Machine Learning

Data Mining – Fouille de données

Ali Idri, Ph.D.
ENSIAS, Université Mohamned V de Rabat
MSDA, Université Mohammed VI Polytechnique, Ben Guerir
E-mails: ali.idri@um5.ac.ma, Ali.Idri@um6p.ma

Objectifs

- Connaître les tâches et les objectifs du DataMining
- Utiliser les techniques du Machine Learning
- Ateliers ML avec Python

Pré-requis

- Algorithmique et Structures de données
- Analyse de données
- Intelligence Artificielle

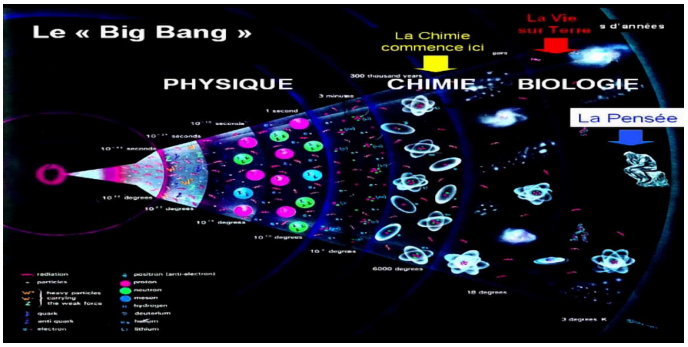
Course Evaluation

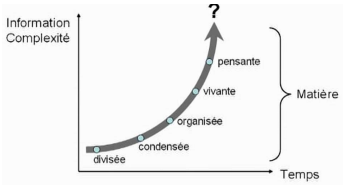
- Examen écrit (40%)
- Mini-projets (40%)
- Assiduité et présence (20%)

Plan

- Historique
- Introduction
- Data Preprocessing
- Classification
- Prédiction
- Clustering
- Association
- Mini-projets avec Python

Artificial Intelligence?





Historique –IA-

- **1900, David Hilbert, Décidabilité?**
- **1931, Kurt Gödel démontra que, dans tout système axiomatique manipulant l'arithmétique des nombres naturels, il existe des propositions pour lesquelles le système est incapable d'assigner la valeur *vraie ou fausse***
- ***1936, Alan Turing, Calculabilité vs* Décidabilité*?***
- **1950, Turing esquisse une théorie de l'esprit fondée sur les concepts de *Test de Turing et* de la *Machine universelle de Turing***
- **1956, la conférence du Dartmouth College Artificial Intelligence**

Historique -IA-

- ***Intelligence artificielle faible (une discipline d'ingénierie), étudiée par certains* groupes du MIT (McCarthy, Minsky et al.), a comme objectif le développement des systèmes intelligents simulant certains des comportements humains admis et reconnus par tous comme étant intelligents**
- ***Intelligence artificielle forte étudiée à Carnegie -Mellon (Newell et Simon) a* comme objectif de modéliser l'esprit humain en construisant un programme qui, conjointement avec l'ordinateur, reproduira toute la cognition humaine:**

Esprit = Symboles + Computation

Data Mining vs Machine Learning

- Data Mining englobe Machine Learning:
  - Intelligence = Learning
  - Machine = Computer (Things)
- Herbert Simon:

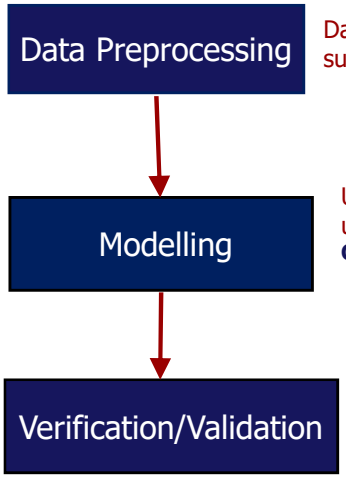
  "Learning is any process by which a system improves performance from experience."
- Learning est une caractéristique de l'intelligence

Machine Learning?

Traditional Programming

Machine Learning

Knowledge Data Discovery



Data imperfections are handled in order to successfully use DM techniques

Use of intelligent techniques in order to obtain useful patterns **(Classification, Regression, Clustering and Association)**

Are we building the product right?
Are we building the right product?

Data Mining? (1/3)

- Le **Data Mining** est l'ensemble des :
  - algorithmes et méthodes
  - destinés à l'exploration et l'analyse
  - …de (souvent grandes) bases de données informatiques
  - … **en vue de détecter dans ces données des règles, des associations, des tendances inconnues (non fixées a priori), des structures particulières restituant de façon concise l'essentiel de l'information utile**
  - … **pour aider à la prise de décision**

13

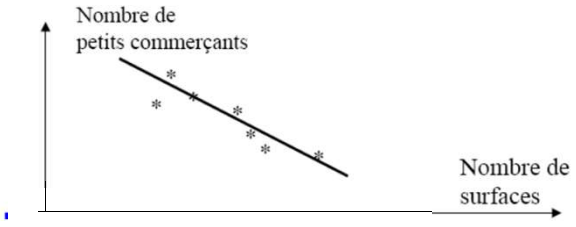
Data Mining? (2/3)

- Ensemble de méthodes et de techniques d'analyse de données et d'extraction d'information structurée en vue d'aider à la prise de décision:
  - Mettre en évidence des informations présentes mais noyées par le volume de données: **Datamining descriptif (ML Non-Supervisé)**
  - Extrapoler des nouvelles informations à partir de données existantes: **Datamining prédictif (ML Supervisé)**

14

Data Mining? (3/3)

- **Datamining descriptif**
  - Recherche d'association entre les attributs
  - Articles figurant dans le même ticket de caisse
  - Ex: achat de riz + limonade==> achat de poissons
- **Datamining prédictif**



Notion d'induction (1/2)

- **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 utilisant des règles d'inférence: **à partir de cas abstraits, on déduit des cas particuliers**
  - Toutes les voitures ont 4 roues
  - La Peugeot 206 est une voiture
    ==> La Peugeot 206 a 4 roues

Notion d'induction (2/2)

- **Induction**: Généralisation d'une observation ou d'un raisonnement établi à partir de cas singuliers.
  - Utilisée en Datamining (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

- Masse importante de données
- Données multi-dimensionnelles (plusieurs attributs)
- Inexploitables par les méthodes d'analyse de données classiques
- Forte pression due à la concurrence du marché
- Besoin de prendre des décisions stratégiques efficaces
  - Exploiter le vécu (données historiques) pour prédire le futur et anticiper le marché

Historique (1/2)

- 1875 : régression linéaire de Francis Galton
- 1896 : formule du coefficient de corrélation de Karl Pearson
- 1900 : distribution du χ^2 de Karl Pearson
- 1936 : analyse discriminante de Fisher et Mahalanobis
- 1941 : analyse factorielle des correspondances de Guttman
- 1943 : réseaux de neurones de Mc Culloch et Pitts
- 1944 : régression logistique de Joseph Berkson
- 1958 : perceptron de Rosenblatt
- 1962 : analyse des correspondances de J.-P. Benzécri
- 1964 : arbre de décision AID de J.P.Sonquist et J.-A.Morgan
- 1965 : méthode des centres mobiles de E. W. Forgy
- 1967 : méthode des k-means de Mac Queen
- 1972 : modèle linéaire généralisé de Nelder et Wedderburn

Historique (2/2)

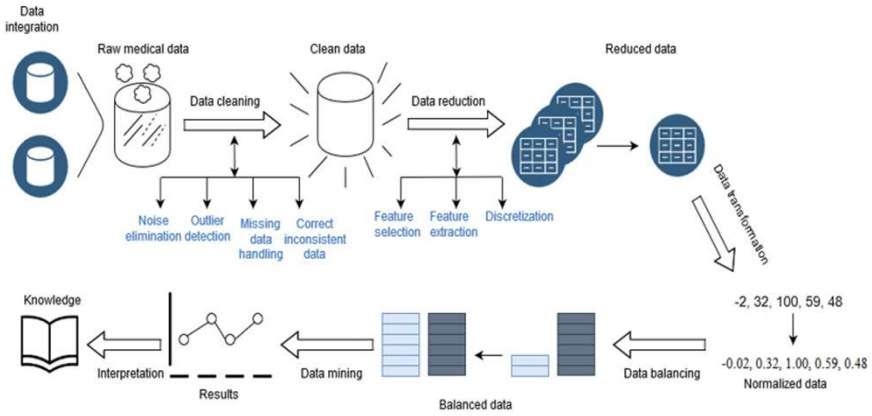
- 1975 : algorithmes génétiques de Holland
- 1980 : arbre de décision CHAID de KASS
- 1984 : arbre CART de Breiman, Friedman, Olshen, Stone
- 1986 : perceptron multicouches de Rumelhart et McClelland
- 1989 : réseaux de T. Kohonen (cartes auto-adaptatives)
- **vers 1990 : apparition du concept de data mining / Machine Learning**
- 1993 : arbre C4.5 de J. Ross Quinlan
- 1996 : bagging (Breiman) et boosting (Freund-Shapire)
- 1998 : support vector machines de Vladimir Vapnik
- 2000 : régression logistique PLS de Michel Tenenhaus
- 2001 : forêts aléatoires de L. Breiman

DataMining et Statistiques

- Le datamining englobe la statistique et l'analyse des données traditionnelle, il en diffère par :
  - Certaines techniques de DM n'appartiennent qu'à lui (réseaux de neurones, arbres de décision, etc.)
  - Le nombre d'individus étudiés est souvent plus important en DM, où l'optimisation des algorithmes est importante
  - Le DM fait moins d'hypothèses contraignantes sur les lois statistiques suivies
  - DM recherche parfois plus la compréhensibilité des modèles que leur précision

Processus KDD(1/3)

- DataMining est au coeur du Knowledge Data Discovery



Data ==> Information ==> Knowledge

Processus KDD (2/3)

- **Data preprocessing?**
- **Why:**
  - *« Garbage in, Garbage out »*
  - *40% du temps KDD est didié au Data preprocessing*
- **Data preprocessing tasks**
  - Data Cleaning
  - Data Reduction
  - Data Transformation
  - Data Balancing

Processus KDD (3/3)

- DataMining Modelling
  - Sélectionner la bonne technique de modélisation
  - Construire le modèle
  - Évaluer le modèle sur le reste de la base de données
  - Réitérer le processus si nécessaire
- Vérifier et Valider le modèle
  - Analyser et interpréter la connaissance (intérêt)
  - Gérer la connaissance découverte et la mettre à la disposition des décideurs (visualisation)
  - Exploiter la connaissance

Data Cleaning

- Real-world data tend to be incomplete, noisy, and inconsistent
- Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies
- ***Missing data*** occur when no *data value* is stored for a variable in an observation. *Missing data* are a common occurrence and can have a significant effect on the conclusions that can be drawn from the *data*
- **Noise** is incorrect, void, null information that is not useful at all, under any circumstances
- An **outlier** is not a false value or void in meaning. It is definite and accurate but when it is linked with the other tuples, it is just not in the same range
- **Data inconsistency** refers to a situation of keeping the same data in different formats. It often arise between various duplicates, due to inaccurate data entry or updating some but not all data occurrences or in data representations (e.g., "2010/12/25" and "25/12/2010" for date).

Data Cleaning –Missing data

⊙ Incomplete data can occur due to:

- Data privacy
- They were not considered important at the time of entry
- Misunderstanding or of equipment malfunctions
- Etc.

⊙ Different methods exist to handle missing values:

- **Deletion** : instances with at least one missing value are discarded from the data set. Deletion techniques are the most appealing to practitioners due to their simplicity, however they have many disadvantages following the elimination of valuable data such as the loss of precision and bias of results
- **Toleration** : toleration techniques perform analysis directly on incomplete data sets
- **Imputation** : aims to fill in the missing values with estimated ones

Data Cleaning –Missing data

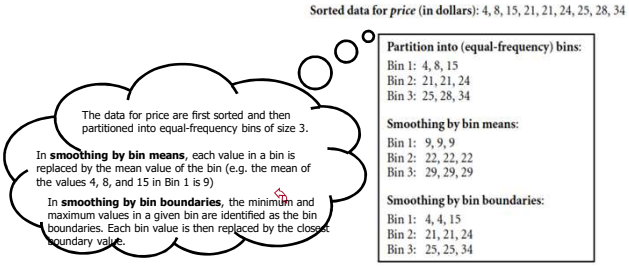
⊙ Imputation techniques

- **Machine learning imputation techniques** (ML) such as K-Nearest Neighbors Imputation, Support Vector Regression imputation and Decision Trees imputation
- **Statistical imputation techniques** such as Mean Imputation, Mode Imputation and Expectation Maximization Imputation

	Variables			Missing values replaced by means		
Respondent	A	B	C	A	B	C
1	2	6		2	6	8
2		6	2	8	6	2
3		6		8	6	8
4	10	10	10	10	10	10
5	10	10	10	10	10	10
6	10	10	10	10	10	10
Average	*8*	*8*	*8*	*8*	*8*	*8*

Data Cleaning –Noisy data

- Noisy data can be handled in following ways:
  - **Binning:** Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the values around it. The sorted values are distributed into a number of "buckets," or bins. Because binning methods consult the neighborhood of values, they perform local smoothing



Data Cleaning –Noisy data

- **Regression:** is a technique that conforms data values to a function. For instance, Linear regression involves finding the "best" line to fit two attributes (or variables) so that one attribute can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface

Linear Regression

- **Simple:**

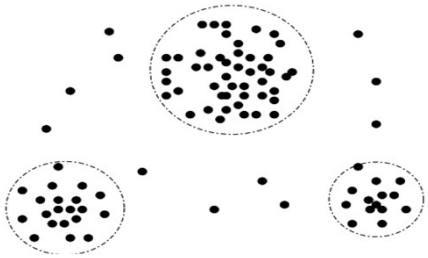
$$y = b_0 + b_1 * x$$

- **Multiple:**

$$y = b_0 + b_1 * x_1 + \ldots + b_n * x_n$$

Data Cleaning –Outliers

- **Outliers** are often discarded as noise. However, one person's garbage could be another's treasure. For example, exceptions in credit card transactions can help us detect the fraudulent use of credit cards methods that can be used to detect outliers
- **Outliers** may be detected by clustering, for example, where similar values are organized into groups, or "clusters." Intuitively, values that fall outside of the set of clusters may be considered outliers:

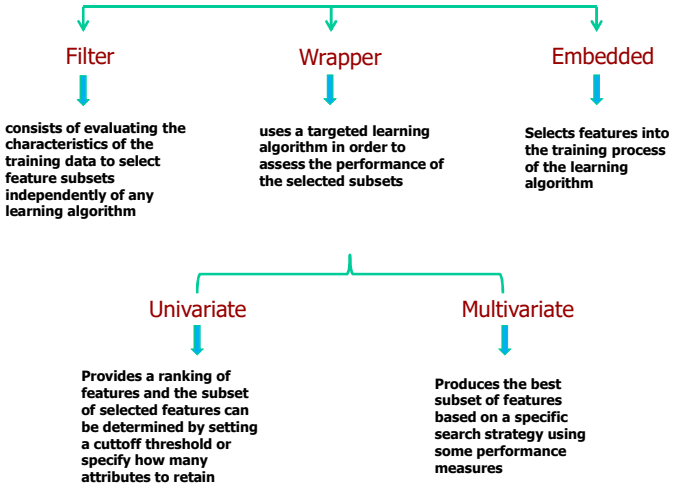


Data Reduction

- Data reduction obtains a reduced representation of the data set that is much smaller in volume, yet produces the same (or almost the same) analytical results:
  - Feature selection
  - Feature extraction
- **Feature Selection** techniques seek to reduce the number of original features by selecting those that are relevant and eliminating irrelevant and redundant ones
- **Feature Extraction** techniques, new attributes are constructed from the given set of attributes

Data Reduction –Feature Selection

- Feature selection techniques are generally classified as:



Data Reduction –Feature Extraction

- Finding new features that are calculated as a function of the original features. Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions by using linear or a nonlinear combination of the original features

Original features
X1, X2, X3, …….Xn

To

Extracted features
C1, C2,….Cm

m<n

Data Reduction -Examples

Data reduction task	Method Type		Examples	Pros	Cons
Feature Selection	Filter	Univariate	Chi-square, Info Gain, ReliefF..	- Simple and fast - Scalable - Independent from the classification algorithm	- Do not consider feature correlations - Do not dictate the optimum number of features to be selected.
		Multivariate	Correlation-based feature selection (CFS)	- Ability to take into account feature dependencies - Better results than univariate methods - Independent from the classification algorithm	Higher computational complexity
	Wrapper	Univariate	SVM weight vector	Excellent choice when dealing with high nonlinearity of data	Critically depends on having clean data
		Multivariate	Genetic Algorithm, Particle Swarm Optimization	Ability to take into account feature dependencies	- Higher risk of overfitting - Computationally intensive
	Embedded		Decision Tree, Random Forest..	Require less computation than wrapper methods	Specific to a machine learning algorithm
Feature Extraction	-		Principal Component Analysis (PCA)	-Removes correlated features - Simple and low cost	- Information loss - Independent variables become less interpretable - Unsupervised method - Requires data normalization

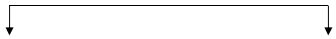
Data Transformation -Normalization

- This step is taken in order to transform the data in appropriate forms suitable for mining process
- The most frequently used data transformation task is **Normalization**
- In the normalization process, the attribute data are scaled so as to fall within a smaller range, such as [-1,1] or [0, 1], to give all attributes an equal weight
- **Normalization** is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering
- There are many methods for data normalization, however, **Min-max** and **Z-score** are two of the most commonly used techniques:

Techniques	Pros	Cons
Min-max: Performs a linear transformation on the original data	A simple and good technique to use when you know the distribution is not Gaussian	- Can be biased in the presence of outliers - Infeasible if the min or max values are not known
Z-score: Consists of centering the variable at zero and standardizing the variance at 1	Effective if the attribute distribution is Gaussian	- Sensitive to the presence of outliers

Data Transformation -Discretization

- **Discretization** transforms numerical attributes into discrete or nominal attributes with a finite number of intervals, obtaining a non-overlapping partition of a continuous domain.
- Discretization techniques are generally classified as:



Supervised

- Uses class information during the discretization process

Unsupervised

- Do not consider the class label (e.g. only considers the attribute being discretized

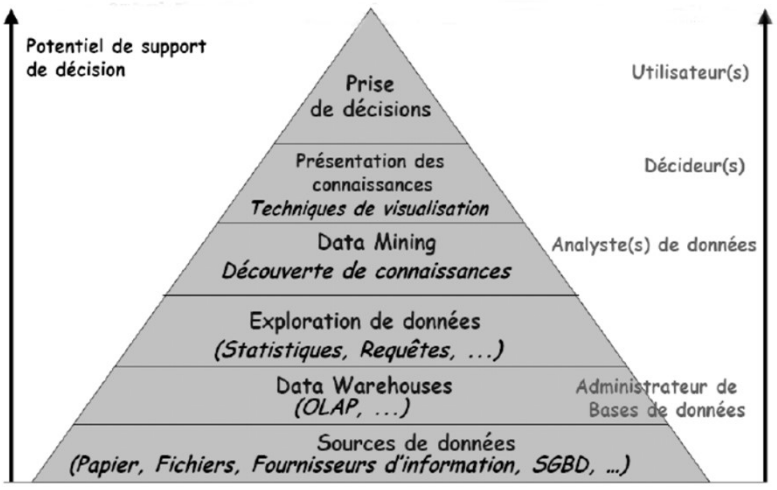
Data Balancing

- **Imbalanced data** refers to a dataset within which one or some of the classes have a much greater number of training examples than the others
- The most prevalent class is called the **majority class**, while the rarest class is called the **minority class**
- There are many methods for data balancing such as **sampling**, **cost-sensitive**, kernel-Based and active learning methods
- **Sampling** consists of the modification of an imbalanced data set by some mechanisms in order to provide a balanced distribution
- In **cost-sensitive learning** instead of each instance being either correctly or incorrectly classified, each class is given a misclassification cost. Thus, instead of trying to optimize the accuracy, the problem is then to minimize the total misclassification cost.

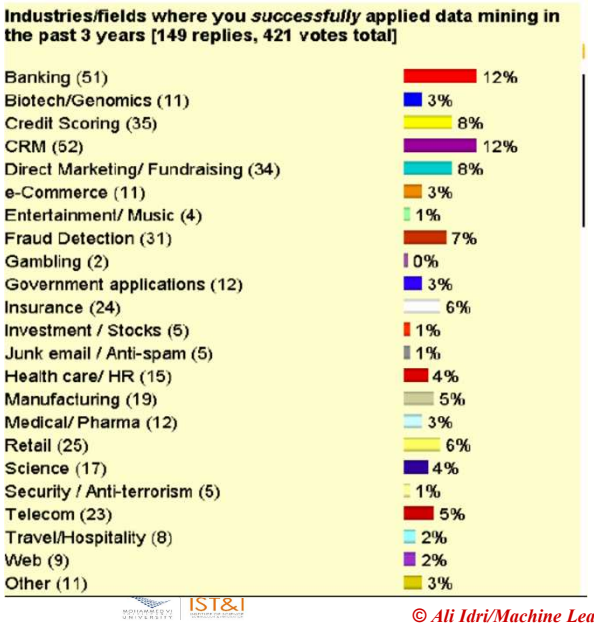
Data Balancing -Examples

Techniques		Pros	Cons
Sampling methods	Random Oversampling: is defined as adding more copies of the minority class	Unlike undersampling, this method leads to no information loss	- It increases the likelihood of overfitting - Increases the number of training examples, thus increasing the learning time
	Random Undersampling: randomly selects a set of majority class examples and removes them	Can help improve the runtime of the model and solve the memory problems when the training data set is enormous	Could lead to loss of potentially important information
	Synthetic Minority Oversampling Technique (SMOTE): uses a nearest neighbors algorithm to generate new and synthetic data	- Simple to implement and interpret - Alleviates overfitting caused by random oversampling as synthetic examples are generated rather than replication of instances	- Not very practical for high dimensional data - Does not take into consideration neighboring which can increase the overlapping of classes
Cost-sensitive	Cost-Sensitive Decision Trees C5.0	Yields good results for large data sets	Assume that cost information is provided

DataMining et prise de décision



Sondage www.kdnuggets.com
juillet 2005



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, effet des périodes de solde ou de publicité, panier de la ménagère
- Détection de fraudes: Télécommunications, Banques,…
- Gestion de stocks: quand commander un produit, quelle quantité demandée, …

Domaines d'application

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

Domaines d'application
Marketing

- Vous êtes gestionnaire marketing d'un opérateur de télécommunications mobiles
- Les clients reçoivent un téléphone gratuit (valeur150€) avec un contrat d'un an
- **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.

Domaines d'application
Marketing

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

Domaines d'application
Assurance

- 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 ?

Domaines d'application
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.

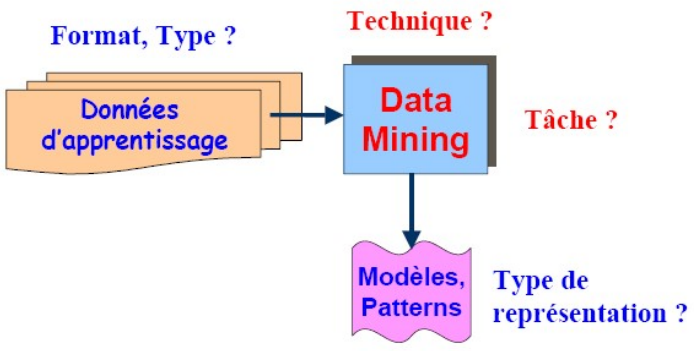
Domaines d'application
Banque

- 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.)

Communautés impliquées

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

Paramètres du processus KDD



Données (1/2)

- Valeurs des champs des enregistrements des instances de l'entrepôt de données
- **Données continues** : dont les valeurs forment un sous-ensemble de IR (exemple : salaire, age,etc.)
- **Données discrètes** : dont les valeurs forment un sous-ensemble de IN (exemple : nombre d'enfants, nombre d'étudiants, etc.)
- **Données énumératives** (ou qualitatives) dont l'ensemble des valeurs est fini. Ces valeurs sont alphanumériques (couleur, gender) ou numériques: ce ne sont que des codes et non des quantités (ex : n° de département)
- Dates
- Données textuelles : **Text Mining**
- Pages/liens web, Multimédia, … **Web Mining**

Données (2/2)

- Les données continues et discrètes sont des quantités :
  - on peut effectuer sur elles des opérations arithmétiques
  - elles sont ordonnées (on peut les comparer par la relation d'ordre <)
- Les données énumératives ne sont pas des quantités
  - mais sont parfois ordonnées : on parle de données énumératives **ordinales** (exemple : faible, moyen, fort)
  - les données énumératives **nominales** ne sont pas ordonnées. On ne peut que les distinguer.

Données et Techniques DM

- La régression linéaire traite les variables continues
- L'analyse discriminante traite les variables explicatives continues et les variables « cibles » nominales
- La régression logistique traite les variables explicatives continues, binaires ou nominales, et les variables « cibles » nominales ou ordinales
- Les réseaux de neurones traitent de préférence les variables continues dans [0,1]
- Certains arbres de décision (CHAID, ID3) traitent directement les variables discrètes et énumératives mais discrétisent les variables continues
- D'autres arbres de décision (CART, C4.5, C5.0) peuvent aussi traiter directement les variables continues
- Toutes les méthodes ne gèrent pas tous les types de données

Déscritisation

Avantages

- Traiter simultanément des données quantitatives et qualitatives
- Traiter un nombre limité de valeurs (< 7)
- Neutraliser les valeurs extrêmes qui sont dans la 1ère et la dernière tranches
- Gérer facilement les valeurs manquantes
- Renforcer la robustesse d'un modèle

53

Techniques des tâches descriptives

- Visent à **mettre en évidence des informations présentes** mais cachées par le volume des données (c'est le cas des segmentations de clientèle et des recherches d'associations de produits sur les tickets de caisse)
  - réduisent, résument, synthétisent les données
  - il n'y a pas de variable « cible » à prédire.
  - **Clustering** : groupes d'instances ayant des caractéristiques similaires : *classes non prédéfinies* (k-means, APCIII, hiérarchique, réseaux de Kohonen…)
  - **Recherche d'association** (règles d'association, Apriori,..)

54

Techniques des Tâches Prédictives

- Visent à **extrapoler de nouvelles informations** à partir des informations présentes (induction):
  - expliquent les données
  - il y a une (des) variable(s) « cible(s) » à prédire.
  - **Classification/Classement** : prédire la classe d'une instance donnée : *classes prédéfinies* (K-NN, Régression logistique, Arbres de décision, Réseaux de neurones, etc.)
  - **Prédiction** : prédire des variables continues (Régression linéaire simple et multiple, arbres de régression, réseaux de neurones, algorithmes génétiques, Etc.)

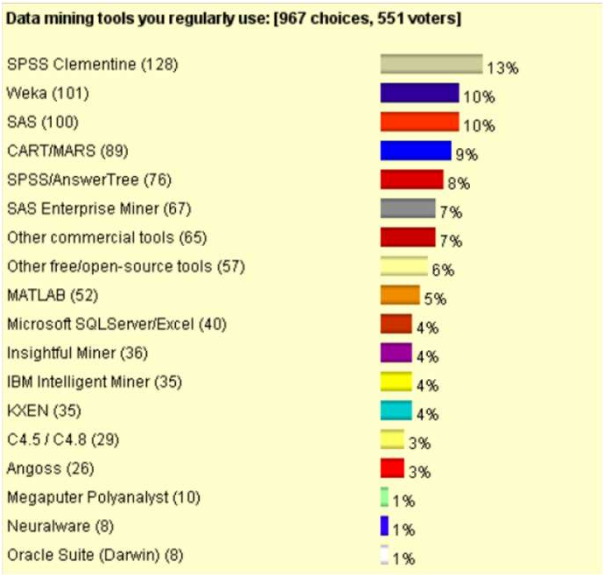
Logiciels DataMining

- Il existe de nombreux logiciels de statistique et datamining sur PC :
  - Faciles à installer et pas très chers
  - Avec des algorithmes de bonne qualité
  - Généralement conviviaux
  - Bons pour des PME car pouvant gérer plusieurs dizaines de milliers d'individus
  - **S-PLUS**™ de Insight, **Alice**™ de Isoft, **Predict**™ de Neuralware, **R** (version gratuite de S-PLUS) et les freewares **Weka** et **TANAGRA**…
- Cependant :
  - Ils ne permettent pas de traiter exhaustivement de très grandes bases de données
  - Ils ne mettent souvent en oeuvre qu'une ou deux techniques (sauf quelques produits tels S-PLUS, R, Tanagra et Weka)

- Certains logiciels sont conçus :
  - pour exploiter de grands volumes de données
  - pour couvrir une large palette de techniques
- Ils existent parfois en version « statistique » ou « datamining » (le 2nd étant parfois une sur-couche du 1er)
- Ils peuvent fonctionner en mode client-serveur
  - **SPSS**™ et **Clementine**™ de SPSS
  - **SAS/STAT**™ et **Enterprise Miner**™ de SAS
  - **Statistica Data Miner**™ de StatSoft
  - **S-PLUS**™ et **Insightful Miner**™ de Insightful
  - **IBM-Intelligent Miner** de IBM

Sondage sur www.kdnuggets.com

Juin 2002



Critères de choix

- Variété des algorithmes de datamining, de statistique et de préparation des données
- Qualité des algorithmes implémentés
  - documentation éditeur pas toujours accessible
- Capacité à traiter de grands volumes de données
  - peut être cruciale à partir de plusieurs centaines de milliers d'individus à traiter
- Existence d'un langage de programmation évolué
- Convivialité du logiciel et facilité à produire des rapports
- Prix !

Fonctionnalités d'un logiciel DM (1/5)

- Algorithmes de statistique et de datamining :
  - **Classification** (analyse discriminante linéaire, régression logistique binaire ou polytomique, modèle linéaire généralisé,régression logistique PLS, arbres de décision, réseaux de neurones, k-plus proches voisins...)
  - **Prédiction** (régression linéaire, modèle linéaire général, régression robuste, régression non-linéaire, régression PLS, arbres de décision, réseaux de neurones, k plus proches voisins…)
  - **Clustering** (centres mobiles, nuées dynamiques, k-means, classification hiérarchique, méthode mixte, réseaux de Kohonen…)
  - **Détection des associations** (règles d'association)
  - Etc.

Fonctionnalités d'un logiciel DM (2/5)

- Fonctions de préparation des données
  - Manipulation de fichiers (fusion, agrégation, ..)
  - Visualisation des individus, coloriage selon critère
  - Détection et filtrage des extrêmes
  - Analyse et imputation des valeurs manquantes
  - Transformation de variables (normalisation automatique, discrétisation…)
  - Création de nouvelles variables (fonctions logiques, statistiques, mathématiques…)
  - Sélection des variables les plus explicatives

Fonctionnalités d'un logiciel DM (3/5)

- Fonctions statistiques
  - détermination des caractéristiques de tendance centrale, de dispersion, de forme…
  - tests statistiques de moyenne, de variance, de distribution, d'indépendance, de multicolinéarité, etc.
- Fonctions d'échantillonnage et de partition des données
  - pour créer des échantillons d'apprentissage, de test et de validation
  - bootstrap, jackknife, etc.
- Fonctions d'analyse exploratoire des données et d'analyse factorielle
  - ACP, ACP avec rotation, AFC, ACM
- Langage avancé de programmation
  - macros

Fonctionnalités d'un logiciel DM (4/5)

- Présentation des résultats
  - visualisation des résultats
  - manipulation des tableaux
  - bibliothèque de graphiques (2D, 3D, ..)
  - navigation dans les arbres de décision
  - affichage des courbes de performances (ROC, lift, gain…)
  - facilité d'incorporation de ces éléments dans un rapport
- Gestion des métadonnées
  - variables définies identiquement pour tous les fichiers du projet

Fonctionnalités d'un logiciel DM (5/5)

- Plates-formes supportées (Windows, Unix, Sun, IBM MVS…)
- Formats d'entrée/sortie des données gérés :
  - Tables Oracle, Sybase, DB2, SAS, fichiers Excel, à plat...
- Enchaînements programmés de plusieurs algorithmes
- Volume de données pouvant être raisonnablement traité
- Pour plus de puissance
  - architecture client-serveur : calculs sur le serveur et visualisation des résultats sur le client
  - algorithmes parallélisés
- Exécution en mode interactif ou différé
- Portabilité des modèles construits (C, XML, Java, SQL...)

DataMining challenges

- Manque de données dans certains domaines
- Présence de points extrêmes (outliers)
- Overfitting
- Nombre d'attributs très élevé
- Qualité de données
- Visualisation des résultats
- Interprétation des résultats
- Bases de données très larges (Big Data)

Classification
Classement

Généralités

- Elle permet de prédire si une instance est membre d'un groupe ou d'une classe prédéfinie.
- La **classification/classement** consiste à placer chaque instance dans une classe, parmi plusieurs classes prédéfinies, en fonction des caractéristiques de l'instance indiquées comme variables explicatives
- Classes
  - Groupes d'instances avec des profils particuliers
  - Apprentissage supervisé: classes connues à l'avance
  - Applications : marketing direct (profils des consommateurs), médecine (malades/non malades), etc.
  - Exemple : les acheteurs de voitures de sport sont de jeunes ayant un revenu important

Exemple

Name	Gender	Heigh	class
Kristina	F	1.6m	Short
Jim	M	2m	Tall
Maggie	F	1.9m	Meduim
Martha	F	1.88	Meduim
Stephanie	F	1.7m	Short
Bob	M	1.85m	Meduim
Khaty	F	1.6m	Short
Dave	M	1.7m	Short
Worth	M	2.2m	Tall
Steven	M	2.1m	Tall
Debbie	F	1.8m	Meduim
Todd	M	1.95m	Meduim
Kim	F	1.9m	Meduim
Amy	F	1.8m	Meduim
Wynette	F	1.75m	Meduim

John est M et 1,75m -> classe(john)=??

Applications

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

Techniques

- **Techniques brutes (Lazy techniques)**
  - ne comprennent qu'une seule étape (éventuellement réitérée), au cours de laquelle chaque individu est directement classé (ou objet d'une prédiction) par référence aux autres individus déjà classés
  - il n'y a pas élaboration d'un modèle
- **Techniques inductives :**
  - une phase d'apprentissage (**phase inductive**) pour élaborer un modèle, qui résume les relations entre les variables et qui peut ensuite être appliquée à de nouvelles données pour en déduire un classement ou une prédiction (**phase déductive**)

Techniques brutes
Basées sur la similarité

- Soient N individus décrits par M attributs.
- Les classes prédéfinies sont C1, C2, …,Cn
- Soit un nouveau individu : I
  - Affecter I à la classe dont similarité(I, Ci) > similarité(I, Cj) avec Ci≠ Cj
- Algorithme

```
C1,..Cn : classes prédéfinies
I : nouveau individu
Dist=-∞
For j=1 to n
   if (sim(I,Cj)>dist) then {c=j; dist=sim(I,Cj)}
Afficher (c)
```

Similarité et Distance

- Il n'y a pas de définition unique de la similarité entre objets
  - Différentes mesures de distances d(x,y)
- La similarité entre objets dépend de :
  - type des données considérées
  - type de similarité recherchée

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

Attributs numériques : taille, age, poids,…

Distance : Données énumeratives

- 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.

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

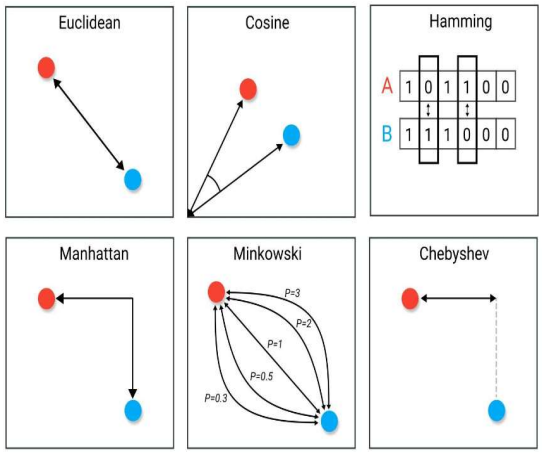
m : nombre de correspondances, **P**: nombre total de variables

- Distance de Hamming

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

$$\delta(x_i, y_i) = \begin{cases} 0 & si \quad x_i = y_i \\ 1 & si \quad x_i \neq y_i \end{cases}$$

Distances



KNN : K Nearest Neighbors

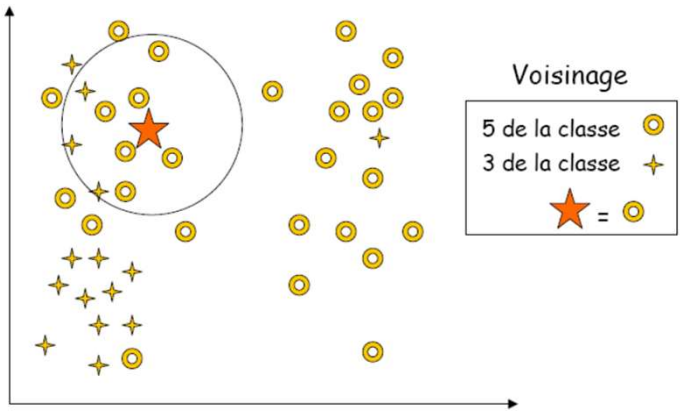
- Objectif : affecter une classe à une nouvelle instance
- Donnée: un échantillon de m enregistrements classés (x, c(x))
- Entrée: un enregistrement y
  1. Déterminer les k plus proches enregistrements de y
  2. Combiner les classes de ces k exemples en une classe c
- Sortie: la classe de y est c(y)=c

KNN – selection classe

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

77

KNN -Exemple



78

KNN - critiques

- Complexité : O(n) n:nombre d'individus
- KNN manipule l'ensemble des individus déjà classés, pour tout nouveau classement. Ce qui nécessite donc une grande puissance de stockage et de calcul
- Choix du k
- Choix de la mesure de similarité : distance
- Combinaison de classes
- **Il n'y a pas d'élaboration d'un modèle -> pas d'apprentissage**

Techniques basées sur les modèles

- 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 et utilisation 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, ...
- Vérification du modèle sur les instances d'apprentissage
- Validation du modèle sur des instances non utilisés dans l'apprentissage

Validation du modèle
Accuracy

- 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
- Taux d'erreur: matrice de confusion. Cas de deux classes P (Positive) et N (Négative)

Accuracy=TP+TN/TP+FP+TN+FN

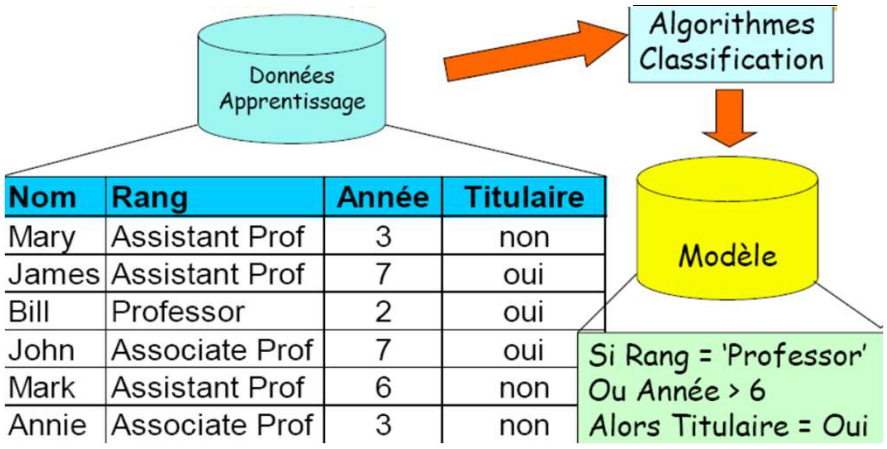
	P	N
P	TP	FP
N	FN	TN

Validation du modèle
Autres critères

- Problème de Imbalanced Data (une classe est majoritaire que les autres)
- **Sensitivity** (Recall) $Sensitivity = \frac{TP}{TP + FN}$
- **Specificity** $Specificity = \frac{TN}{TN + FP}$
- **Balanced Accuracy**

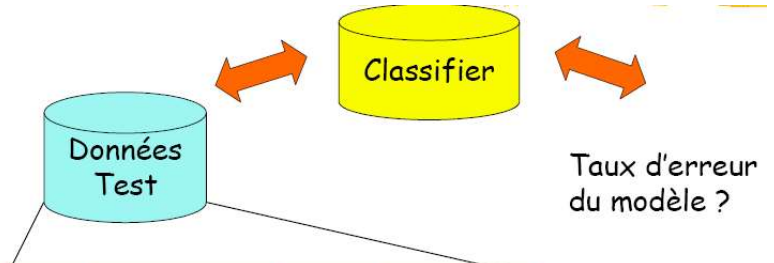
$$Balanced\ accuracy = \frac{\left(\frac{TN}{TN + FP} + \frac{TP}{TP + FN}\right)}{2}$$

Validation -exemple



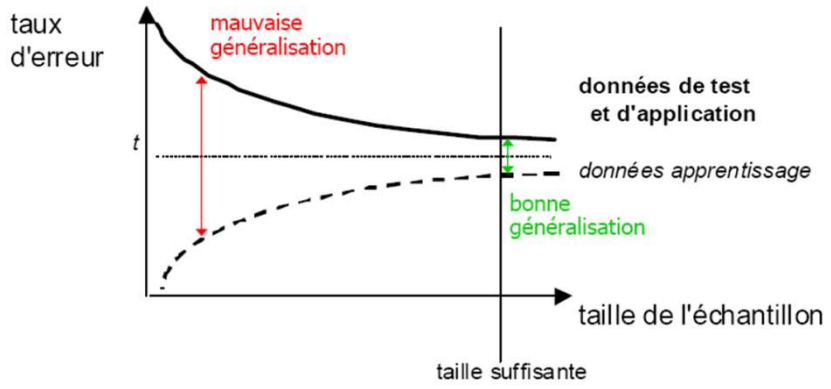
Nom	Rang	Année	Titulaire
Mary	Assistant Prof	3	non
James	Assistant Prof	7	oui
Bill	Professor	2	oui
John	Associate Prof	7	oui
Mark	Assistant Prof	6	non
Annie	Associate Prof	3	non

Validation -exemple



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

Taux erreurs vs données d'apprentissage



Critères de validation

- Taux d'erreur (Accuracy)
- Temps d'exécution (construction, utilisation)
- Robustesse (bruit, données manquantes,...)
- Interprétabilité
- Simplicité

Arbres de décision

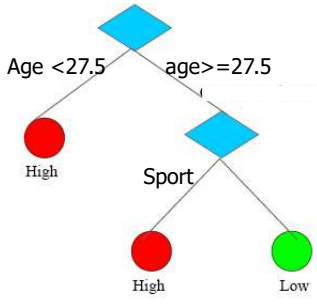
- Génération d'arbres de décision à partir des données
- Arbre = Représentation graphique d'une procédure de classification
- Un arbre de décision est un arbre:
  - Noeud interne = un attribut
  - Branche d'un noeud= un test sur un attribut
  - Feuille = classe donnée

Exemple 1

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

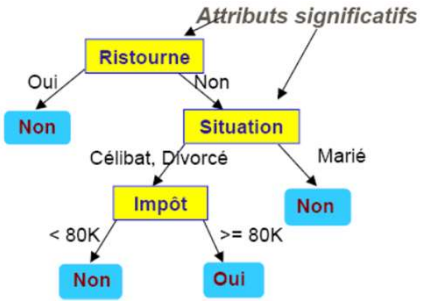


Age=40, CarType=Family ⟹ Class=Low

Exemple 2

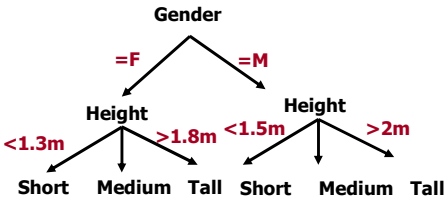
Détection de fraudes fiscales

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



Exemple 3

Name	Gender	Heigh	Output1
Kristina	F	1.6m	Short
Jim	M	2m	Tall
Maggie	F	1.9m	Meduim
Martha	F	1.88	Meduim
Stephanie	F	1.7m	Short
Bob	M	1.85m	Meduim
Khaty	F	1.6m	Short
Dave	M	1.7m	Short
Worth	M	2.2m	Tall
Steven	M	2.1m	Tall
Debbie	F	1.8m	Meduim
Todd	M	1.95m	Meduim
Kim	F	1.9m	Meduim
Amy	F	1.8m	Meduim
Wynette	F	1.75m	Meduim



Génération d'un arbre de décision

- Deux phases dans la génération de l'arbre :
  - Construction de l'arbre
    - Arbre peut atteindre une taille élevée
  - Élaguer l'arbre (**Pruning**)
    - Identifier et supprimer les branches qui représentent du "bruit"
    - Réduire le taux d'erreur

Construction de l'arbre (1/2)

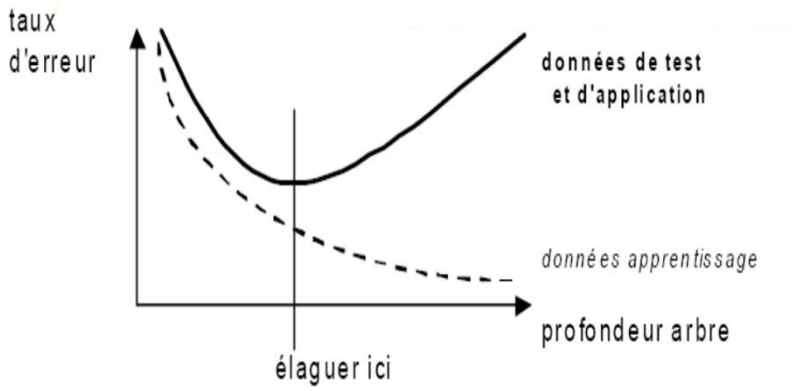
- Au départ, toutes les instances d'apprentissage sont à la racine de l'arbre
- Sélectionner un attribut et choisir un test de séparation (split) sur l'attribut, qui sépare le "mieux" les instances.
- La sélection des attributs est basée sur une fonction à définir.
- Partitionner les instances entre les noeuds fils suivant la satisfaction des tests logiques
- Traiter chaque noeud fils de façon récursive

Construction de l'arbre (2/2)

- Répéter jusqu'à ce que tous les noeuds soient des terminaux (feuilles). 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,
  - La profondeur de l'arbre a atteint une limite fixée
  - le nombre de feuilles a atteint un maximum fixé
  - Le noeud est "presque pur",i.e. la majorité des instances appartiennent à une seule classe(Ex : 95%)
  - Nombre minimum d'instances par branche
- Étiqueter le noeud terminal par la classe majoritaire

Elagage de l'arbre

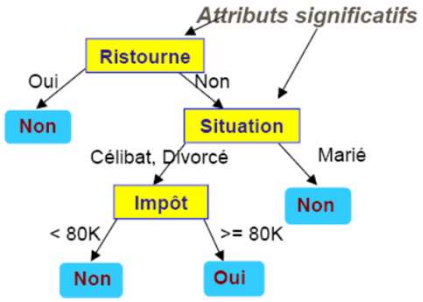
- Supprimer les sous-arbres qui n'améliorent pas l'erreur de la classification (accuracy) -> arbre ayant un meilleur pouvoir de généralisation, même si on augmente l'erreur sur l'ensemble d'apprentissage
- Eviter le problème de sur-spécialisation (overfitting),i.e., on a appris "par coeur" l'ensemble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



Un bon arbre doit être élagué pour éviter la remontée du taux d'erreur dû au sur-apprentissage

Arbre de décision et règles si-alors

- Une règle est générée pour chaque chemin de l'arbre (de la racine à une feuille)
- Les paires attribut-valeur d'un chemin forment une conjonction
- Le nœud terminal représente la classe prédite
- Les règles sont généralement plus faciles à comprendre que les arbres



Si *Ristourne* alors fraude=Non
Si *non Rirtourne* et *marié* alors fraude=non
...

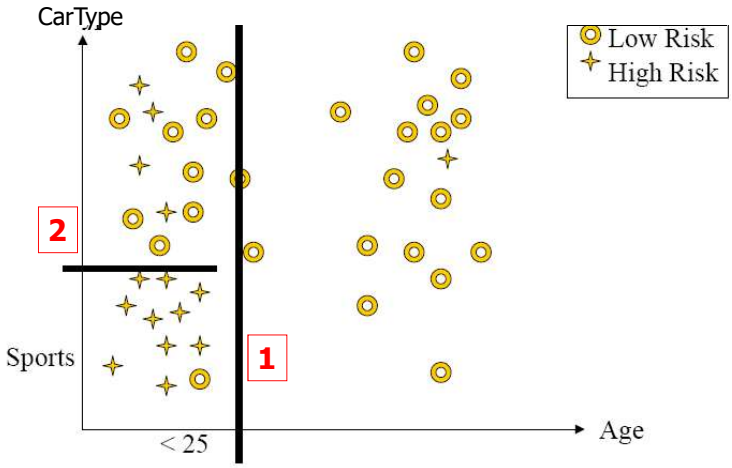
Algorithmes

- Les principaux algorithmes de construction d'arbres sont:
  - ID3 (données discretes)
  - C4.5 et C5 (+ID3 : données continues, missing data, pruning, etc.)
  - CART (données énumeratives, discretes, continues, arbres binaires)

- **Algorithme général Construction de l'arbre**

```
Input
D    // training data
Output
T   // Decision tree
DTBuild algorithm
T=∅
Determine best splitting criterion
T= Create root node and label with splitting attribute
T= Add arc to root node for each split predicate an
  label
For each arc do
D=Database created by applying splitting predicate to D
If stopping point reached for this path then
 T'= Create leaf node an label with appropriate class
Else
T'= DTBuild(D)
T= Add T' to arc
```

Bonne sélection



ID3

- Développé par Quinlan en 1986
- ID3 utilise le concept de l'entropie pour calculer la quantité d'information d'un événement à l'aide de sa probabilité d'occurence:

plog(1/p)

  - **Si p tend vers 1 alors l'événnement est très probable, donc peu d'information si l'événement se réalise**
  - **Si p tend vers 0 alors l'événnement est peu probable, donc beaucoup d'information si l'événement se réalise**

Gain d'information

- Sélectionner l'attribut qui sépare le mieux un ensemble d'individus classés
- C'est celui dont les valeurs divisent l'ensemble des individus à des sous-ensembles « purs » formés chacun par des individus homogènes (ayant la même classe)
- Plus grand gain d'information

Gain d'information

- Soit X un attribut à deux valeurs A1 et A2
- L'ensemble des individus est scindé en deux S1 et S2
  - S1 individus ayant X=A1, contient N1 individus
  - S2 individus ayant X=A2, contient N2 individus
- Etant donné un individu U de S1, l'information nécessaire pour déterminer sa classe dépendra de la répartition des individus de S1 en classes C1, C2,.., Cn
- Répartition granulaire => information nécessaire importante

- La qualité de la répartition est mesurée par l'entropie

$$H(S1) = \sum_{i=1}^{c_{S1}} \frac{\#C_i}{N1} \log_2 \left(\frac{N1}{\#C_i}\right)$$

  - c_{s1} : nombre de classes dans S1
  - $\#C_i$: nombre d'individus dans C_i
  - H(S1) mesure l'impureté de S1
- Plus l'entropie de S1 est grande => l'information nécessaire est importante => pas de gain d'information
- ID3 choisit l'attribut qui minimise l'entropie associée à un attribut X

$$H(X) = \sum_{i=1}^{N_X} H(S_i)$$

  - N_x: nombre de valeurs distincts de X

ID3 -exemple

Name	Gender	Heigh	Output1
Kristina	F	1.6m	Short
Jim	M	2m	Tall
Maggie	F	1.9m	Meduim
Martha	F	1.88	Meduim
Stephanie	F	1.7m	Short
Bob	M	1.85m	Meduim
Khaty	F	1.6m	Short
Dave	M	1.7m	Short
Worth	M	2.2m	Tall
Steven	M	2.1m	Tall
Debbie	F	1.8m	Meduim
Todd	M	1.95m	Meduim
Kim	F	1.9m	Meduim
Amy	F	1.8m	Meduim
Wynette	F	1.75m	Meduim

- Deux attributs:
  - Gender: énumératif
  - Heigh: numérique

- Attribut=Gender
  - Ayant F comme Gender 3/9log(9/3) + 6/9log(9/6) = 0,2764
  - Ayant M comme Gender : 1/6log(6/1)+2/6log(6/2)+3/6log(6/3)=0,4392
  - H(Gender)= [9/15*0.2764 + 6/15*0.4392]=0,09688

- Attribut=height
  - Discrétisation : (0, 1.6], (1.6, 1.7], (1.7, 1.8], (1.8, 1.9], (1.9, 2.0], (2.0,∞)
  - H((1.9, 2.0])=0.301
  - H(Height)=2/15 *0.301=O.3983

Attribut élu : Gender

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
- 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

Prédiction

Prédiction/Régression ?

- La prédiction consiste à estimer la valeur d'une variable continue (dite « à expliquer », « cible », « réponse », « dépendante » ou « endogène ») en fonction de la valeur d'un certain nombre d'autres variables (dites « explicatives », « de contrôle », « indépendantes » ou « exogènes »)

- Cette variable « cible » est par exemple :
  - le poids (en fonction de la taille)
  - la taille des ailes d'une espèce d'oiseau (en fonction de l'âge)
  - le prix d'un appartement (en fonction de sa superficie, de l'étage et du quartier)
  - la consommation d'électricité (en fonction de la température extérieure et de l'épaisseur de l'isolation)

Techniques

- **Deux types de techniques:**

  - **Techniques 'Paramétrique'**: fixent une forme de la fonction de prédiction et utilisent des techniques du statistique ou du mathématique pour l'approximation de cette fonction: **Régression Linéaire, Interpolation Polynomiale**, etc.

  - **Techniques 'Non-Paramétrique'**: ne stipulent aucune forme pour la fonction de prédiction et utilisent les techniques de ML pour l'approximation de cette function: **Réseaux de neurons, Support Vector Regression, Case-Based Reasonning**, etc.

    => **Approximateurs Universels**

Régression linéaire

- **La variable à expliquer s'ecrit comme combinaison linéaire de variables explicatives**

$$y = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_M x_M$$

- **Détermination des paramètres a_i est accomplie en utilisant les techniques de régression linéaire simple ou multiple**
- **Modèles linéaires simples**

$$y = A + (B \times x)$$

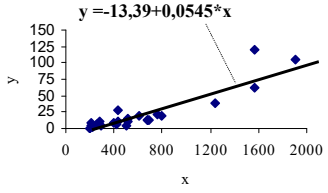
  - **Détermination de A et B par la descente de la gradient**

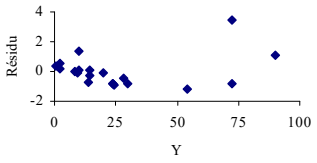
$$F(a,b) = \sum_{i=1}^{n} e_i^{\,2} = \sum_{i=1}^{n} (y_{i,\text{réel}} - A - B \times x_i)^2$$

- **Tests de la qualité de l'ajustement d'un modèle linéaire**
  - **Coefficient de détermination R^2**
    - **Le carré du coefficient de corrélation linéaire entre Y_i et Y^*_i**

$$R^2 = \frac{\sum (y_i - \bar{y})^2 - \sum (y_i - y_i^*)^2}{\sum (y_i - \bar{y})^2}$$

    - **Entre 0 et 1**
    - **Une valeur proche de 1 implique que l'ajustement est meilleure**
  - **Distribution des résidus e_i indépendamment des deux variables y et x et selon une loi gaussienne centrée et réduite**





R^2=0,874

- **Transformation des deux variables y et x pour linéariser la relation y/x**
  - **Transformation logarithmique**

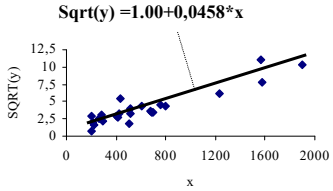
$$y = B \times x^{C}$$

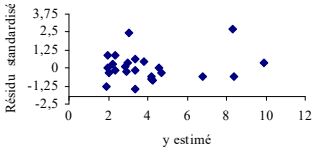
$$\log(y) = \log(B) + C \times \log(x)$$

$$\sum (\log(y_{réel}) - \log(B) - C \times \log(x))^2$$

  - **Autres transformations**
    - ✓ **X=racine(X)**
    - ✓ **X=1/X**
    - ✓ **X=log(X/1-X)**

- **Exemple: transformations non-logarithmiques**
  - **y → racine (y)**





R^2=0,899

⊙ Modèles linéaires multiples

$$y = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_M x_M$$

- **Détermination des constantes a_i**

$$\sum_{i=1}^{n} e_i^{\,2} = \sum_{i=1}^{n} (y_{i,réel} - a_0 - a_1 x_1^i - a_2 x_2^i - .. - a_M x_M^i)^2$$

⊙ Tests de la qualité d'ajustement

- **Coefficient de détermination R^2**
- **Distribution des résidus e_i**
- **Mesurer la multicolinéarité entre les variables explicatives x_i.**
  - **Facteurs d'inflation de la variance**
  - **Valeurs propres de la matrice de corrélation entre les x_i**
- **Détermination du sous ensemble des x_i qui donnent des estimations satisfaisantes de y**
  - **Ajout successif (celui qui augmente le plus le R^2) ou élimination successive (celui qui réduit le moins R^2)**
  - **Stepwise régression : effectue des tests de signification pour ne pas introduire une variable non significative et éliminer éventuellement des variables déjà introduites**

Régression linéaire: Categorical Data

- **Variable catégorique prends des valeurs linguistiques (qualifications): Gender (F, M), Situation familiale (Single, Mariée, Divorcée, Veuve)**
- **Dummy variables: ont deux valeurs linguistiques (deux catégories) qu'on code par 0 et 1: 1** signifie que l'instance appartient à la catégorie**, 0** n'appartient pas
- **Exemple 1: Gender a deux catégories: Female et Male**
  - **On crée une Dummy variable: Gender-Female pour indiquer si un Individu est Female ou Non**
    - ✓ **Gender-Female = 1** si l'Individu est **Female**
    - ✓ **Gender-Female = 0 sinon** c.a.d. l'Individu est **Male**

Situation Familiale	Gender	Heigh	Poids (Kg)
Single	F	1.6m	68,2
Single	M	2m	96,4
Marie	F	1.9m	72,5
Marie	F	1.88	66,8
Veuve	F	1.7m	58,3
Divorce	M	1.85m	75,6
Single	F	1.6m	60,3
Marie	M	1.7m	82,5
Divorce	M	2.2m	95,0
Veuve	M	2.1m	90,2
Single	F	1.8m	70,1
Marie	M	1.95m	91,1
Divorce	F	1.9m	78,3
Single	F	1.8m	72,4
Marie	F	1.75m	68,4

Situation Familiale	Gender-Female	Heigh	Poids (Kg)
Single	1	1.6m	68,2
Single	0	2m	96,4
Marie	1	1.9m	72,5
Marie	1	1.88	66,8
Veuve	1	1.7m	58,3
Divorce	0	1.85m	75,6
Single	1	1.6m	60,3
Marie	0	1.7m	82,5
Divorce	0	2.2m	95,0
Veuve	0	2.1m	90,2
Single	1	1.8m	70,1
Marie	0	1.95m	91,1
Divorce	1	1.9m	78,3
Single	1	1.8m	72,4
Marie	1	1.75m	68,4

- **Dummy variable: Gender-Female**
- **La catégorie Male sera la catégorie de référence (Base category)**

$$\text{Poids} = A + (B \times \text{Gender - Female})$$

$$\text{Si Individu est Female} \Rightarrow \text{Poids} = A + B$$

$$\text{sinon Poids} = A$$

- **Exemple 1: Situation Familiale a 4 catégories: Single, Mariée, Divorcée, Veuve.**
  - **On crée 3 Dummy variables:**
    - ✓ **On choisit Veuve comme la catégorie de référence**
    - ✓ **D1: ST-Single =1** si l'Individu est **Single, 0** sinon
    - ✓ **D2: ST-Mariee =1** si l'individu est **Marié, 0** sinon
    - ✓ **D3: ST-Divorcee =1** si l'Individu est **Divorcé, 0** sinon

Situation Familiale	ST-Single	ST-Mariee	ST-Divorce
Single	1	0	0
Single	1	0	0
Marie	0	1	0
Marie	0	1	0
Veuve	0	0	0
Divorce	0	0	1
Single	1	0	0
Marie	0	1	0
Divorce	0	0	1
Veuve	0	0	0
Single	1	0	0
Marie	0	1	0
Divorce	0	0	1
Single	1	0	0
Marie	0	1	0

$$Poids = A + (B * ST - Single) + (C * ST - Mariee) + (D * ST - Divorcee)$$

Si Individu est Single $\Rightarrow$ Poids $= A + B$

Si Individu est Marie $\Rightarrow$ Poids $= A + C$

Si Individu est Divorce $\Rightarrow$ Poids $= A + D$

Si Individu est Veuf $\Rightarrow$ Poids $= A$

- **Règle générale: V une variable catégorique avec K catégories, il faut K-1 Dummy variables pour représenter V**

- **Critères de performance**
  - **Root Mean Squared Error**

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}$$

  - **Mean Absolute Error**

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i - x_i|$$

  - **Magnitude Relative Error**

$$MRE = \left|\frac{Y_{réel} - Y_{estimé}}{Y_{reél}}\right| \times 100 \qquad MMRE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{Y_{i,réel} - Y_{i,estimé}}{Y_{i,reél}}\right| \times 100$$

  - **Prediction Level**

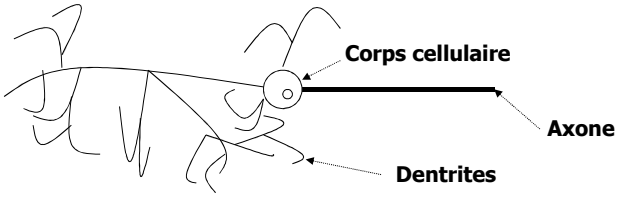
$$\Pr ed(p) = \frac{K}{N} * 100$$

    - K: nombre de prédictions avec un MRE <= p, p<=25
    - N: Nombre total de prédictions

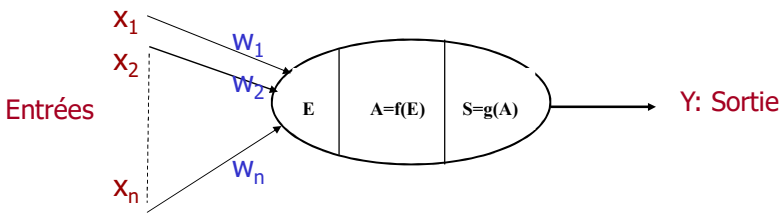
Réseaux de neurones

- **Les réseaux de neurones regroupent certains modèles dont l'intention est d'imiter certaines des fonctions du cerveau humain en reproduisant certaines de ses structures de base.**
- **Historique et événements**
  - **McCulloch et Pitts, 1943**
  - **Hebb, 1949**
  - **Rosenblatt, 1958**
  - **Minsky et Papert , 1969**
  - **Grossberg, Adaptative Resonance Theory, 1980**
  - **Hopfield, mémoires auto-associatives, 1982**
  - **Kohonen, Self-Organized Maps, 1982**
  - **Rumelhart et McClelland, Backpropagation, 1986**

Neurone?



- Nombre de neurones (humain) ~10^{10}
- Connexions (synapses) par neurone : ~10^4–10^5



Neurone ? *(suite)*

- Fonction d'entrée

$$E = \sum_{i=1}^{i=n} w_i x_i$$

- Fonction d'activation
  - Fonction binaire à seuil
  - Fonction sigmoïde

$$y = \frac{1}{1+e^{-x}}$$

  - Fonction gaussienne

$$y = e^{\left(-\frac{\|x-c_j\|^2}{\sigma_j^2}\right)}$$

Topologies ANNs

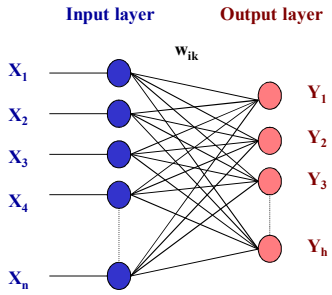
- **Un ANN est caractérisé par trois éléments:**
  - **Une architecture (ensemble de neurones, typologie de connexions)**
  - **Une procédure d'apprentissage**
  - **Des fonctions d'activation**
- **Architecture**
  - **ANNs feedforward: pas de boucle dans la typologie de connexions**
  - **ANNs feedback (récurrents): existence de boucles dans la typologie de connexions**
- **Apprentissage**
  - **Supervisé: compare les sorties du réseau avec les sorties réelles et propage l'erreur afin d'ajuster les poids de connexions**
    - ✓ Plausible quand les valeurs réelles sont disponibles
    - ✓ Facile à utiliser (cas du Backpropagation)
    - ✓ Mais grand consommateur du temps de calcul

Topologies ANNs *(suite)*

- **Non-supervisé: ajuster les poids dépendamment des caractéristiques, des régularités, des corrélations, ou des catégories découvertes dans les exemples d'apprentissage**
  - ✓ Utile dans le cas de non disponibilité des valeurs réelles
  - ✓ Adaptés aux problèmes de clustering
  - ✓ N'est pas facile à implanter et à contrôler sa convergence
- **Fonctions d'activation**
  - **Fonction binaire à seuil**
  - **Fonction linéaire à seuil**
  - **Fonction sigmoide**
  - **Etc.**
- **Modèles de réseaux de neurones**
  - **Perceptron simple**
  - **Perceptron multicouches**
  - **Radial Basis Functions (RBF)**
  - **Kohonen**
  - **Etc.**

Perceptron simple

- Développé par Rosenblatt en 1958
- Composé de deux couches: une couche des entrées (X_k) et une couche des sorties (Y_i)

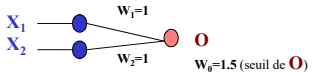


$$Y_i = g(\sum_k w_{ik} x_k)$$

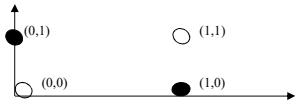
Perceptron simple (suite)

- La fonction d'activation à seuil
- Il peut implanter certaines fonctions logiques: (cas du AND)

0	0	0
0	1	0
1	0	0
1	1	1



- Il peut implanter tous les problèmes qui sont linéairement séparables
- Le contre-exemple du XOR



- Règle d'apprentissage (inspirée de la règle de Hebb)

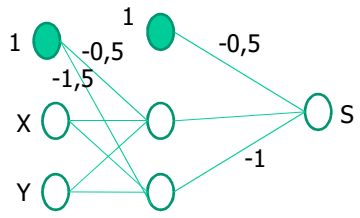
$$\Delta w_{ik} = \eta(Y_i - O_i)x_k$$

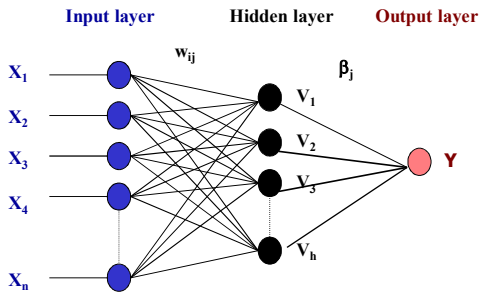
Perceptron multicouches

- **Il est une généralisation du Percetron simple afin d'éviter la limite des problèmes non linéairement séparables**
- **Il a une grande capacité pour traiter des problèmes complexes:**
  - ✓ Avec une seule couche, il peut représenter n'importe quelle fonction logique
  - ✓ Avec une seule couche cachée, il peut approximer toute fonction continue
- **Sa popularité provient du célèbre algorithme d'apprentissage: Backpropagation**
- **Backpropagation fait de l'apprentissage supervisé**
- **Il exige que les fonctions d'activation soient dérivables, souvent on considère la fonction sigmoide**

XoR et MLP

- **MLP:** Les poids non mentionnés sont a 1. La fonction d'activation est la fonction a seuil 0,4. Appliquer et interpréter ce réseau sur les entrés X : (0 ou 1) et Y (0 ou 1).





$$Y = \sum_{j=1}^{h} V_j \beta_j \qquad avec \qquad V_j = f\left(\sum_{i=1}^{n} w_{ij} x_i\right)$$

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

MLP - Construction

- Nombre de neurones de la couche d'entrée et la couche de sortie :
  - Entrées : correspond à la dimension des données du problème ou leur codage
    - Attributs continues : normalisation entre 0 et 1
    - Attributs énumératifs ou discrets : codage en binaire par exemple
  - Sorties : nombre de variables à prédire
- Nombre de couches cachées : une en général
- Nombre de neurones couche cachée : 2 à nombre de neurones d'entrée
- Les poids sont générés au début aléatoirement

Algorithme Backpropagation

1- On initialise les poids w_{ij} et β_j à des petites valeurs

2- Après chaque présentation d'un exemple, on calcule l'erreur
$E= 1/2(Y-O)^2$

3- Mise à jour des poids de la couche caché à la couche de sortie:
$\Delta\beta_j = \eta\ \delta Vj$ avec $\delta=(Y-O)O$,
η taux d'apprentissage choisi dans [0,1]

4- Mise à jour des poids de la couche d'entrée à la couche cachée
$\Delta w_{ij} = \eta\ \beta_j \delta f'(I_j)X_i$, I_j est l'entrée du neurone j

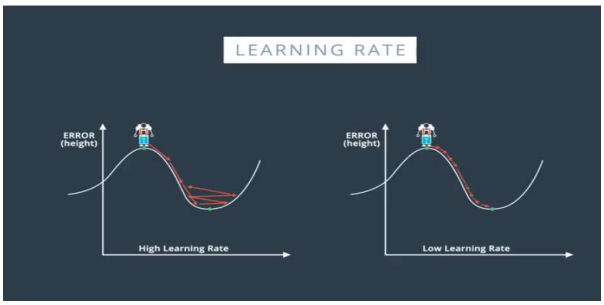
5- On reprend les étapes 2, 3, 4 et 5 avec tous les exemples

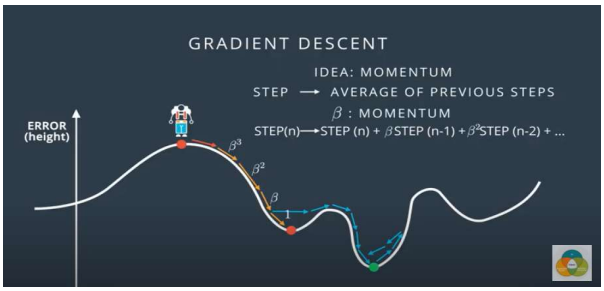
6- Si l'erreur cumulée E est inférieure à un seuil choisi, on arrête l'apprentissage. Sinon on reprend un autre cycle d'apprentissage

ANN Hyper Parameters

- **Epoch**: One Epoch is when an ENTIRE dataset (training) is passed forward and backward through the neural network only ONCE.
- **BatchSize:** is the number of instances processed before the model is updated (1->N)
  - Batch Gradient Descent: Batch Size = Size of Training Set
  - Stochastic Gradient Descent: Batch Size = 1
  - Mini-Batch Gradient Descent: 1 < Batch Size < Size of Training Set.

⊙ Learning rate and Momentum





Limites MLP et Backpropagation

- Problème des minima locaux. Certains précautions pour éviter:
  - Choisir des poids petits
  - Choisir un taux d'apprentissage petit, souvent entre 0 et 1
  - Changer les poids après chaque présentation et non pas après avoir passer toutes les présentations
  - Il ne faut pas adopter la même séquence des exemples d'apprentissage
- Il n'y a aucune règle pour le choix du nombre de couches ainsi que le nombre de neurones par couche
- L'algorithme Backpropagation est un gros consommateur du temps de calcul
- ✓ Backpropagation et le 'credit-assignment problem'?

ANNs- Avantages et limites

- **Avantage 1: Apprentissage**
  - Importance de l'apprentissage dans notre intelligence
  - L'apprentissage est adéquat pour traiter les situations dans lesquelles la connaissance et les besoins ne sont pas définis d'une manière détaillée (cas d'un robot)
  - Les réseaux de neurones offrent des mécanismes d'apprentissage automatiques avec les solutions qu'ils proposent
    - ✓ Perceptron simple avec la règle de Widrow-Hoff
    - ✓ Perceptron multicouche avec le Backpropagation
    - ✓ Perceptron avec apprentissage compétitif
  - Etc.

- **Avantage 2: Parallélisme**
  - Les réseaux de neurones adoptent des architectures qui permettent un traitement parallèle de l'information (Brain-like computation)
  - Profiter de la grande puissance de calcul de toutes les machines disponibles (PVM, MPI)
  - Rendre le système robuste
  - Traiter les problèmes complexes nécessitant une solution modulaire
- **Avantage 3: Approximateurs universels**
  - Ils peuvent approximer n'importe quelle fonction: Cas d'un Perceptron à trois couches avec toutes les fonctions continues

⊙ Limite 1: Boîte noire

- **On ne peut expliquer le processus mis en œuvre pour générer des sorties à partir des entrées**
- **Cet inconvénient se manifeste quand la sortie générée est différente de la sortie attendue**
- **Certains domaines d'application nécessitent que l'approche utilisée dans la modélisation soit facilement interprétable (sécurité, prise de décision)**
- **On ne peut corriger une erreur facilement dans une modélisation par les réseaux de neurones**
- **Certains ANNs peuvent être considérés plus au moins facilement interprétables**
  - ✓ Perceptron simple linéaire
  - ✓ Réseau de Kohonen avec chapeau mexicain
  - ✓ Radial Basis Function Networks

⊙ Limite 2: Performances de Généralisation

- **La généralisation est la capacité d'un ANN à donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir desquels il a appris**
- **Le NN doit apprendre la fonction implicite qui existe entre les exemples d'apprentissage et leurs sorties**
- **Problème d'un apprentissage très poussé (overfitting)**
- **La qualité d'un apprentissage dépend de :**
  - ✓ L'algorithme de l'apprentissage adopté
  - ✓ Le nombre d'exemples utilisés dans l'apprentissage
  - ✓ Le temps alloué à l'apprentissage
  - ✓ Du problème traité
- **Quand l'apprentissage doit s'arrêter?**

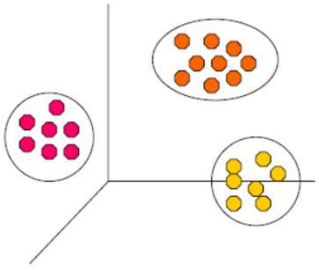
Clustering
Segmentation

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

Objectif et Techniques

- Découverte de clusters (groupes) d'instances de la base de données
- Clusters : groupes d'instances ayant des caractéristiques similaires
- Apprentissage non supervisé (clusters inconnues)
- Techniques : **partitionnement** et **hiérarchique**



Regroupement géographique

Domaine d'application

- Marketing : segmentation du marché en découvrant des groupes de clients distincts à partir de bases de données d'achats.
- Environnement : identification des zones terrestres similaires(en termes d'utilisation) dans une base de données d'observation de la terre.
- Assurance : identification de groupes d'assurés distincts associés à un nombre important de déclarations.
- Planification de villes : identification de groupes d'habitations suivant le type d'habitation, valeur, localisation géographique, …
- Médecine: Localisation de tumeurs dans le cerveau
  - ✓ Nuage de points du cerveau fournis par le neurologue
  - ✓ Identification des points définissant une tumeur

Qualité d'un clustering

- Une bonne méthode de clustering produira des clusters d'excellente qualité avec
  - Similarité intra-cluster $\sum_{i=1}^{c}\sum_{j=1}^{ki} d(x_{j,i}, \hat{x}_{i)}$
  - Similarité inter-clusters $\sum_{i=1}^{c} d(\hat{c}_{i,}\hat{c})$
- La qualité d'un clustering dépend de
  - La mesure de similarité utilisée
  - Technique utilisée
- La qualité d'une méthode de clustering est évaluée par son habilité à découvrir tous les "clusters" cachés.

Techniques de clustering: Caractéristiques

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

K-means

147

- Mis en oeuvre par Duda et Hart en 1973
- Grouper les exemples similaires dans des clusters en utilisant une métrique distance (distance euclidienne).
- Les centres des clusters se calculent par la moyenne arithmétique des exemples affectés au cluster.
- Le nombre c de clusters est déterminé a priori.
- Initialement, les centres des clusters sont définis d'une façon aléatoire.
- Objectif : minimiser $J = \sum_{i=1}^{c} \sum_{x \in C_i} d(x, c_i)$

Algorithme K-means

148

- Entrée: un échantillon de m exemples $\mathbf{x_1}, \ldots, \mathbf{x_m}$
- Chaque exemple x_i est décrit par M attributs

1- Choisir k centres initiaux $\mathbf{c_1}, \ldots, \mathbf{c_k}$

2- Répartir chacun des m exemples dans le groupe i dont le centre ci est le plus proche

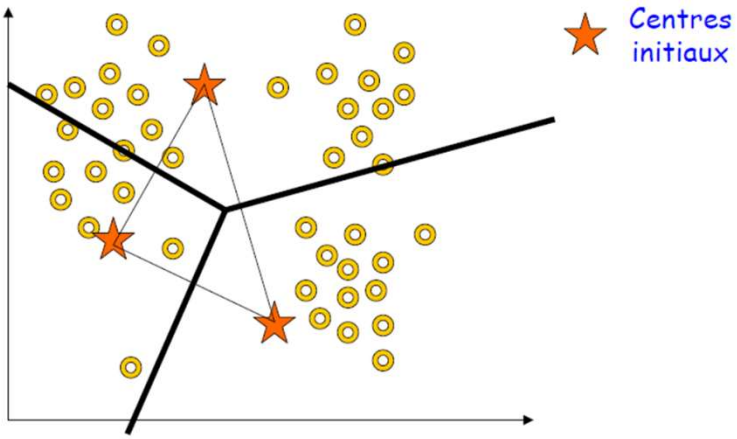
3- Si aucun élément ne change de groupe ou une autre condition d'arrêt est satisfaite alors arrêt et sortir les groupes

4- Calculer les nouveaux centres : pour tout i, ci est la moyenne des éléments du cluster i

5- Aller a 2.

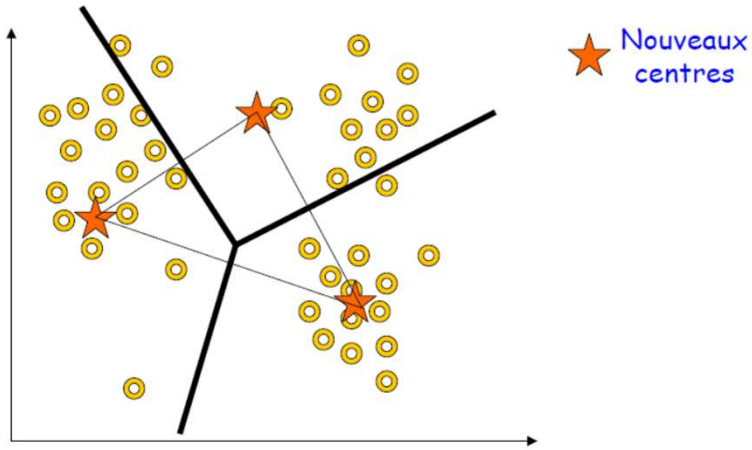
149

K-means : Illustration

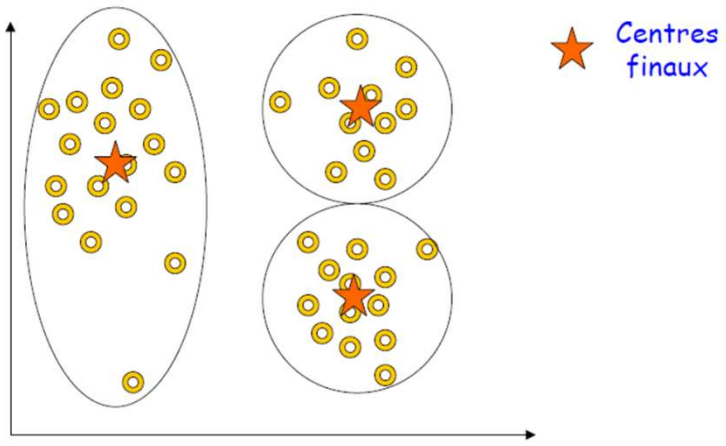


150

K-means : Illustration



K-means :Illustration



K-means : Exemple

Prendre K=2et effectuer les deux premières itérations du K-means

	Age	Salaire
P1	29	11000
P2	40	7200
P3	27	10000
P4	38	8000
P5	31	12000
P6	35	9000

K-means : Limitations

- Applicable seulement dans le cas où la moyenne des objets est définie. Pour les variables qualitatives: **K-modes**. Pour les variables mixtes: **K-prototypes**
- 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 ?)

Méthode Hiérarchique

- 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)

Méthode Hiérarchique

- Entrée: un échantillon de m enregistrements x1, …, xm

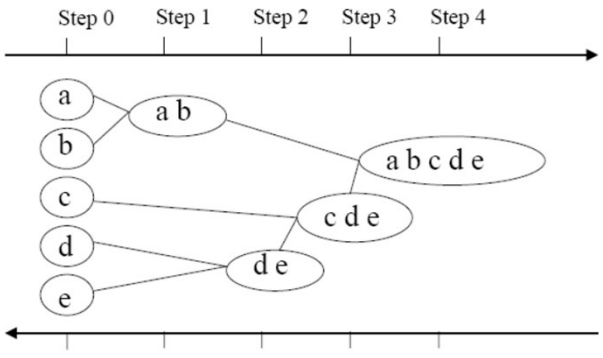
1. On commence avec m clusters (cluster = 1 enregistrement)

2. Grouper les deux clusters les plus «proches» (minimum de distance, méthode de Ward, etc.)

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

4. Aller a 2.

Méthode Hiérarchique

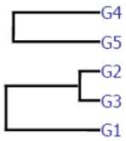


Méthode Hiérarchique

	G1	G2	G3	G4	G5	G6
G1	0					
G2	0,1	0				
G3	0,16	0,03	0			
G4	0,9	0,8	0,5	0		
G5	0,25	0,4	0,77	0,12	0	
G6	1,3	1,5	0,95	1,0	1,1	0

	G1	G2-G3	G4	G5	G6
G1	0				
G2-G3	0,13	0			
G4	0,9	0,65	0		
G5	0,25	0,58	0,12	0	
G6	1,3	1,22	1,0	1,1	0

	G1	G2-G3	G4-G5	G6
G1	0			
G2-G3	0,13	0		
G4-G5	0,57	0,61	0	
G6	1,3	1,2	1,0	0



Clustering Hiérarchique

	G1	G2	G3	G4	G5	G6
G1	0					
G2	0,1	0				
G3	0,16	0,03	0			
G4	0,9	0,8	0,5	0		
G5	0,25	0,4	0,77	0,12	0	
G6	1,3	1,5	0,95	1,0	1,1	0

Simple

	G1	G2-G3	G4	G5	G6
G1	0				
G2-G3	0,1	0			
G4	0,9	0,5	0		
G5	0,25	0,4	0,12	0	
G6	1,3	0,95	1,0	1,1	0

Complet

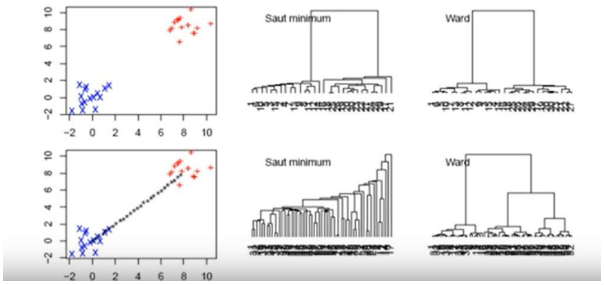
	G1	G2-G3	G4	G5	G6
G1	0				
G2-G3	0,16	0			
G4	0,9	0,8	0		
G5	0,25	0,77	0,12	0	
G6	1,3	1,5	1,0	1,1	0

Moyen

	G1	G2-G3	G4	G5	G6
G1	0				
G2-G3	0,13	0			
G4	0,9	0,65	0		
G5	0,25	0,58	0,12	0	
G6	1,3	1,22	1,0	1,1	0

Méthode Ward

- Problème de l'effet de chaîne



- Perte d'information à chaque étape de regroupement de deux clusters
- Initialement, chaque cluster est composé d'un seul xi => distance Intra-cluster=0 => somme Intra-cluster=0
- Choisir de regrouper les deux clusters A et B qui minimisent l'augmentation de la somme Intra-cluster (minimise la perte d'information) $\frac{n_A * n_b}{n_A + n_b} d^2(\hat{x}_A - \hat{x}_B)$

Méthode Hiérarchique

- 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

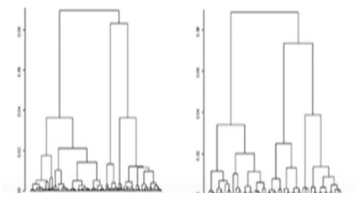
Méthode Hiérarchique

- + Conceptuellement simple
- + Propriétés théoriques sont bien connues
- + Quand les clusters sont groupés, la décision est définitive => le nombre d'alternatives différentes à examiner est réduit

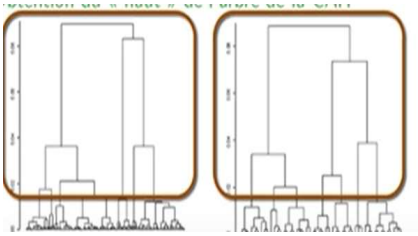
- - Groupement de clusters est définitif => décisions erronnées sont impossibles à modifier ultérieurement
- - Méthodes non extensibles à des ensembles de données de grandes tailles

Méthode Hiérarchique vs K-means

- La méthode Hiérarchique pourrait être utilisée pour générer les clusters initiaux
  - +Consolidation des clusters
  - - Perte de la hiérarchie
- Dans le cas d'un nombre important d'individus, K-means pourrait être utilisé pour réduire le nombre d'individus avant de commencer la méthode Hiérarchique



300 individus 50 clusters par K-means



Règles d'association

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)

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 (items set) achetés fréquemment (ensemble)
- Extraction d'informations sur le comportement de clients
  - SI achat de riz + limonade 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
  - Services des compagnies de télécommunication
  - Services bancaires
  - Traitements médicaux, …

Règles d'association

- **Recherche de règles d'association**

  Découvrir des patterns, corrélations, associations fréquentes, à partir d'ensembles d'items 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, Design de catalogue, etc.

Règles d'association

- **Formats de représentation des règles d'association :**
  - Poissons ⇒ limonade [50%, 60%]
  - achète: poissons ⇒ achète: limonade [50%, 60%]
  - **SI** achète poissons **ALORS** achète limonade dans 60% de cas. Les poissons et la limonade sont tous deux achetés dans 50% des transactions de la base de données."

- **Condition:** partie gauche de la règle
- **Conséquence:** partie droite de la règle
- **Support:** fréquence ("partie gauche **et** droite sont présentes ensemble dans la base")
- **Confiance:** ("si partie gauche de la règle est vérifiée, probabilité que la partie droite de la règle soit vérifiée")

Règles d'association

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

$$Support(A \Rightarrow B) = \frac{\#transactions \text{ avec A et B}}{\#\text{ total de transactions}}$$

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

$$Confiance(A \Rightarrow B) = \frac{\#transactions \text{ avec A et B}}{\#\text{ de transactions avec A}}$$

Exemple

TID	Items
1	Pain, lait
2	Limonade, couches, pain, œufs
3	Limonade, jus, couches, lait
4	Limonade, pain, couches, lait
5	Jus, pain, couches, lait

Couches + lait =>limonade

$$s = \frac{\#\mathrm{I}(\text{couches}, \text{lait}, \text{limonade})}{\#\text{instances}} = \frac{2}{5} = 0.4$$

$$c = \frac{\#\mathrm{I}(\text{couches}, \text{lait}, \text{limonade})}{\#\mathrm{I}(\text{couches}, \text{lait})} = 0.66$$

Recherche de règles

- Données d'entrée : liste d'achats
- Achat = liste d'articles

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

Recherche de règles

- **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

- 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 (1)=55%, Confiance(2) = 50% ; Confiance(3) = 67%
- On préfère la règle 3 à la règle 2.

Recherche de règles

- **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

Recherche de règles

- **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.

- Calculer le nombre d'occurrences de chaque article
- Calculer le tableau des co-occurrences pour les paires d'articles.
- Déterminer les règles de niveau 2 en utilisant les valeurs de support, confiance et amélioration.
- Calculer le tableau des co-occurrences pour les triplets d'articles.
- Déterminer les règles de niveau 3 en utilisant les valeurs de support, confiance et amélioration
- etc

Complexité

- Soient
  - m : nombre de transactions dans la BD
  - n : Nombre d'attributs (items) différents
- Complexité
  - Nombre de règles d'association : $O(n*2^{n-1})$
  - Complexité de calcul : $O(n*m*2^n)$
- Taille des tableaux en fonction de n 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	$1.7\ 10^{11}$	$4.2\ 10^{14}$

Algorithme Apriori
(Agrawal 93)

- 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 (confiance ≥MINCONF)
    - Une règle fréquente est retenue si et seulement si sa confiance c≥MINCONF
    - Exemple: ABCD fréquent
    - AB =>CD est retenue si sa confiance ≥MINCONF

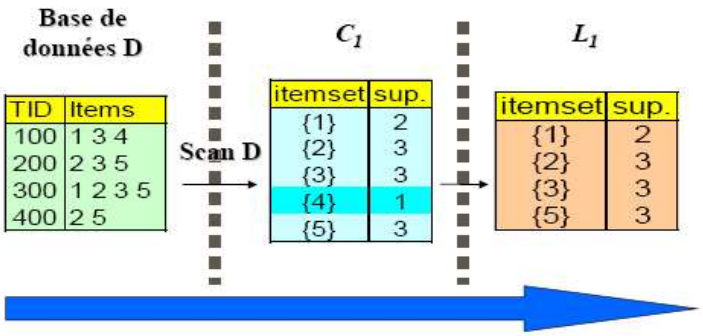
Exemple *(1/2)*

- Exemple
  - T = {AB, ABCD, ABD, ABDF, ACDE, BCDF}
  - I = {A, B, C, D, E, F}
  - 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}

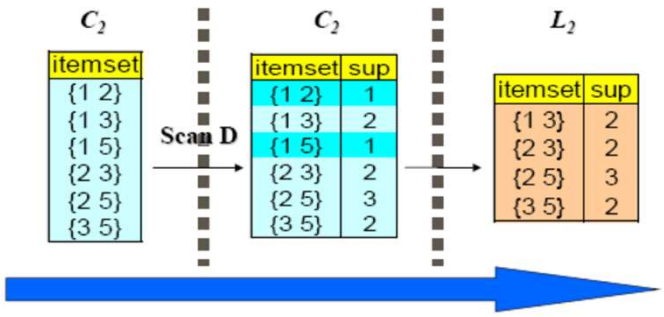
Exemple (2/2)

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

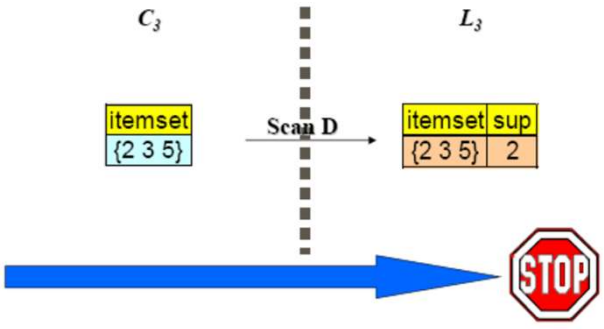
Exemple



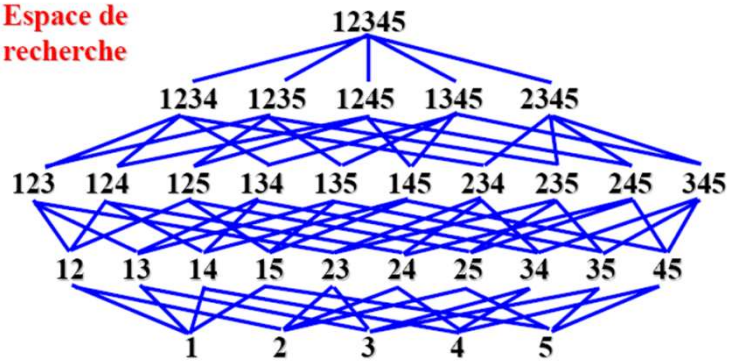
Exemple



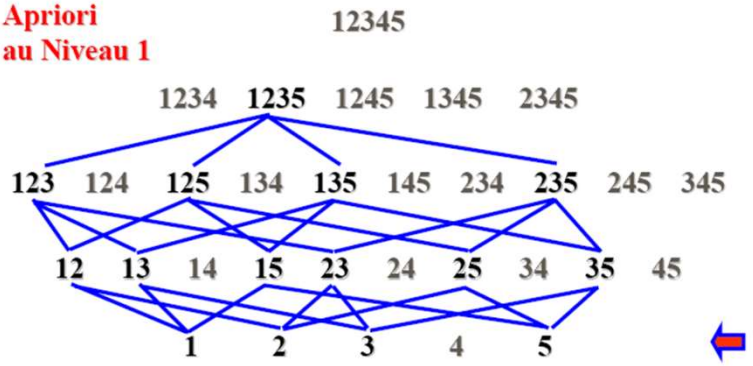
Exemple



Espace de recherche



Apriori au Niveau 1



Apriori au niveau 2