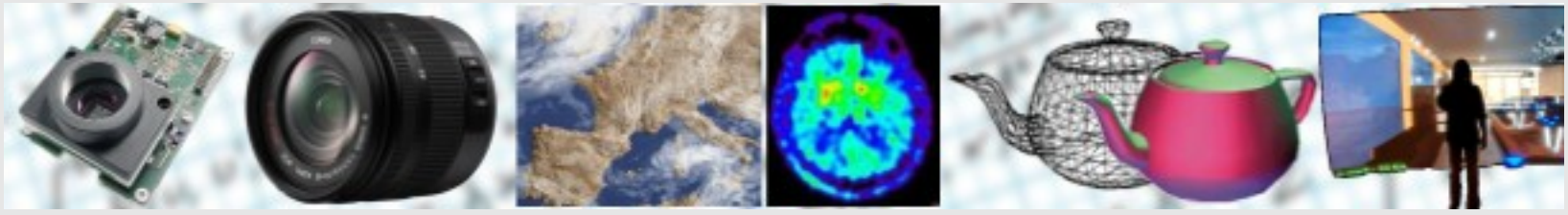


RdF – Reconnaissance des Formes
Semaine 9 : arbres de décision

Master ASE : http://master-ase.univ-lille1.fr/
Master Informatique : http://www.fil.univ-lille1.fr/
Spécialité IVI : http://master-ivi.univ-lille1.fr/

IEEA

Plan du cours

1 – Introduction

transition numérique vs. nominal

2 – Exemples introductifs

exemples d'arbres, cas variés

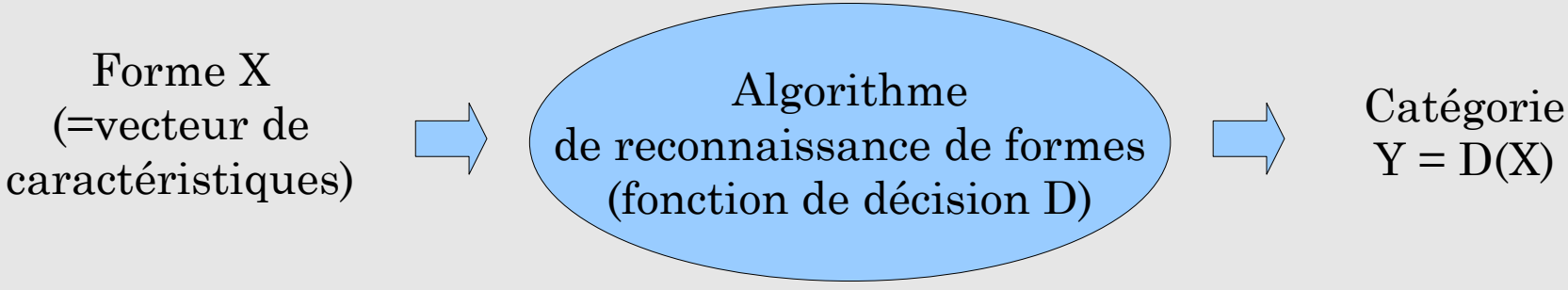
3 – Induction

principe de construction d'arbres de décision

Reconnaissance de Formes

But de la RdF

classer correctement des formes décrites par un ensemble de caractéristiques



Méthodes statistiques en reconnaissance des formes

cf. cours précédents

Méthodes syntaxiques : objet des 4 prochains cours

structuration en arbres
chaînes, règles, grammaires

Numérique vs. Nominal (non-numérique)

Avant

on a travaillé jusqu'ici dans un domaine numérique
valeurs réelles et vecteurs de caractéristiques

$$\vec{X}=(x_1,\ldots,x_n), x_i \in \mathbb{R}$$

notions utiles de similarité et d'ordre

$$dist(a,b)=d, d\in\mathbb{R}$$

$$r_1, r_2 \in \mathbb{R} : r_1 < r_2 \text{ ou } r_1 = r_2 \text{ ou } r_1 > r_2$$

exemple : température, vitesse, ratio, longueur, etc.

Maintenant

on s'intéresse à des listes d'attributs de type nominal

Numérique vs. Nominal (non-numérique)

Attribut nominal

exemples : couleur, énumération, etc.

$\{\text{color} = \text{red}, \text{texture} = \text{shiny}, \text{taste} = \text{sweet}, \text{size} = \text{small}\}$

Cas particuliers

attribut binaire

- propriété vérifiée ou non, défaillant, etc.
(test d'égalité uniquement)

attribut ordinal

- petit/moyen/grand, matin/midi/soir, etc.
(test d'égalité, notion d'odre)

Arbres de décision

Arbres de décision

méthode d'apprentissage supervisé

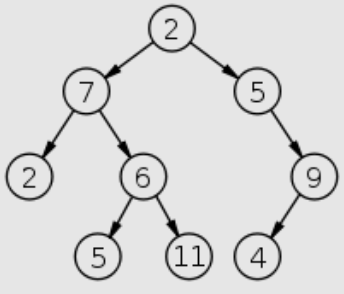
objectif : prédire les valeurs d'une variable à partir d'un ensemble de descripteurs (variables prédictives, variables discriminantes, variables d'entrées, ...)

Définition

un arbre (informatique) est une structure de données comportant des noeuds dont une seule racine

cas particulier de graphe orienté connexe

exemple d'arbre binaire



Arbres de décision

Principe de fonctionnement

on va de la racine à une feuille en effectuant les tests sur les noeuds

classe d'une feuille : classe majoritaire parmi les exemples d'apprentissage appartenant à cette feuille

Exemples introductifs (1/3)

Tiré de l'ouvrage de Quinlan (1993)

il s'agit de prédire le comportement de sportifs (jouer : variable à prédire) en fonction de données météo (ensoleillement, température, humidité, vent : variables prédictives)

Exemples introductifs (1/3)

Tiré de l'ouvrage de Quinlan (1993)

il s'agit de prédire le comportement de sportifs (jouer : variable à prédire) en fonction de données météo (ensoleillement, température, humidité, vent : variables prédictives)

Exemples introductifs (2/3)

Tiré du cours de Fabien Moutarde (CAOR, Mines Paris Tech

Exemples introductifs (3/3)

Tiré de l'ouvrage de Duda, Hart, Stork (2001)

on note que la question « Size? » apparaît plusieurs fois, et que les questions peuvent avoir des nombres de branches différents par ailleurs, plusieurs feuilles (en rose) peuvent avoir la même étiquette (e.g. « Apple »)

Avantages des arbres de décision

Entre autres

lisibilité

interprétabilité (vs. d'autres classifieurs comme RNA)

intégration directe de connaissances à priori d'experts humains

catégories traduisibles en disjonctions de conjonctions

Exemple

traduction de « Apple » :

$$\text{Apple} = (\text{green} \wedge \text{medium}) \vee (\text{red} \wedge \text{medium})$$

simplification en :

$$\text{Apple} = (\text{medium} \wedge \neg \text{yellow})$$

Faire « pousser » un arbre

Méthode pour l'induction

on démarre avec un ensemble d'exemples étiquetés, et un certain nombre de propriétes discriminantes

le processus divise (« split ») progressivement l'ensemble en sous-ensembles plus petits

- cas idéal : chaque sous-ensemble contient des exemples de même catégorie (on dit qu'ils sont purs) => on arrête là
- en général : mélange de catégories dans les sous-ensembles => on doit décider si on continue à faire grandir l'arbre ou non (en acceptant l'imperfection)

Construction récursive!

problème : étant donné un ensemble d'exemples *E*, construire un arbre de décision le plus petit possible (principe du rasoir d'Occam), consistant avec *E*

Faire « pousser » un arbre

Algorithme général

```
Constuire_arbre(X)
SI tous les éléments de X sont de même catégorie
    Créer une feuille associée à cette classe
SINON
    Selon un certain critère choisir le meilleur couple
(attribut;test) pour créer un noeud
    // Ce test sépare X en m parties Xk
    Pour chaque Xk:
        Construire_arbre(Xk)
```

Faire « pousser » un arbre

CART (Classifiction and Regression Trees)

cadre général de création d'arbres

Six questions centrales avec l'approche CART

1 - les propriétés doivent-elles être réduite à des **valeurs binaires**? ou bien les autorise-t-on à être multivaluées?

2 - **quelle propriété** devrait être testée à un noeud donné?

3 - quand un noeud doit-il être déclaré **feuille**?

4 - quand un arbre devient trop grand, comment le rendre plus petit et plus simple (comment **l'élaguer**)?

5 - quand un noeud est « impur », **quelle étiquette** lui assigner?

6 - comment traiter les **valeurs manquantes**?

1 – Valeurs binaires?

On peut toujours se ramener au cas binaire

pouvoir expressif universel

exemple : le noeud racine « Color? » peut être remplacé par deux noeuds « Color=green? » (oui/non) et « Color=yellow? » (oui/non)

==> dessin au tableau!

2 - Critères de choix d'attribut et de test

Question posée ici

étant donné un noeud *N*, quel test choisir?
heuristique évidente : choisir le test qui augment la « pureté » (homogénéité) des sous-noeuds crées :

$$\Delta i(N) = i(N) - \sum_j P(N_j) i(N_j)$$

avec $i(N)$ un indice d'hétérogénéité (impureté) du noeud N

et $P(N_j)$ la proportion des éléments de N dirigés vers N_j par ce test

parfois normalisé par l'entropie, pour éviter un biais favorisant un trop grand nombre de sous-noeuds :

$$\frac{\Delta i(N)}{-\sum_k P(N_k) \log_2 P(N_j)}$$

2 - Critères de choix d'attribut et de test

Entropie (ID3, C4.5)

$$i(X) = -\sum_j P(w_j) \log_2 P(w_j), \text{ où } P(w_j) = \frac{N_j}{N}$$

Indice de Gini (CART)

$$i(X) = 1 - \sum_j P^2(w_j)$$

Indice d'erreur de classification

$$i(X) = 1 - max_j P(w_j)$$

Le cas des attributs continus

nombre fini d'exemples d'apprentissage, idem pour le nombre de valeurs prises par les attributs

- tri par valeur croissante

- médianes de valeurs successives pour le choix du seuil

==> exemple : longeur 10 ou 20 ; « longueur>15? »

3 - Critère d'arrêt

Intuitivement

quand tous les éléments sont dans la même classe
quand tous les éléments ont les mêmes valeurs d'attributs
quand un seuil est atteint (diminution de l'hétérogénéité des noeuds, nombre d'éléments dans les noeuds)

Cas extrêmes

- si on arrête trop « tôt » : noeuds impurs, erreur apprentissage, mauvaise performance
- si on arrête trop « tard » : noeud purs, voire un seul exemple par noeud, mauvaise généralisation (overfit)

Elagage a posteriori

suppression des branches peu représentatives, nuisant à la généralisation

4 - Elagage

Alternative aux critères d'arrêt

arrêter « tardivement » la construction de l'arbre (exprès) puis fusionner itérativement des paires de feuille soeurs

Principe

- toutes les soeurs sont candidates

- toutes celles dont l'élimination produit une hausse acceptable (petite) de l'hétérogénéité sont supprimées, et le noeud père est déclaré feuille (à son tour, il devient une soeur candidate!)

Opération inverse du *split*

5 - Quelle étiquette?

Etape la plus simple

si tous les éléments dans le noeud sont de même catégorie (pur): l'étiquette est évidement cette catégorie

sinon, l'étiquette du noeud est celle des exemples majoritaires dans le noeud

Rappel

une très petite hétérogénéité n'est pas nécessairement désirable (sur-apprentissage des exemples)

6 - Comment traiter les valeurs manquantes?

Quand se pose ce problème?

lors de la construction de l'arbre
lors de la classification d'un nouvel exemple

Construction

- on peut ignorer l'attribut concerné (beaucoup de v.m.)
- ou bien, mesurer le gain d'homogénéité pour cet attribut uniquement sur les valeurs présentes (le reste ne change pas!)

Classification

solution 1 : modification de la procédure de construction (question 2), pour prévoir à chaque noeud, des **tests alternatifs**
solution 2 : inventer une **valeur probable** (calcul de corrélation)

Algorithmes courants

CART

principe général qu'on vient de voir
les autres sont des variantes (choix attribut, arrêt, élagage, etc.)

ID3

attributs nominaux (valeurs réelles doivent être discrétisées)
choix d'attribut basé sur le gain d'information (entropie)
arrêt quand tous les noeuds sont purs
pas d'élagage dans la version standard

C4.5

raffinement de ID3
autorise valeurs réelles (comme CART)
heuristique d'élagage basée sur signifiance statistique des splits

Pour approfondir

Duda, Hart, Stork, « Pattern Classification », 2ème édition, Wiley-Interscience, 2001.
http://rii.ricoh.com/~stork/DHS.html

Quinlan, « C4.5: Programs for Machine Learning », Morgan Kaufman, 1993.
http://download-book.net/quinlan-c4.5-pdf-doc.html