedge$ (*Salary=high)*

  - Enfant1 : (*Marital_status=single)* $\wedge$ *(Gender=male).*
  - Enfant2 : (*Marital_status=married)* $\wedge$ (*Salary=high).*

Opérateurs génétiques : Crossover (2)

- P_1,P_2 n'ont aucun attribut commun dans *C*
  - Sélection aléatoire d'un terme dans P_1
  - Insertion dans P_2
    - Proba = (MAXTERM - K)/MAXTERM
    - K: Nombre de termes dans la partie *C* de P_2
  - *Vice versa*
- Exemple :
  - P1 : (*Marital_status=married)* $\wedge$ *(Gender=male)*
  - P2 : (*Age = young)* $\wedge$ (*Salary=high)*
  - E1 : (*Marital_status=married)* $\wedge$ *(Gender=male)* $\wedge$ **(*Age=young)***
  - E2 : (*Marital_status=married)* $\wedge$ (*Salary=high)* $\wedge$ ***(Gender=male)***

Opérateurs génétiques : Mutation (1)

- Deux types de mutation
  - Mutation d'attributs
  - Mutation de valeurs d'attributs
- Le type de mutation est choisi aléatoirement
- Mutation d'attribut
  - Remplacement d'un attribut par un autre (choix aléatoire)
  - La valeur du nouvel attribut est choisie aléatoirement
  - Exemple :
    - P : $(Marital\_status=married) \wedge (Gender=male)$
    - Enfant : $(Age=young) \wedge (Gender=male)$

Opérateurs génétiques : Mutation (2)

- Mutation de valeur d'attribut
  - Sélection d'un attribut aléatoirement
  - Remplacement de sa valeur par une autre choisie aléatoirement

  - Exemple :
  - Parent : (*Marital_status=married)* $\wedge$ *(Gender=male)*
  - Enfant : (*Marital_status=single)* $\wedge$ *(Gender=male)*

Opérateurs génétiques : Suppression

- Suppression de termes
  - But : règles plus faciles à comprendre (petites)
  - Suppression d'un terme choisi aléatoirement avec une probabilité proportionnelle à sa longueur

  - Exemple :
  - P : (*Marital_status=married*) $\wedge$ (*Gender=male*) $\wedge$ (*Age=young*)
  - E : (*Marital_status=married*) $\wedge$ (*Gender=male*)

Application

- BD : Nursery school
  - From *http://www.ics.uci.edu/AI/ML/Machine-Learning.html*
  - 12960 data instances with 9 attributes

	Attribute name	Attribute values
1	Parents	Usual, pretentious, great_pret
2	Has_nurs	Proper, less_proper, improper, critical, very_crit
3	Form	Complete, completed, incomplete, foster
4	Children	1, 2, 3, more
5	Housing	Convenient, less_conv, critical
6	**Finance**	Convenient, inconv
7	**Social**	Nonprob, slightly_prob, problematic
8	Health	Recommended, priority, not_recom
9	**Recommendation**	Recommend, priority, not_recom, very_recom

- Hardware platform
  - SGI/IRIX (100MHz R4600, 32MB RAM, 549MB disque)

- Paramètres de l'AG
  - 3 attributs buts
  - MAXTERM=5
  - 150 individus /3 sous-populations

Evaluation expérimentale (1)

- Publication
  - N. Melab and E-G. Talbi. **A Parallel Genetic Algorithm for Rule Mining**. IEEE Intl. Workshop on Bio-Inspired Solutions to Parallel Processing Problems (BioSP3), San Francisco, USA, Apr. 23, 2001.
- Evaluation de l'AG
  - Qualité des règles extraites
  - Paramètres mesurés :
    - Validité : facteur de confiance des règles

$$FC = \frac{|C\&P|}{|C|}$$

Evaluation expérimentale (2)

Règle	|C|	|P|	|C&P|	FC_{Train}	FC_{Test}
R1	18	1296	9	0.500000	0.500000
R2	6	1296	3	0.500000	0.500000
R3	288	196	124	0.430556	0.000000
R4	18	864	18	1.000000	1.000000
R5	18	864	18	1.000000	1.000000
R6	54	864	18	0.333333	0.333333
R7	57	864	18	0.333333	0.333333
R8	162	864	54	0.333333	0.333333
Moyenne				0.552500	0.4987500

- FC mesurés
  - Sur les données d'apprentissage (20%) : **FC_{train}**
  - Sur les données de test (80%) : **Fctest**
- Exemple : R4 : SI *((parents=usual) && (health=not_recomm))* ALORS *(recommandation=not_recomm)*

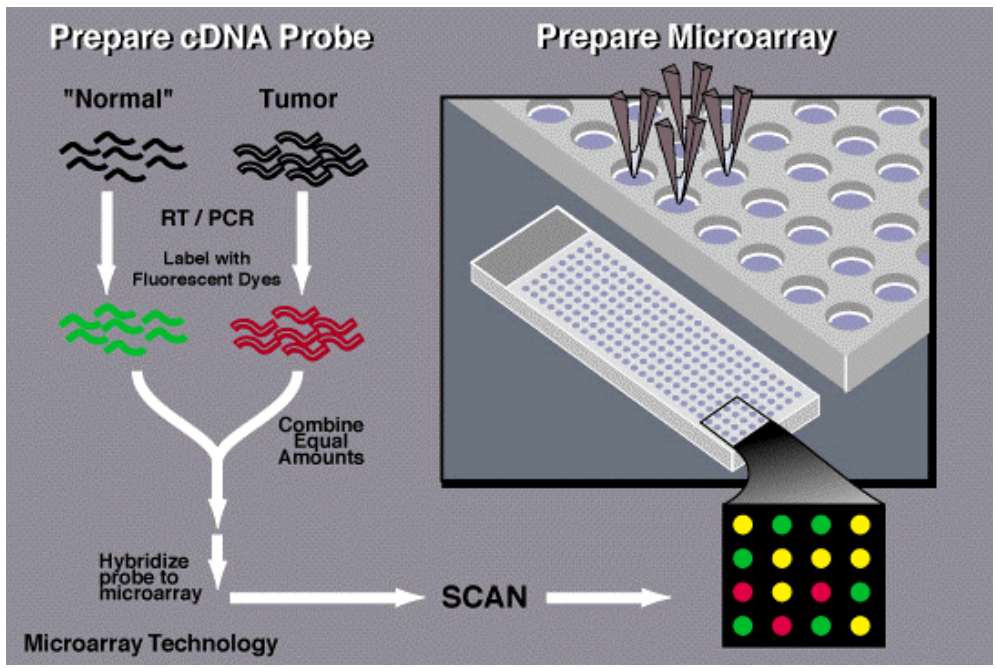
Technique "Puces à ADN"

- Avantage principal des techniques "Puces à ADN"
  - Permet l'analyse simultanée d'expressions de milliers de gènes dans une seule expérience

- Processus "Puces à AND"
  - Arrayer
  - Expérience : Hybridation
  - Capture des images résultats
  - Analyse

Analyse de l'expression de gènes : Technologie Puces à ADN

- Des robots alignent les ESTs (Expressed Sequence Tags) sur les lames de microscopes
- cellules mRNA marquées par des tags fluorescents
- Liaison mRNA - cDNA exprimée (fluorescence) indique que le gène est actif



Ressources

Image Analysis		
BioDiscovery	http://www.biodiscovery.com/software.html	BioDiscovery's ImaGene Image Analysis Software
ScanAlyze	http://bronzino.stanford.edu/ScanAlyze	Brown Lab's Image Analysis software
Microarray Data Warehousing & Analysis		
Affymetrix	http://www.affymetrix.com/products/lims/lims.html	GeneChip LIMS data warehouse
Brown Lab, Stanford University	http://cmgm.stanford.edu/pbrown/explore/	Searchable database of published yeast microarray data
MicroArray Project, NIH	http://www.nhgri.nih.gov/DIR/LCG/15K/HTML/dbase.html	Database schema and software tools for analysis of high-throughput gene expression data
Rosetta Inpharmatics	http://www.rosetta.org/	Resolver data warehouse & analysis software
Silicon Genetics	http://www.sigenetics.com/GeneSpring/Overview.htm	GeneSpring data warehouse & analysis software

Objectif de "Microarray Mining"

Analyse des expressions de gènes sous différentes conditions

gene \ test	A	B	C	…	…	….
1	0.6	0.4	0.2	…		
2	0.2	0.9	0.8	…		
3	0	0	0.3	…		
4	0.7	0.5	0.2	…		
..	..	..	..	…		
..	..	..	..	…		
1000	0.3	0.8	0.7	…		

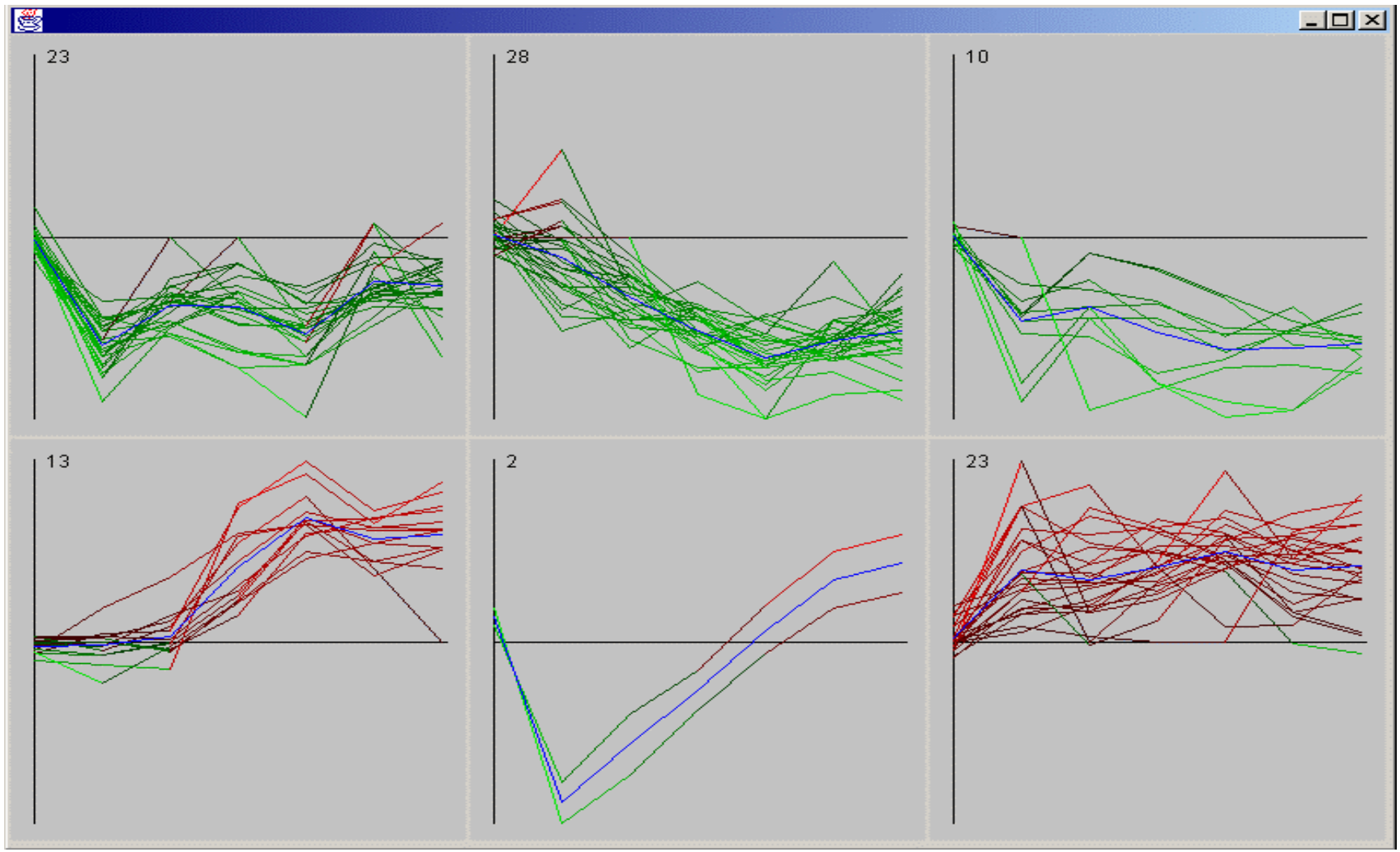
Objectif du "Microarray Mining"

Analyse des expressions de gènes sous différentes conditions

gène \ test	A	B	C	…	…	….
1	0.6	0.4	0.2	…		
2	0.2	0.9	0.8	…		
3	0	0	0.3	…		
4	0.7	0.5	0.2	…		
..	..	..	..	…		
..	..	..	..	…		
1000	0.3	0.8	0.7	…		

Clustering de gènes

Genes participating in the same pathway are most likely expression at same time.



Règles d'association

Gene1, Gene2, Gene3, Gene4, Gene5.

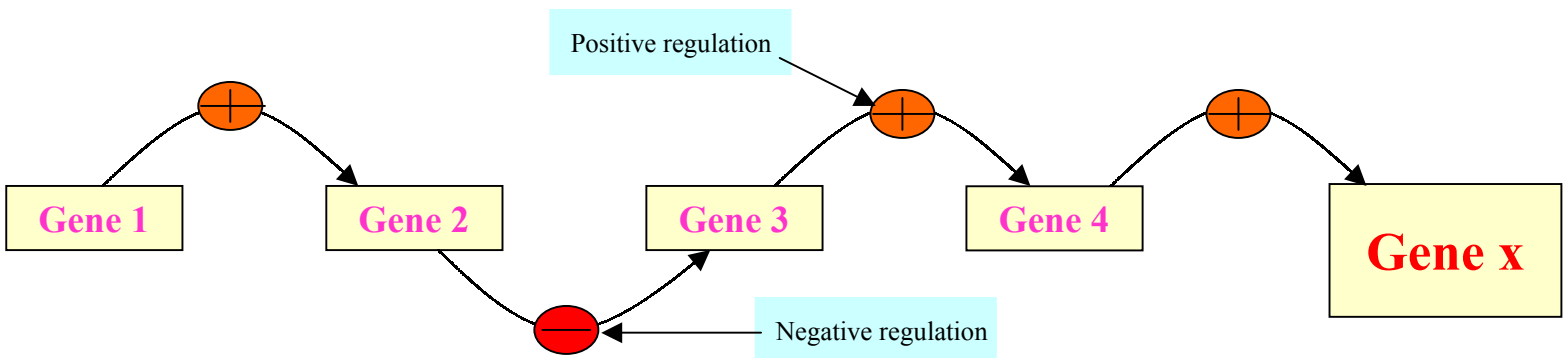
Gène représentant la conséquence ?

Chaque condition (microarray) est une instance.

Gènes représentent les itemsets.

Règles d'association avec confiance élevée (100%?)

Gènes cibles = conséquence des règles



Expérimentations

- Ensemble de données
  - Source : Lawrence Berkeley National Lab (LBNL) Michael Eisen's Lab http://rana.lbl.gov/EisenData.htm
  - Données d'expression Microarray de "yeast saccharomyces cerevisiae", contenant 6221 gènes sous 80 conditions

Règles d'association – Résumé

- Probablement la contribution la plus significative de la communauté KDD
- Méthodes de recherche de règles :
  - A-priori
  - Algorithmes génétiques
- Plusieurs articles 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, …

Règles d'association - Références

- R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93, 207-216, Washington, D.C.
- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97, 265-276, Tucson, Arizona.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94, 401-408, Gaithersburg, Maryland.
- H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94, 181-192, Seattle, WA, July 1994.
- G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W. J. Frawley, editors, Knowledge Discovery in Databases, 229-238. AAAI/MIT Press, 1991.

Outils pour le Data Mining

Comment Choisir un outil ?

- **Systèmes commerciaux de data mining possèdent peu de propriétés communes :**
  - Différentes méthodologies et fonctionalités 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**

Comment Choisir un outil ?

- **Types de donnés :** relationnel, transactionnel, texte, séquences temporelles, spaciales ?
- **Issues systèmes**
  - Support systèmes d'exploitation ?
  - Architecture client/serveur ?
  - Fournit une interface Web et permet des données XML en entrée et/ou en sortie ?
- **Sources des données :**
  - Fichiers texte ASCII, sources de données relationnels multiples, …
  - Support ODBC (OLE DB, JDBC) ?

Comment Choisir un outil ?

- **Fonctionalités et méthodologies**
  - une vs. plusieurs fonctions de data mining
  - une vs. plusieurs méthodes par fonction
- **Couplage avec les systèmes de gestion de base de données et les entropots de données**

- **Outils de visualization :** visualisation des données, visualisation des résultats obtenus, visualisation du processus, visualisation interactive (split attribut, …), etc.

Comment Choisir un outil ?

- **Extensibilité (Scalability)**
  - instances (Taille de la base de données)
  - attributs (dimension de la base)
  - Extensibilité en terme d'attributs est plus difficile à assurer que l'extensibilité en terme d'instances

- **Langage de requête et interface graphique (IHM)**
  - easy-to-use et qualité de l'interface
  - data mining interactif

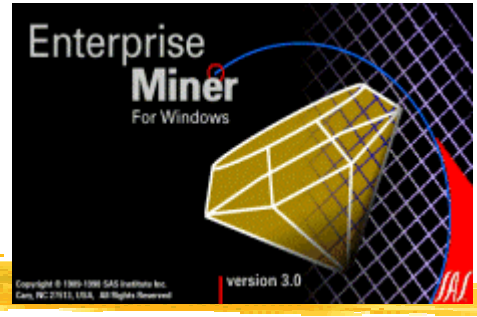
Exemple d'outils (1)

- 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
- Darwin de Thinking Machines
  - Trois techniques : réseaux de neurones, arbres de décision et régression.
  - Client-Serveur

Exemples d'outils (2)

- MineSet de Silicon Graphics
  - Fonctionnalités interactives et graphiques
  - Techniques sous-jacentes : classification, segmentation, recherche de règles d'association.
- Outils/librairies libres
  - SIPINA
  - WEKA
- Data-Miner Software Kit (DMSK)
  - Kit de programmes : méthodes statistiques, segmentation, groupage, réseaux de neurones, etc.
  - Il existe une version en java
- etc.

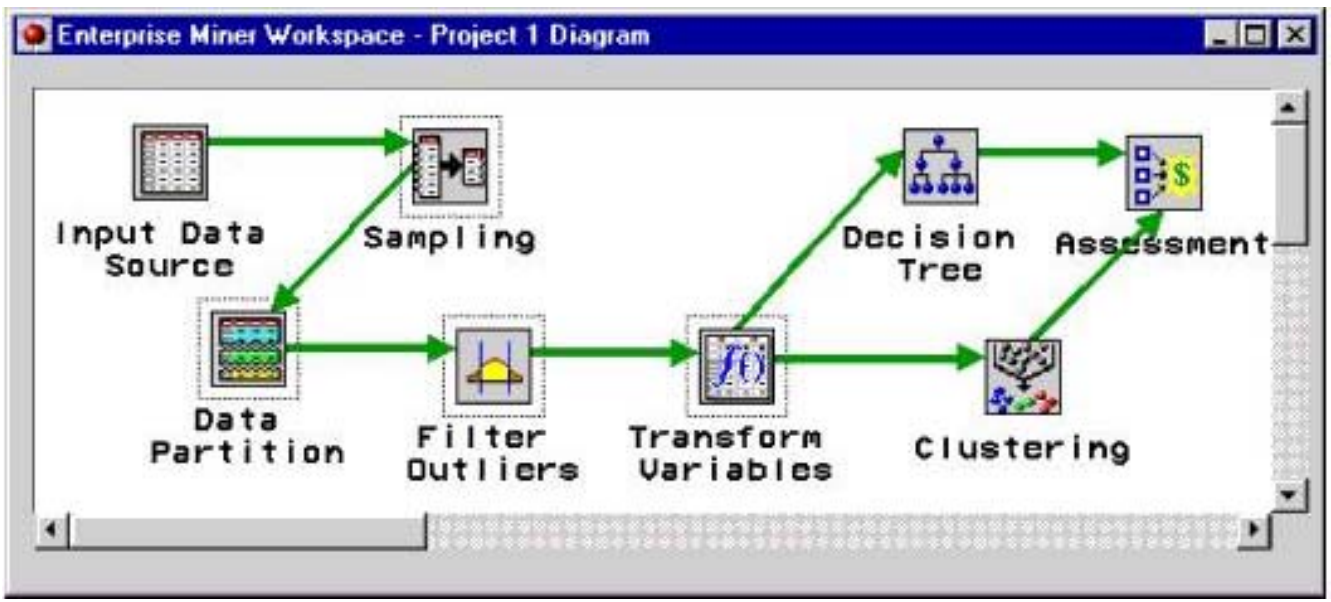
SAS Entreprise Miner (1)



- Société : SAS Institute Inc.
- Création : Mai 1998
- Plate-formes : Windows NT & Unix
- Utilisation
  - Réduction des coûts
  - Maîtrise des risques
  - Fidélisation
  - Prospection
- Outils de data warehouse

SAS Entreprise Miner (2)

- Interface graphique (icônes)
- Construction d'un diagramme



SAS Entreprise Miner (3)

- Deux types d'utilisateurs
  - Spécialistes en statistiques
  - Spécialistes métiers (chef de projet, études…)
- Techniques implémentées
  - Arbres de décision
  - Régression
  - Réseaux de neurones

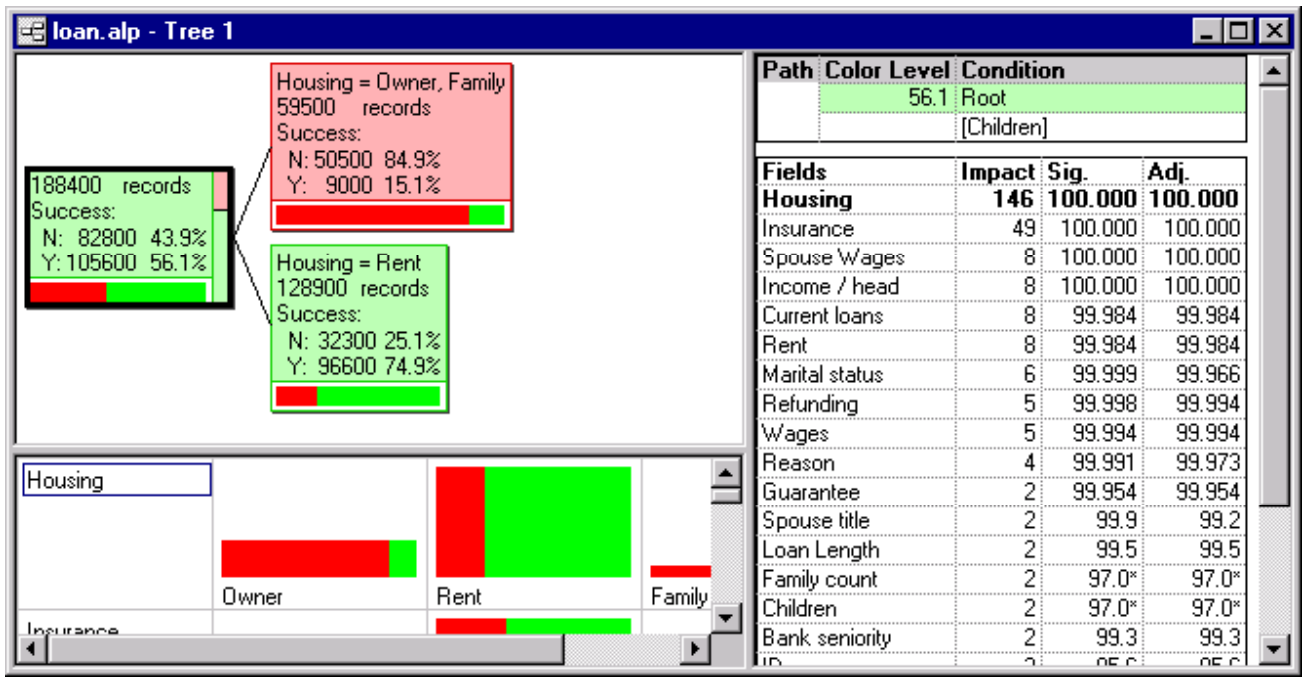
Alice (1)

- Société : ISoft
- Création : 1988
- Plate-formes : Windows 95/98/NT/2000, TSE, Metaframe

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

Alice (2)

- Interface graphique (tools)



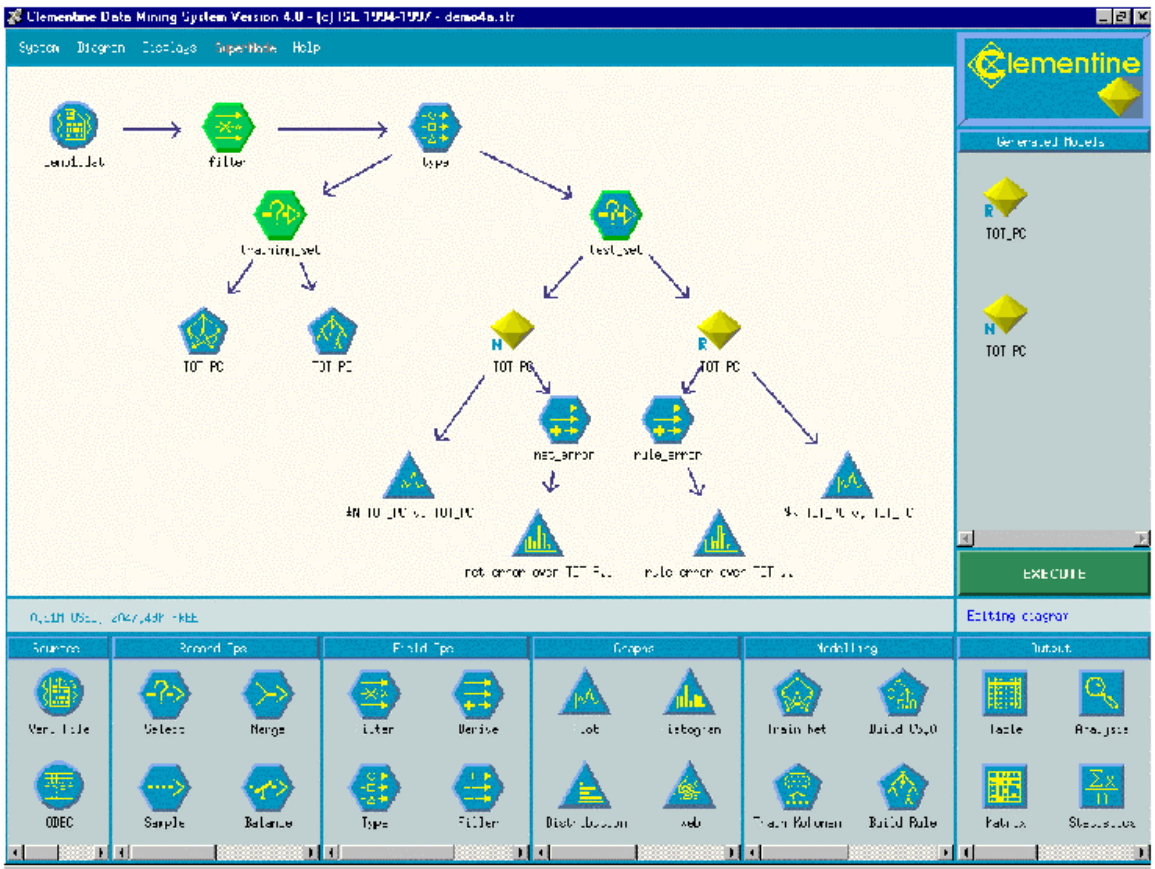
- Type d'utilisateur : responsables operationnels

Clementine (1)

- Société : ISL (*Integral Solutions Limited*)
- Création : 1994
- Plate-formes : Windows NT, Unix
- Utilisation
  - Prévision de parts de marché
  - Détection de fraudes
  - Segmentation de marché
  - Implantation de points de vente …
- Environnement intégré : #Types d'utilisateurs
  - Gens du métier (pas forcement des informaticiens)
  - Développeurs / End users

Clementine (2)

- Interface simple, puissante et complète
  ⇨ interface conviviale



Clementine (3)

- Techniques :
  - Arbres de décision
  - Induction de règles
  - Réseaux de neurones
  - Méthodes statistiques

Forecast Pro (1)

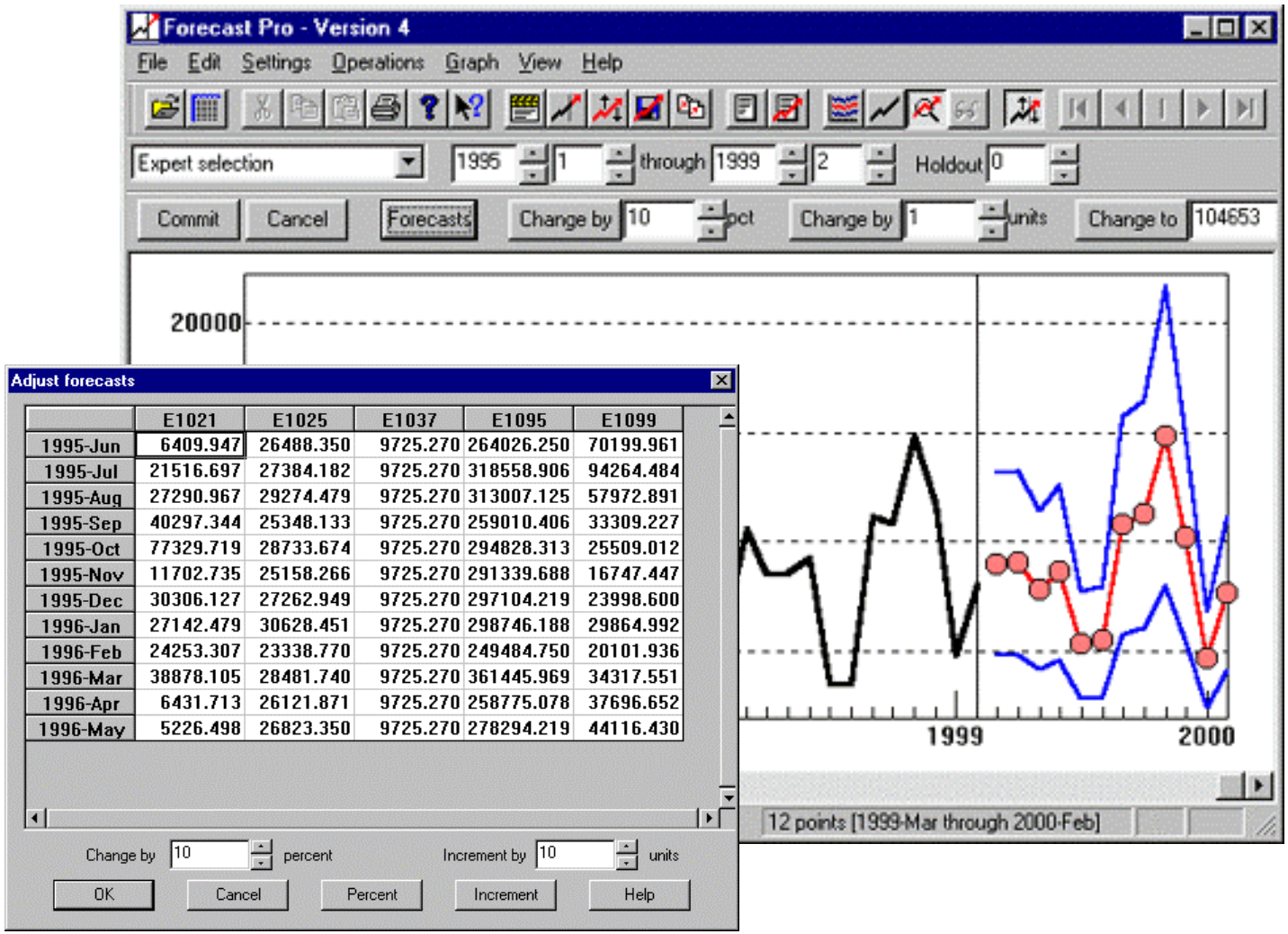
ForecastPRO®

- Société : Business Forecast Systems
- Création : 1997
- Plate-formes : Windows 95, NT
- Utilisation
  - Tous domaines activités et secteurs
  - Notamment la prévision (5 types différents)
- Outil d'analyse incomparable
- Le plus utilisé dans le monde

Forecast Pro (2)

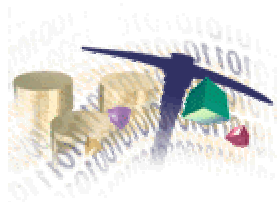
- Types d'utilisateurs : PME/PMI, administrations, consultants, universitaires, chefs de projets,…
- Facilité d'utilisation (connaissances en statistiques non requises)
- Vaste palette de choix graphiques
  - Valeurs observées, prévisions, valeurs calculées sur l'historique, intervalles de confiance, diagnostics (erreurs)

Forecast Pro (3)



	E1021	E1025	E1037	E1095	E1099
1995-Jun	6409.947	26488.350	9725.270	264026.250	70199.961
1995-Jul	21516.697	27384.182	9725.270	318558.906	94264.484
1995-Aug	27290.967	29274.479	9725.270	313007.125	57972.891
1995-Sep	40297.344	25348.133	9725.270	259010.406	33309.227
1995-Oct	77329.719	28733.674	9725.270	294828.313	25509.012
1995-Nov	11702.735	25158.266	9725.270	291339.688	16747.447
1995-Dec	30306.127	27262.949	9725.270	297104.219	23998.600
1996-Jan	27142.479	30628.451	9725.270	298746.188	29864.992
1996-Feb	24253.307	23338.770	9725.270	249484.750	20101.936
1996-Mar	38878.105	28481.740	9725.270	361445.969	34317.551
1996-Apr	6431.713	26121.871	9725.270	258775.078	37696.652
1996-May	5226.498	26823.350	9725.270	278294.219	44116.430

Intelligent Miner (1)



- 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 (2)

- Deux versions
  - Intelligent Miner for Data (IMD)
  - Intelligent Miner for Text (IMT)
- Types d'utilisateurs : spécialistes ou professionnels expérimentés
- Parallel Intelligent Miner

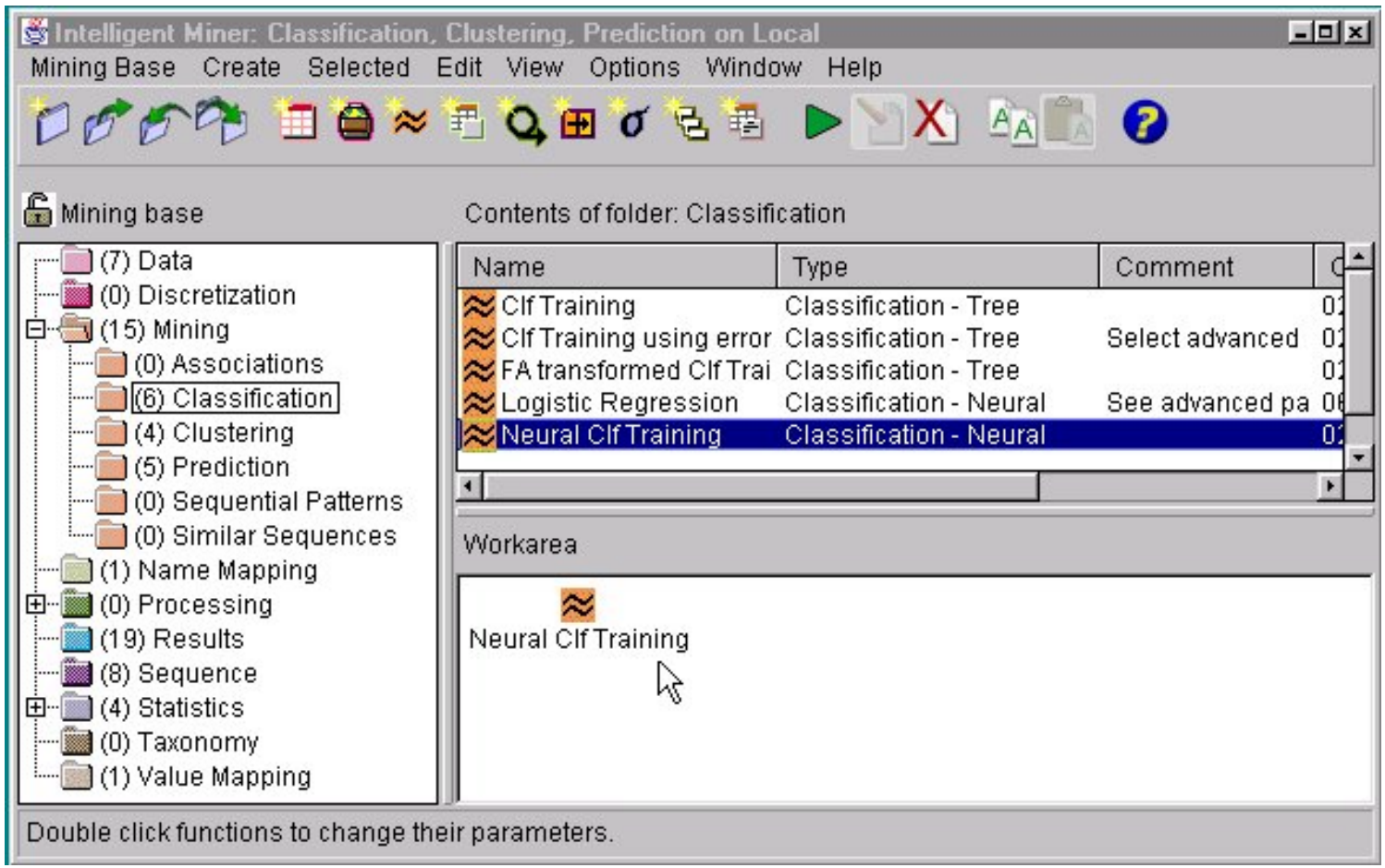
Intelligent Miner (3)

- 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 (4)

- 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 (5)

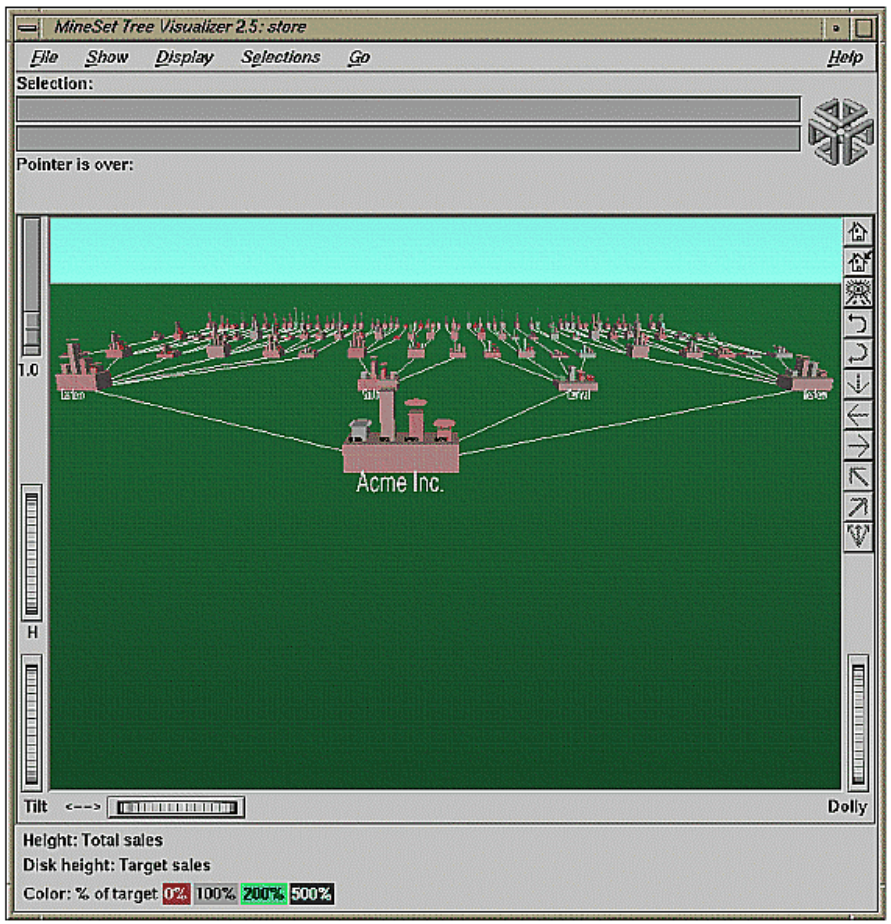


MineSet (1)

- Société : SGI (*Silicon Graphics Inc.*)
- Création : 1996
- Plate-forme : *Silicon Graphics*
- Utilisation
  - Services financiers
  - Prise de décisions
- Algorithmes de visualisation avancés

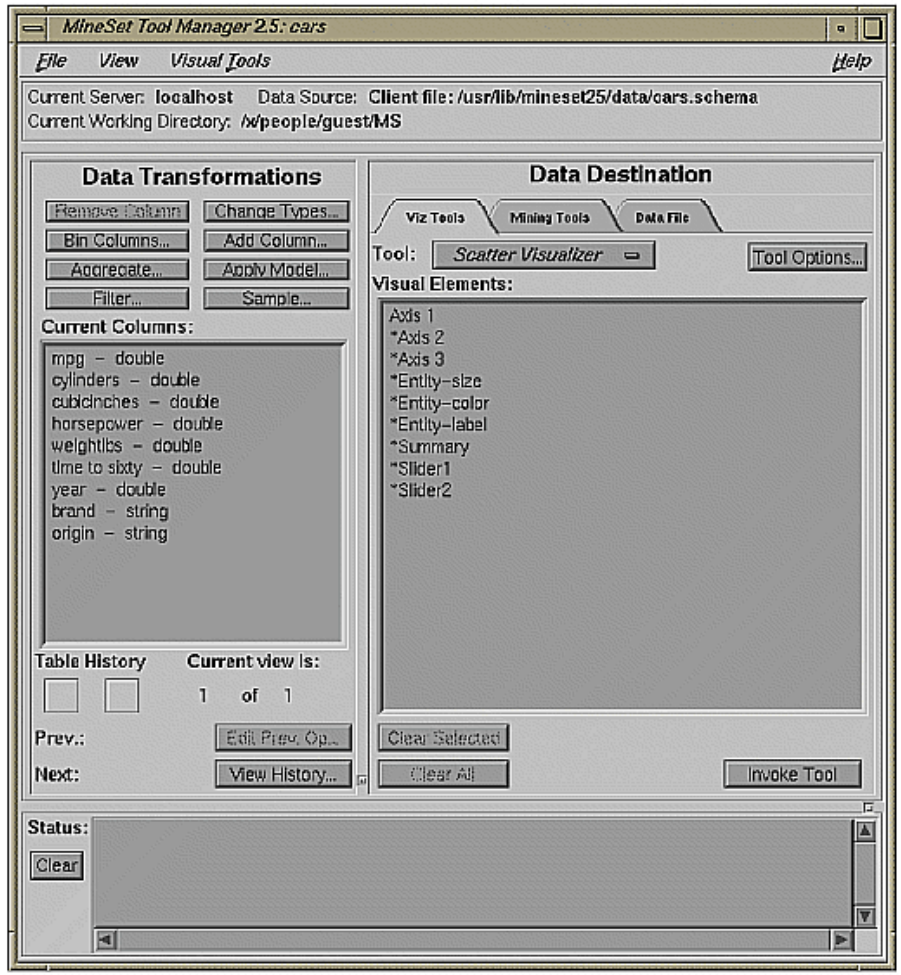
MineSet (2)

- Interface visuelle 3D



MineSet (3)

- Interface graphique
- client/serveur
  - Tool Manager (Client)
  - DataMover (Server)
- Utilisateurs
  - Managers
  - Analystes



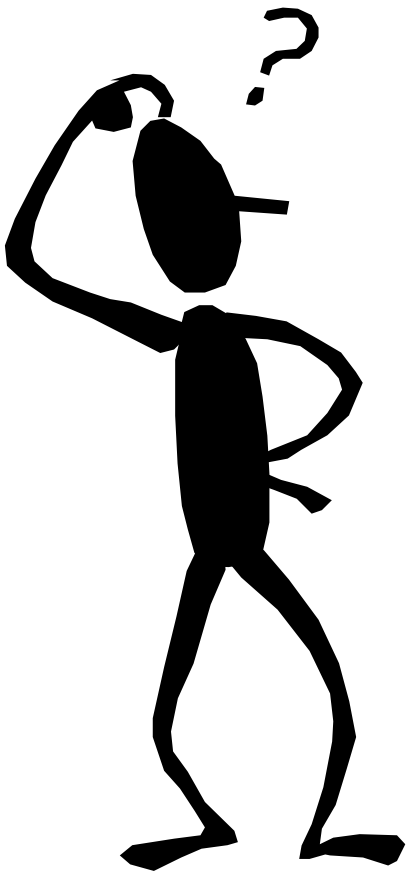
MineSet (4)

- Tâches
  - Règles d'association
  - Classification
- Présentation de la connaissance
  - Arbre
  - Statistiques
  - Clusters (nuages de points)

Synthèse

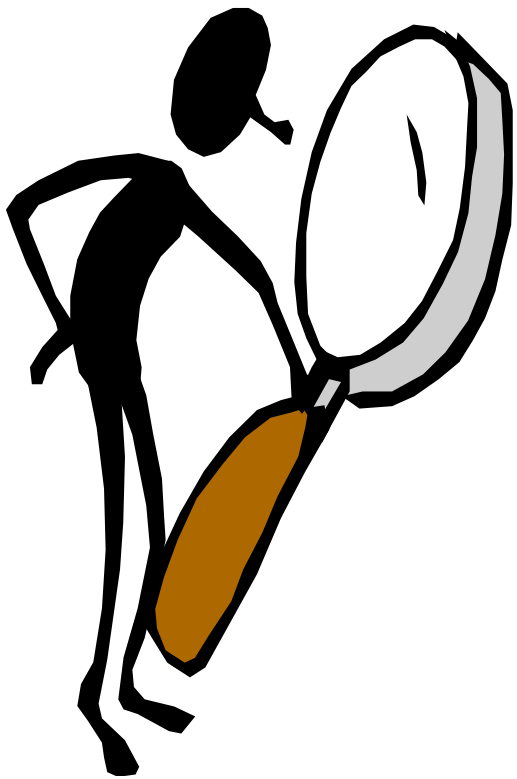
Company	Product	Link Analysis	Classification	Clustering	Statistics	Prediction	OS	Others
IBM	Intelligent Miner	3/4	3/7	2/3	6/6	4/5	1/3	3/5
ISoft	Alice / AC2	0/4	1/7	0/3	3/6	0/5	3/3	1/5
SAS Institute Inc.	SAS Enterprise Miner	0/4	3/7	2/3	6/6	3/5	2/3	0/5
Silicon Graphics Inc.	MineSet	0/4	2/7	1/3	0/6	1/5	2/3	4/5
SPSS Inc.	Clementine	3/4	3/7	1/3	3/6	2/5	2/3	2/5

Autres techniques de Data Mining



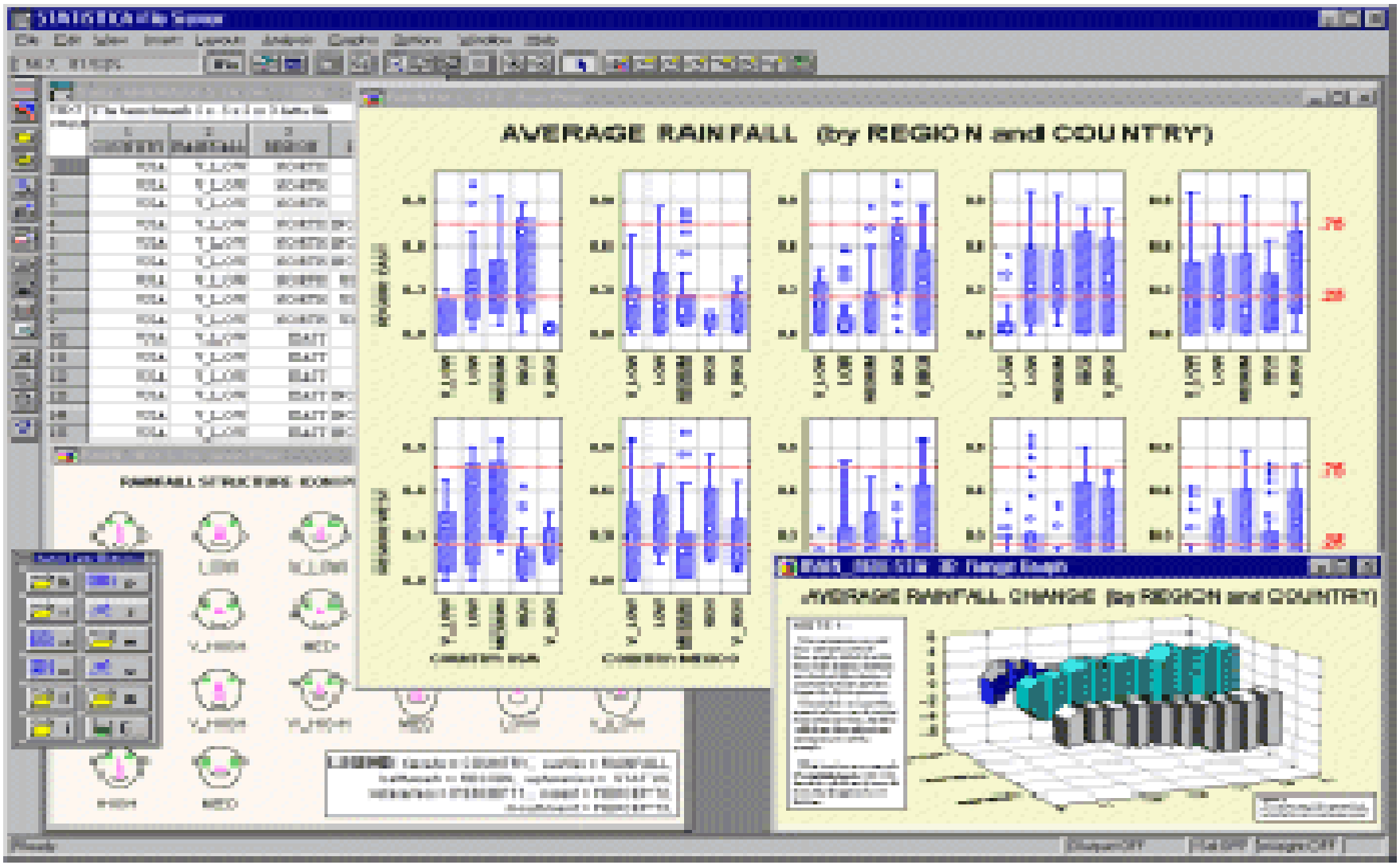
- Web mining (contenu, usage, …)
- Visual data mining (images)
- Audio data mining (son, musique)
- Data mining et requêtes d'interrogation "intelligentes"

Visualisation de données

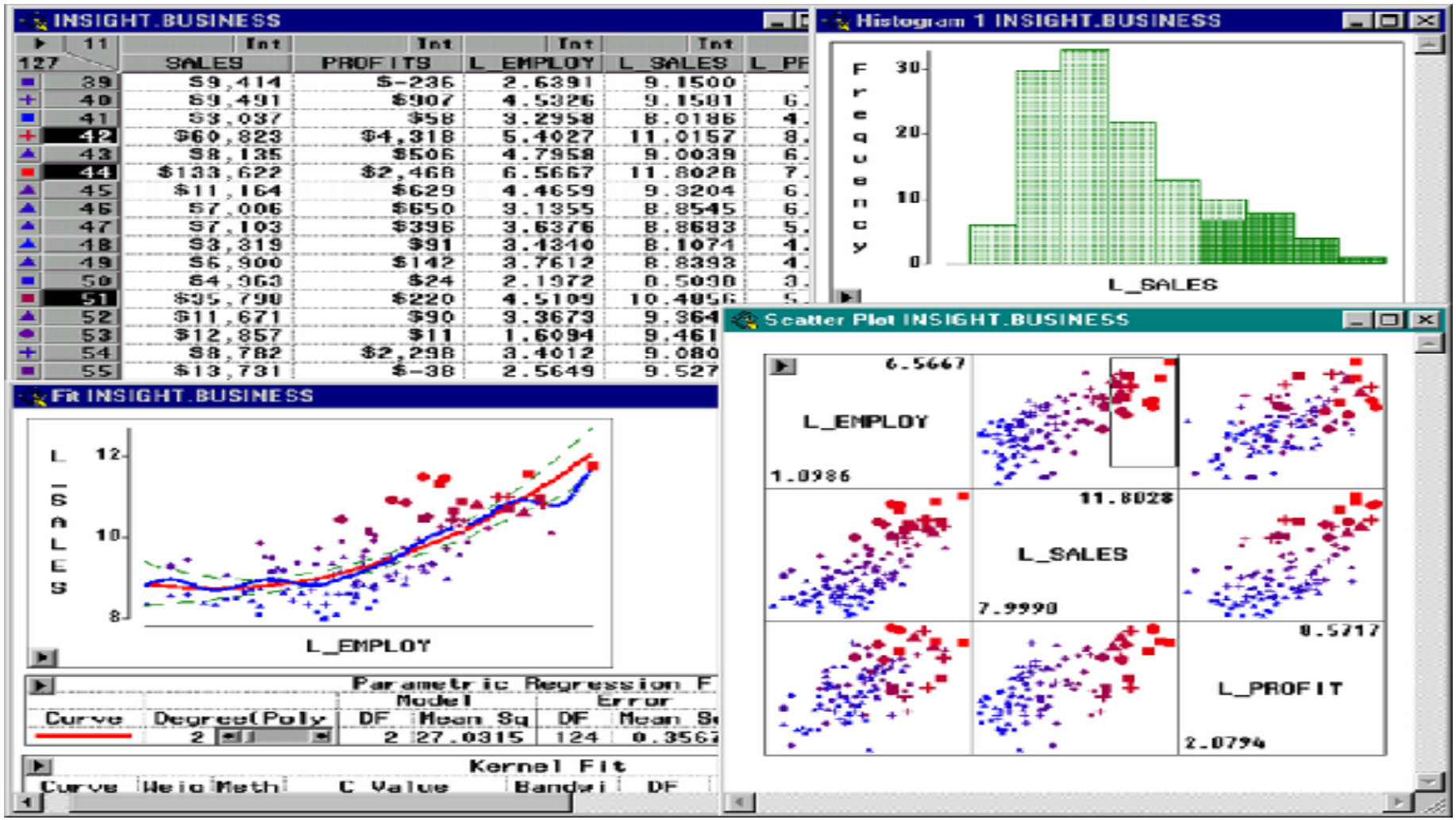


- **Données dans un base de données ou un entropot de données peuvent être visualisées :**
  - À différents niveaux de granularité ou d'abstraction
  - A l'aide de différentes combinaisons d'attributs ou dimensions
- **Résultats des outils de Data Mining peuvent être présentées sous diverses formes visuelles**

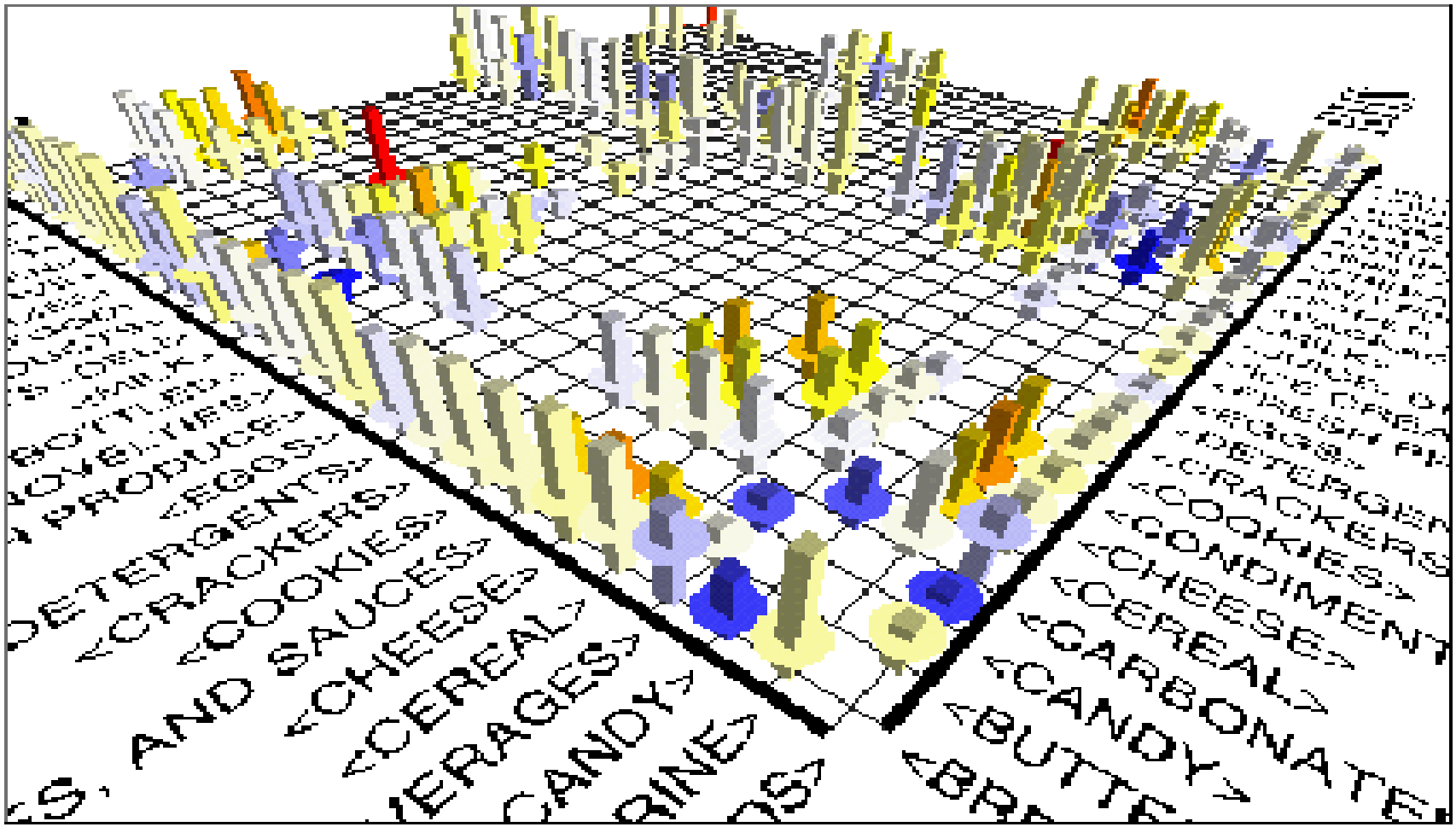
Box-plots dans StatSoft



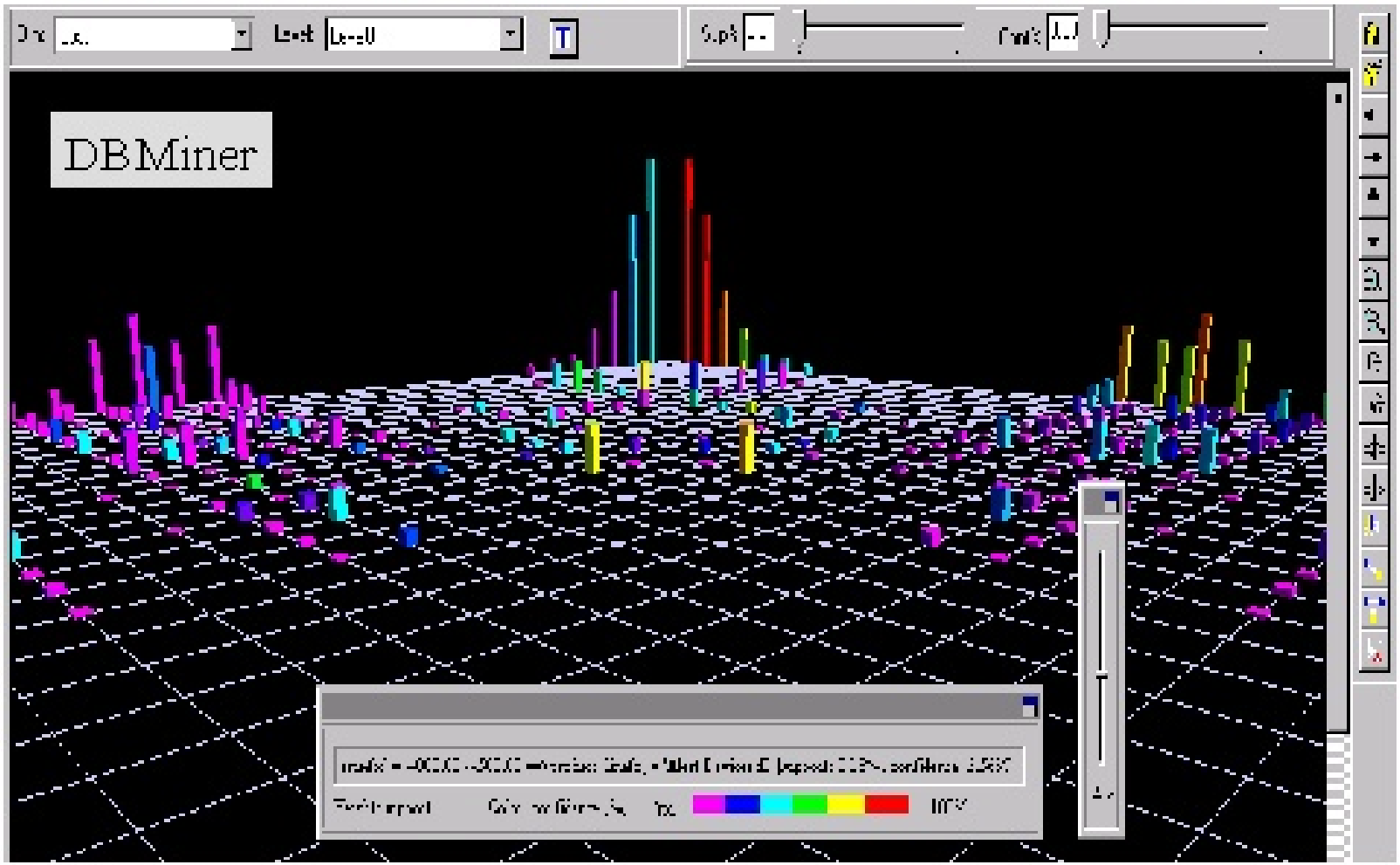
Scatter-plots dans SAS Enterprise Miner



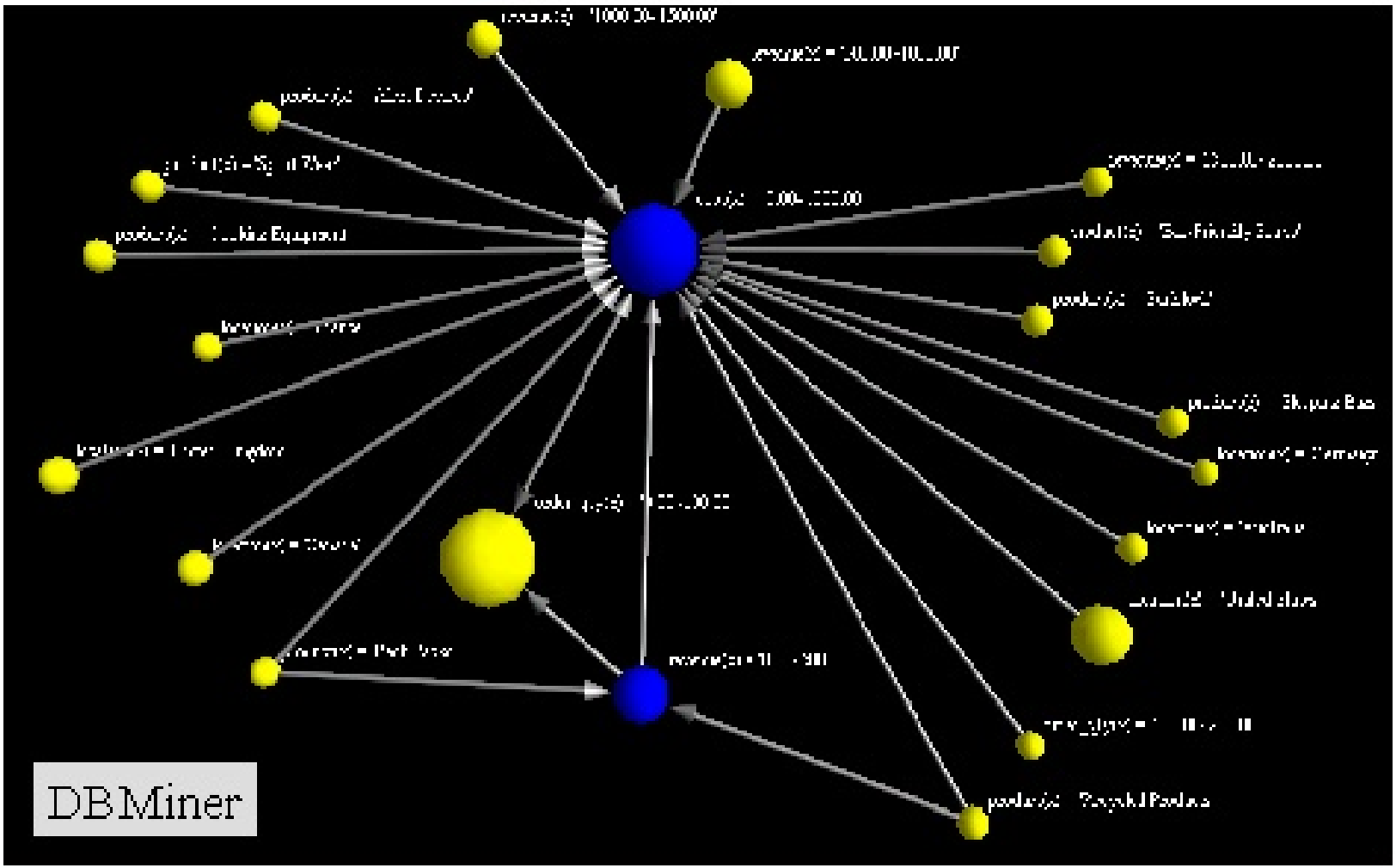
Règles d'association dans MineSet 3.0



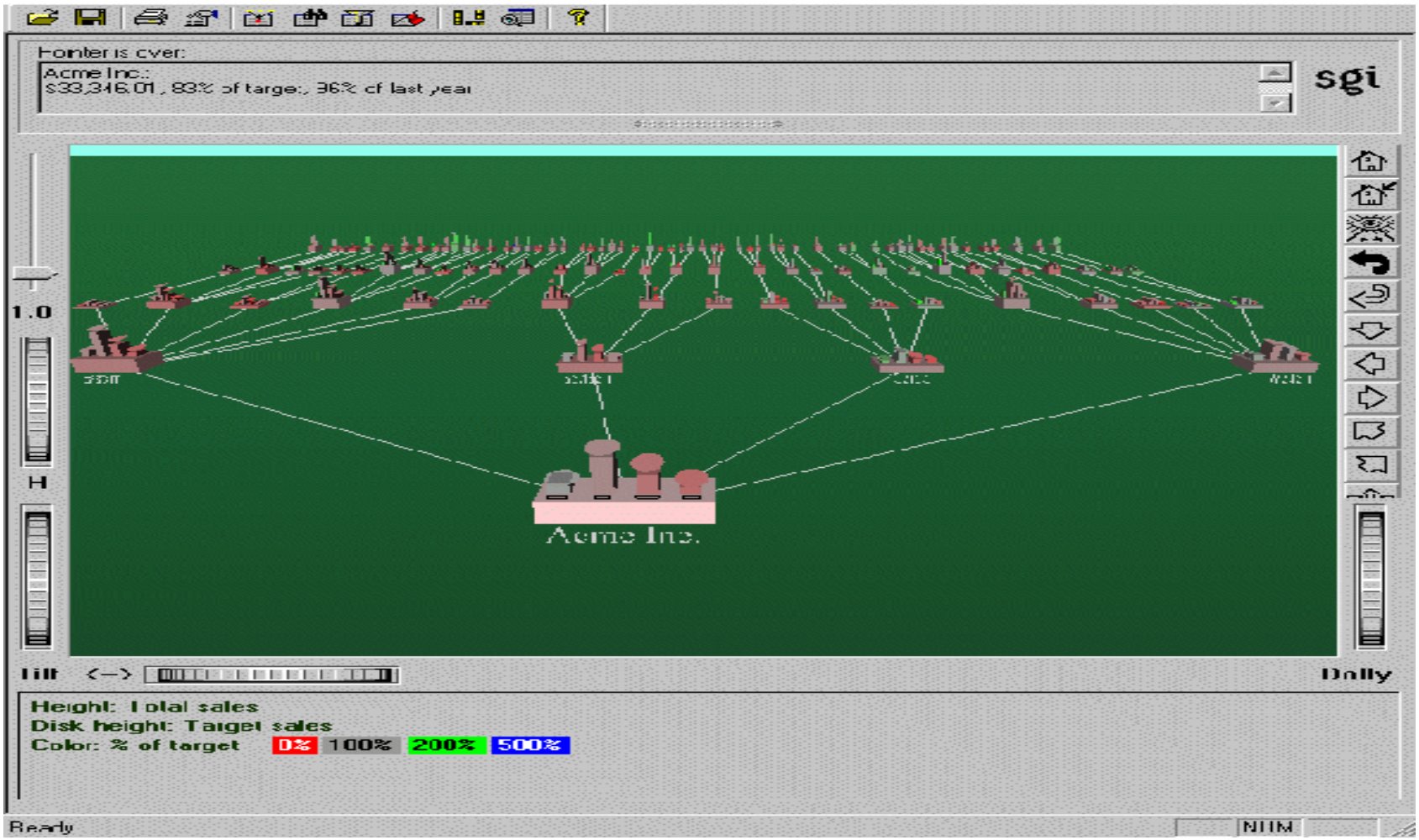
Visualization of Association Rule in Plane Form



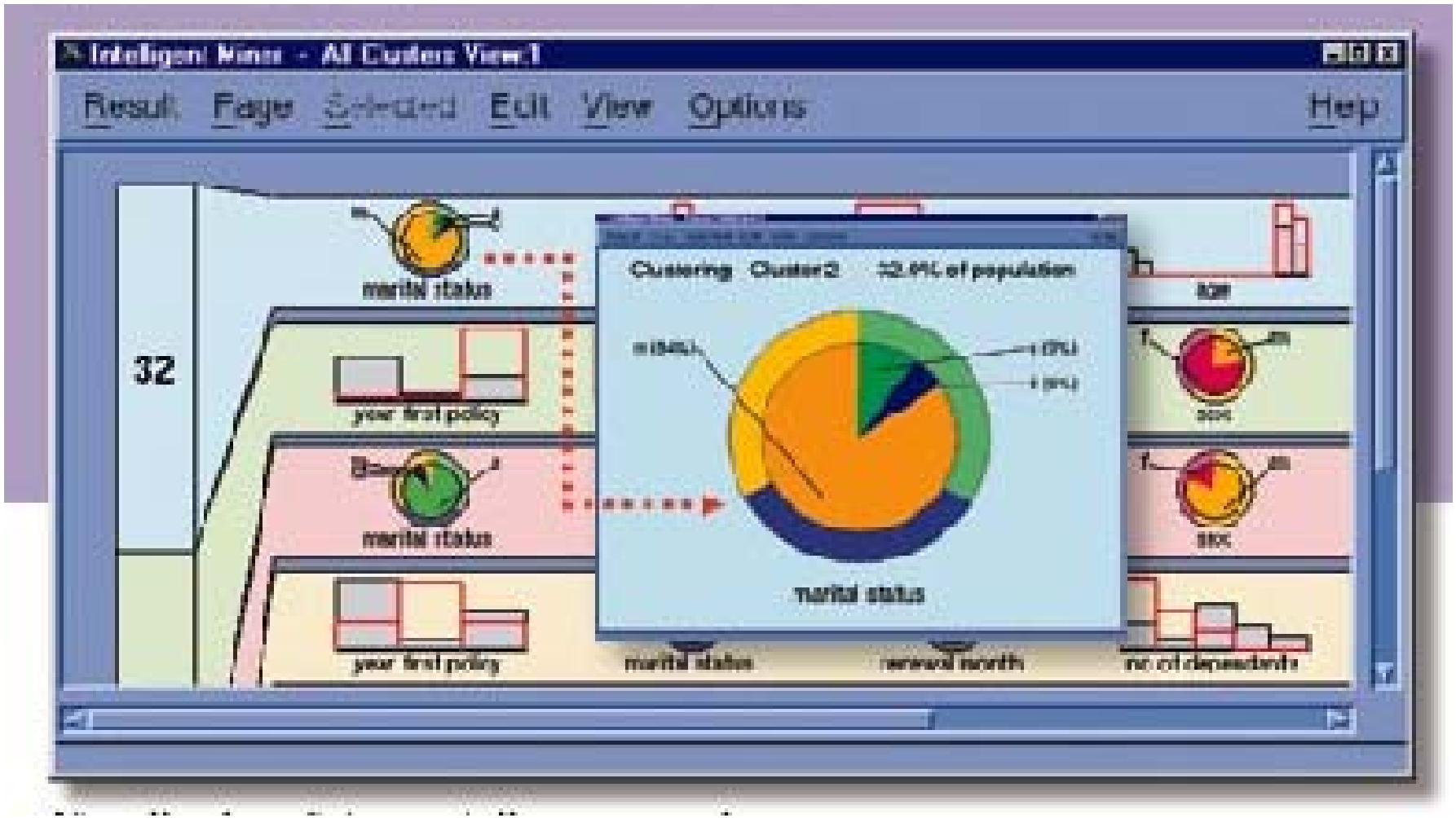
Visualization of Association Rule Using Rule Graph



Arbres de décision dans MineSet 3.0



Clusters dans IBM Intelligent Miner



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 …**

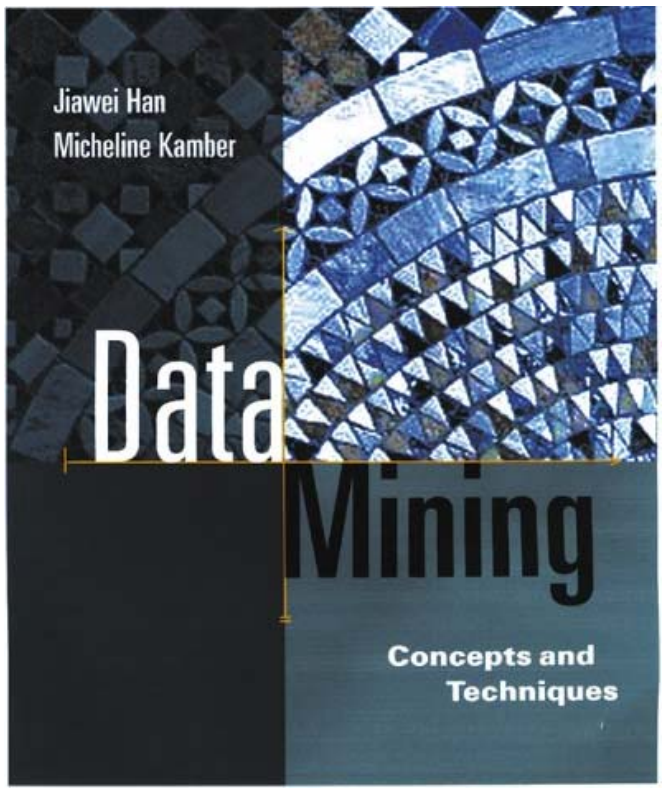
Références bibliographiques (1)

- Georges Gardarin
  - Université de Versailles (laboratoire PRISM)
  - Internet/intranet et bases de données – Data Web, Data Warehouse, Data Mining, Ed. Eyrolles
  - http://torquenada.prism.uvsq.fr/~gardarin/home.html
- Rakesh Agrawal (IBM)
  - IBM Almaden Research Center
  - http://www.almaden.ibm.com/cs/people/ragrawal/
- Mohammed Zaki
  - Rensselaer Polytechnic Institute, New York
  - http://www.cs.rpi.edu/~zaki/

Références bibliographiques (2)

- Vipin Kumar
  - Army High Performance Computing Research Center
  - http://www-users.cs.umn.edu/~kumar
- Rémi Gilleron
  - Découverte de connaissances à partir de données, polycopié (Université de Lille 3)
  - http://www.univ-lille3.fr/grappa
- *The Data Mine*
  - http://www.cs.bham.ac.uk/~anp/TheDataMine.html
- *Knowledge Discovery Nuggets* (*Kdnuggets*)
  - www.kdnuggets.com

Références bibliographiques (3)



•"Data Mining: Concepts and Techniques“
by Jiawei Han and Micheline Kamber,
Morgan Kaufmann Publishers,
August 2000. 550 pages. ISBN 1-55860-489-8

Conférences - Historique

- 1989 Workshop IJCAI
- 1991-1994 Workshops KDD
- 1995-1998 Conférences KDD
- 1998 ACM SIGKDD
- 1999- Conférences SIGKDD
- Et plusieurs nouvelles conférences DM …
  - PAKDD, PKDD
  - SIAM-Data Mining, (IEEE) ICDM
  - etc.

Conférences - Journaux

"Standards"

- **DM:**
  - Conférences : KDD, PKDD, PAKDD, ...
  - Journaux : Data Mining and Knowledge Discovery, CACM
- **DM/DB:**
  - Conférences : ACM-SIGMOD/PODS, VLDB, ...
  - Journaux : ACM-TODS, J. ACM, IEEE-TKDE, JIIS, ...
- **AI/ML:**
  - Conférences : Machine Learning, AAAI, IJCAI, ...
  - Journaux : Machine Learning, Artific. Intell., ...