Classification

- Introduction
- k-NN
- Arbres de décision
- Réseaux baysiens
- Conclusion

1. Apprentissage supervisé

- Découverte de règles ou formules (patterns) pour ranger les données dans des classes prédéfinies
  - représentant un groupe d'individus homogènes
  - permettant de classer les nouveaux arrivants
- Processus en deux étapes
  - construction d'un modèle sur les données dont la classe est connue (training data set)
  - utilisation pour classification des nouveaux arrivants

Applications

- Marketing
  - comprendre les critères prépondérants dans l ’achat d ’un produit
  - segmentation automatique des clients pour le marketing direct
- Maintenance
  - aide et guidage d ’un client suite à défauts constatés
- Assurance
  - analyse de risques
- Isolation de populations à risques
  - médecine

2. k plus proches voisins (k-NN)

- Basé sur l'apprentissage par analogie
- Collection de tuples d'apprentissage
  - Xi=(x1i,x2i,…xni) (xji numérique) de classe connue
  - Représente un point dans l'espace à n dimensions
- Classes prédéfinies
  - C={C1,C2, …Cm}
- Distance et Similarité
  - Distance Euclidienne, Cosinus, etc.
  - Similarité = Max - Distance

Classement

- Soumission d'un tuple inconnu
- Recherche des k plus proches voisins
- Assignation de la classe la plus représentative parmi les k voisins
  - Vote majoritaire (classe la plus fréquente)
  - Plus grande similarité à la classe
  - …

Algorithme k-NN (WEKA : lazy/IBk)

```
Class (X) {
// Training collection T = {X1, X2, … Xn}
// Predefined classes C ={C1,C2, …Cm}
// Compute similarities
For i=1..N similar[i] = Max - distance(X,Xi);
SortDescending(similar[]);
kNN=Select k nearest neighbors with highest similarity;
// Calculer les scores des classes
score[Cj] = f(Cj, kNN) ;
Class(X) = Class Cj with highest score;
}
```

Dataset Iris

- Iris Setosa

- Iris Virginica

- Iris Versicolor

Forces et faiblesses

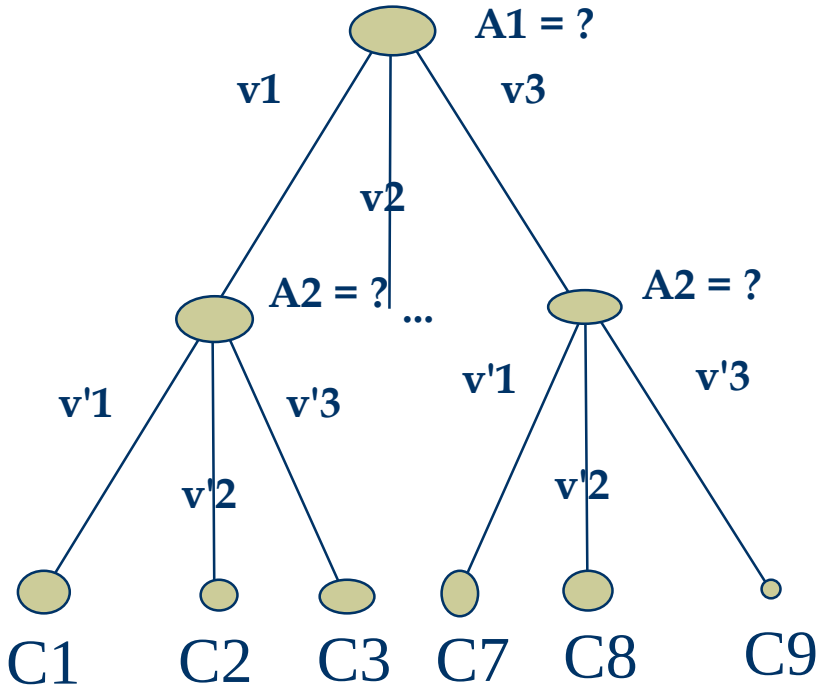
- Les attributs ont le même poids
  - centrer et réduire pour éviter les biais
  - certains peuvent être moins classant que d'autres
- Apprentissage paresseux
  - rien n'est préparé avant le classement
  - tous les calculs sont fait lors du classement
  - nécessité de technique d'indexation pour large BD
- Calcul du score d'une classe
  - peut changer les résultats; variantes possibles

3. Arbres de décision

- Définition
  - Arbre permettant de classer des enregistrements par division hiérarchiques en sous-classes
    - un nœud représente une classe de plus en plus fine depuis la racine
    - un arc représente un prédicat de partitionnement de la classe source
  - Un attribut sert d'étiquette de classe (attribut cible à prédire), les autres permettant de partitionner

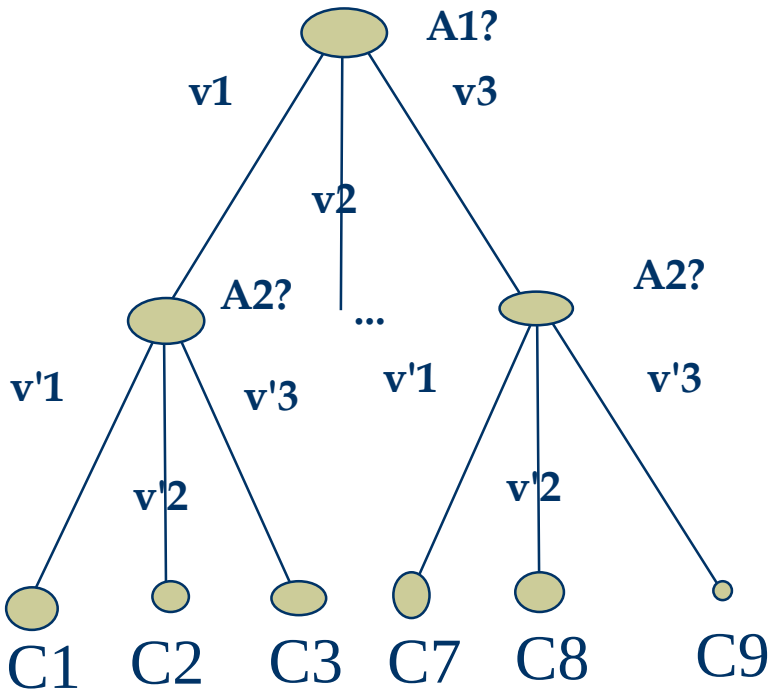
Génération de l'arbre

- Objectif:
  - obtenir des classes homogènes
  - couvrir au mieux les données
- Comment choisir les attributs (Ai) ?
- Comment isoler les valeurs discriminantes (vj) ?



Arbre = ensemble de règles

- (A1=v1)&(A2=v'1) → C1
- (A1=v1)&(A2=v'2) → C2
- (A1=v1)&(A2=v'3) → C3
- …
- (A1=v3)&(A2=v'1) → C7
- (A1=v3)&(A2=v'2) → C8
- (A1=v3)&(A2=v'3) → C9

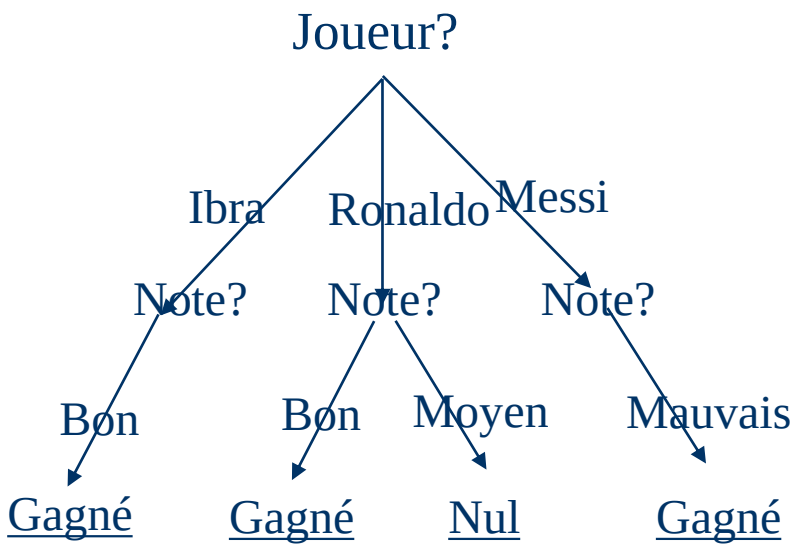


Exemple codant une table

Attributs ou variables

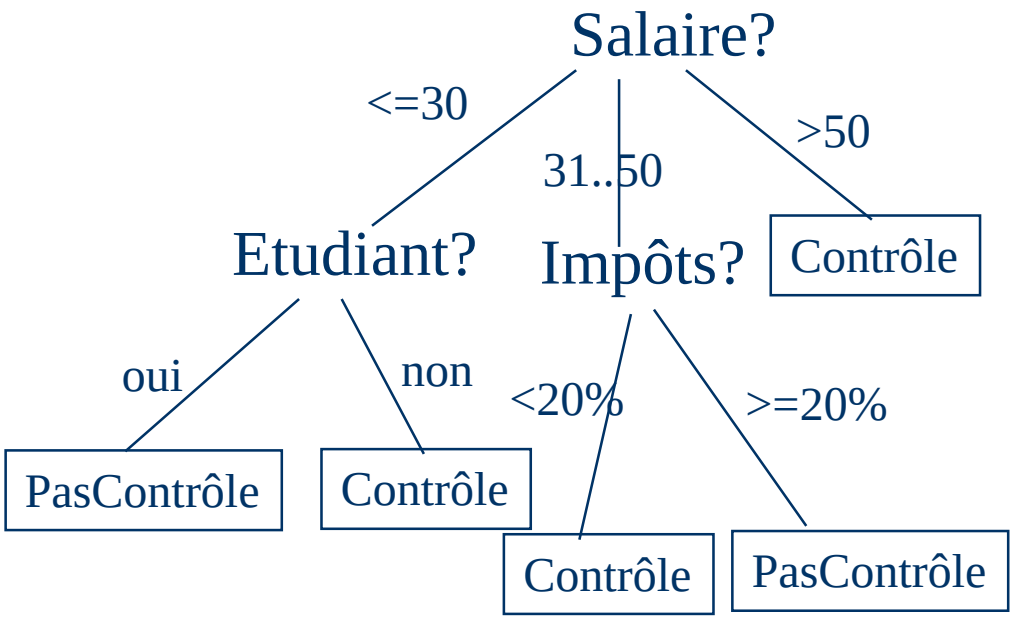
Joueur	**Note**	**Résultat**
Ronaldo	**Bon**	**Gagné**
Ronaldo	**Moyen**	**Nul**
Ibra	**Bon**	**Gagné**
Ibra	**Bon**	**Gagné**
Messi	**Mauvais**	**Gagné**

Classes cibles



Autre Exemple

- Faut-il vous envoyer un contrôleur fiscal ?

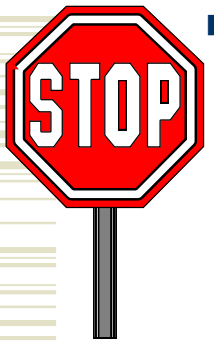


Procédure de construction (1)

- recherche à chaque niveau de l'attribut le plus discriminant
- Partition (nœud P)
  - si (tous les éléments de P sont dans la même classe) alors retour;
  - pour chaque attribut A faire
    - évaluer la qualité du partitionnement sur A;
  - utiliser le meilleur partitionnement pour diviser P en P1, P2, …Pn
  - pour i = 1 à n faire Partition(Pi);

Procédure de Construction (2)

- Processus récursif
  - L'arbre commence à un nœud représentant toutes les données
  - Si les objets sont de la même classe, alors le nœud devient une feuille étiqueté par le nom de la classe.
  - Sinon, sélectionner les attributs qui séparent le mieux les objets en classes homogènes => Fonction de qualité
  - La récursion s'arrête quand:
    - Les objets sont assignés à une classe homogène
    - Il n'y a plus d'attributs pour diviser,
    - Il n'y a pas d'objet avec la valeur d'attribut

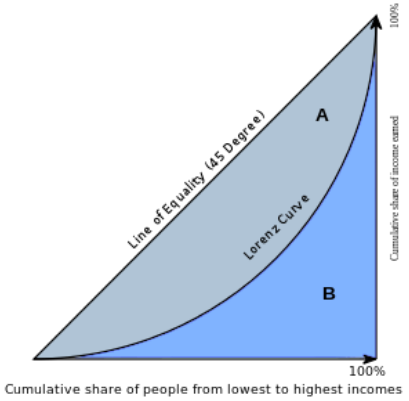


Choix de l'attribut de division

- Différentes mesures introduites
  - il s'agit d'ordonner le désordre
  - des indicateurs basés sur la théorie de l'information
- Choix des meilleurs attributs et valeurs
  - les meilleurs tests
- Possibilité de retour arrière
  - élaguer les arbres résultants (classes inutiles)
  - revoir certains partitionnements (zoom, réduire)

Mesure de qualité

- La mesure est appelé fonction de qualité
  - Goodness Function en anglais
- Varie selon l'algorithme :
  - Gain d'information (ID3/C4.5)
    - Suppose des attributs nominaux (discrets)
    - Peut-être étendu à des attributs continus
  - Gini Index
    - Suppose des attributs continus
    - Suppose plusieurs valeurs de division pour chaque attribut
    - Peut-être étendu pour des attributs nominaux



Mesure d'impureté (variable nominale)

- Mesure des mélanges de classes d'un nœud N
  - $i(N) = \sum i \sum j \{ pi * pj \}$ avec $i \neq j$
  - pi est la proportion d 'individus de la classe i dans N.
- La réduction d'impureté de chaque division du nœud N par la variable xj s'exprime par:
    - $\Delta N = i(N) - \sum_j pj * i(Nj)$
    - pj est la proportion d'individus du nœud dans le fils j
- Sur l 'ensemble des n variables, la division du nœud t est effectuée à l 'aide de la variable qui assure la réduction maximale de l'impureté ($\sum$ minimum)

Mesure d'entropie

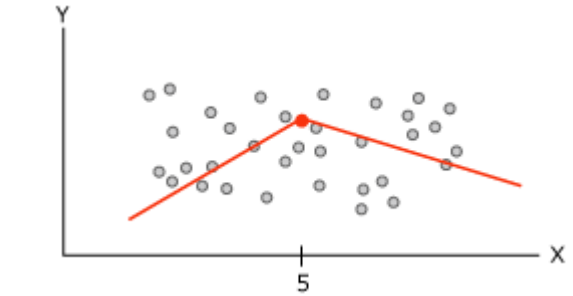
- Minimisation du désordre restant
  - pi = fréquence relative de la classe i dans le nœud N (% d ’éléments de la classe i dans N)
- Mesure d ’entropie d'un segment s
  - $E(N) = -\sum pi \, Log_2(pi)$
- Minimiser son évolution globale [Quinlan]
  - $\Delta N = E(N) - \sum_j Pj * E(Nj)$

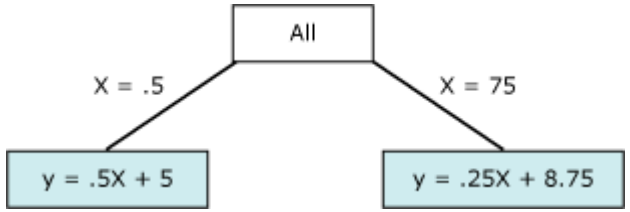
Problème des attributs continus

- Certains attributs sont continus
  - exemple : salaire
- découper en sous-ensembles ordonnés (e.g.,déciles)
  - division en segments [a0,a1[, [a1,a2[, …., [an-1,an]
- utiliser moyenne, médiane, … pour représenter
- minimiser la variance, une mesure de dispersion …
- investiguer différents cas et retenir le meilleur
  - exemple : 2, 4, 8, etc. par découpe d'intervalles en 2 successivement

Attributs continus: Régression

- Partitionnement par droite de régression
- Chaque nœud est représenté par une formule de régression
- Séparation des données = point de non linéarité
- 1 ou plusieurs régresseurs
- Exemple :
  - salaire = a + b*tranche_age





Procédure d'élagage

- Les arbres trop touffus sont inutiles
- Intérêt d'un élagage récursif à partir des feuilles
  - S'appuie sur un modèle de coût d'utilité
- Possibilité de l'appliquer sur l'ensemble des données ou sur un sous-ensemble réservé à la validation

Exemple d'élagage

- Exemple :
  - arbres vus comme encodage de tuples
  - partition utile si gain supérieur à un seuil
  - coût d'un partitionnement
    - CP bits pour coder les prédicats de patition
    - Entropie_Après bits pour coder chaque tuple
  - partitionnement à supprimer si :
    - Gain = n * Entropie_Après + CP - n * Entropie_Avant < seuil
- Ce test peut être appliquer lors de la création

Exemple : Méthode CART (WEKA : SimpleCART)

- Principes
  - si problème à 2 classes, cherche la bi-partition minimisant l'indice d'impureté de Gini
  - si problème à N classes, cherche celle maximisant le gain d'information donné par l'indice de Towing
- Critères d 'arrêt :
  - Seuil de gain informationnel
  - Seuil d 'effectif dans un nœud
  - Procédure d'élagage

Méthodes passant à l'échelle

- La plupart des algorithmes de base supposent que les données tiennent en mémoire
- La recherche en bases de données a proposer des méthodes permettant de traiter de grandes BD
- Principales méthodes:
  - SLIQ (EDBT'96 -- Mehta et al.'96)
  - SPRINT (VLDB96 -- J. Shafer et al.'96)
  - RainForest (VLDB98 -- J. Hekankho et al.'98)
  - PUBLIC (VLDB'98 -- R. Rastogi et al.'98)

Bilan

- De nombreux algorithmes de construction d'arbre de décision
- SPRINT passe à l'échelle et traite des attributs nominaux ou continus
- Autres algorithmes proposés
  - Encore plus rapides ?

4. Réseaux Bayésiens

- Classificateurs statistiques
- Basés sur les probabilités conditionnelles
- Prévision du futur à partir du passé
- Suppose l'indépendance des attributs

Fondements

- Dérivé du théorème de Bayes
  - permet de calculer une probabilité à postériori P(Ci/X) d'un événement Ci sachant que X s'est produit à partir d'une probabilité à priori P(Ci) de production de l'événement Ci
  - $P(Ci/X) = P(X/Ci)*P(Ci) / \sum P(X/Cj)*P(Cj)$
- Plus simplement si E est l'événement:
  - $P(E/X) = P(X/E)*P(E)/P(X)$

Bayésien Naïf (WEKA : naive Bayes)

- Chaque enregistrement est un tuple
  - X= (x1, x2, …xn) sur R(A1, A2, …An)
  - Il s'agit de classer X parmi m classes C1, …Cm
  - L'événement Ci est l'appartenance à la classe Ci
- Assignation de la classe la plus probable
  - Celle maximisant P(Ci/X) = P(X/Ci)*P(Ci)/P(X)
  - P(X) est supposé constant (équi-probabilité des tuples)
- On cherche la classe maximisant :
  - P(X/Ci)*P(Ci) pour i= 1 à m

On calcule la probabilité de chaque classe étant donné le tuple X

On retient la classe la plus probable

Calcul de P(X/Ci)

- P(Ci) est déduite de l'échantillon :
  - Comptage "training set" = Taille(Ci)/ Taille(Ech)
- P(X/Ci) est approchée comme suit :
  - Indépendance des attributs ➔
  - $P(X/Ci) = \Pi_k P(x_k/Ci)$
- $P(x_k/Ci)$ est estimé comme suit:
  - variable nominale = Taille(t=xk de Ci)/Taille(Ci)
  - distribution gaussienne si variable continue

P(xk/Ci) est la probabilité d'avoir une valeur donnée xk pour un attribut d'un tuple dans la classe Ci ; Calculée sur le training set

Exemple de problème

- Faut-il effectuer un contrôle fiscal ?
  - Échantillon de contrôlés

Salaire	Impôts	Etudiant	Contrôle
20	0	oui	négatif
30	0	non	positif
40	5	oui	positif
40	25	non	négatif
60	10	non	positif

  - Faut-il contrôler un nouvel arrivant ?

35	2	oui	???

Calcul de Probabilités

- Il s'agit de choisir Ci maximisant P(Ci/X) :
  - P(Positif/X) = P(X/Positif)P(Positif)/P(X)
  - P(Négatif/X) = P(X/Négatif)P(Négatif)/P(X)
  - P(X) est supposé constant
- Donc, choisir le plus grand de {P(X/Positif)P(Positif), P(X/Négatif)P(Négatif)}
  - $P(X/Positif) = \Pi_k P(Xk/Positif)$ =P(sal30..50/Positif)* P(impots<20%/Positif)*P(Etudiant/Positif) = 2/3*1*1/3=2/9; P(Positif) = 3/5 ➔ Produit = 0.13
  - $P(X/Négatif) = \Pi_k P(Xk/Négatif)$ =P(sal30..50/Négatif)* P(impots<20%/Négatif)*P(Etudiant/Négatif) = 1/2*1/2*1/2=1/8; P(Négatif) = 2/5 ➔ Produit = 0.05
- On effectuera donc un contrôle !

Intérêt

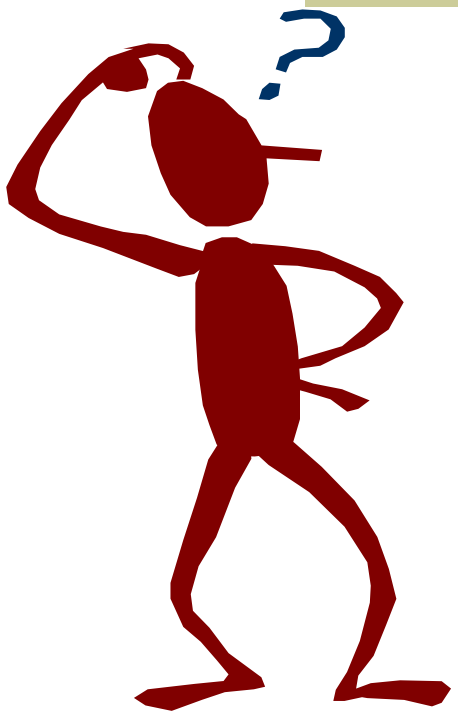
- Permet d'inférer les probabilités dans le réseau
  - méthode d ’inférence du futur à partir du passé
  - les événements Xi doivent être indépendants
  - méthode assez peu appliquée en Data Mining
- Problèmes
  - Comment choisir la structure du réseau ?
  - Comment limiter le temps de calcul ?

Bilan

- Apprentissage
  - si structure connue = calculs de proba.
  - si inconnue = difficile à inférer
- Baysien naïf
  - suppose l'indépendance des variables
- Réseaux baysiens
  - permettent certaines dépendances
  - nécessitent des tables d'apprentissage réduites

5. Bilan Classification

- De nombreuses techniques dérivées de l'IA et des statistiques
- Autres techniques
  - règles associatives, raisonnement par cas, ensembles flous, …
- Problème de passage à l'échelle
  - arbre de décisions, réseaux
- Tester plusieurs techniques pour résoudre un problème



- Y-a-t-il une technique dominante ?