Initiation à la Fouille de Données et à l'Apprentissage

Quatrième séance
Apprentissage d'arbres de décision (2/2)

M2 I2A
2011-2012

Valentin Emiya

Plan

Algorithmes d'apprentissage d'arbres de décision

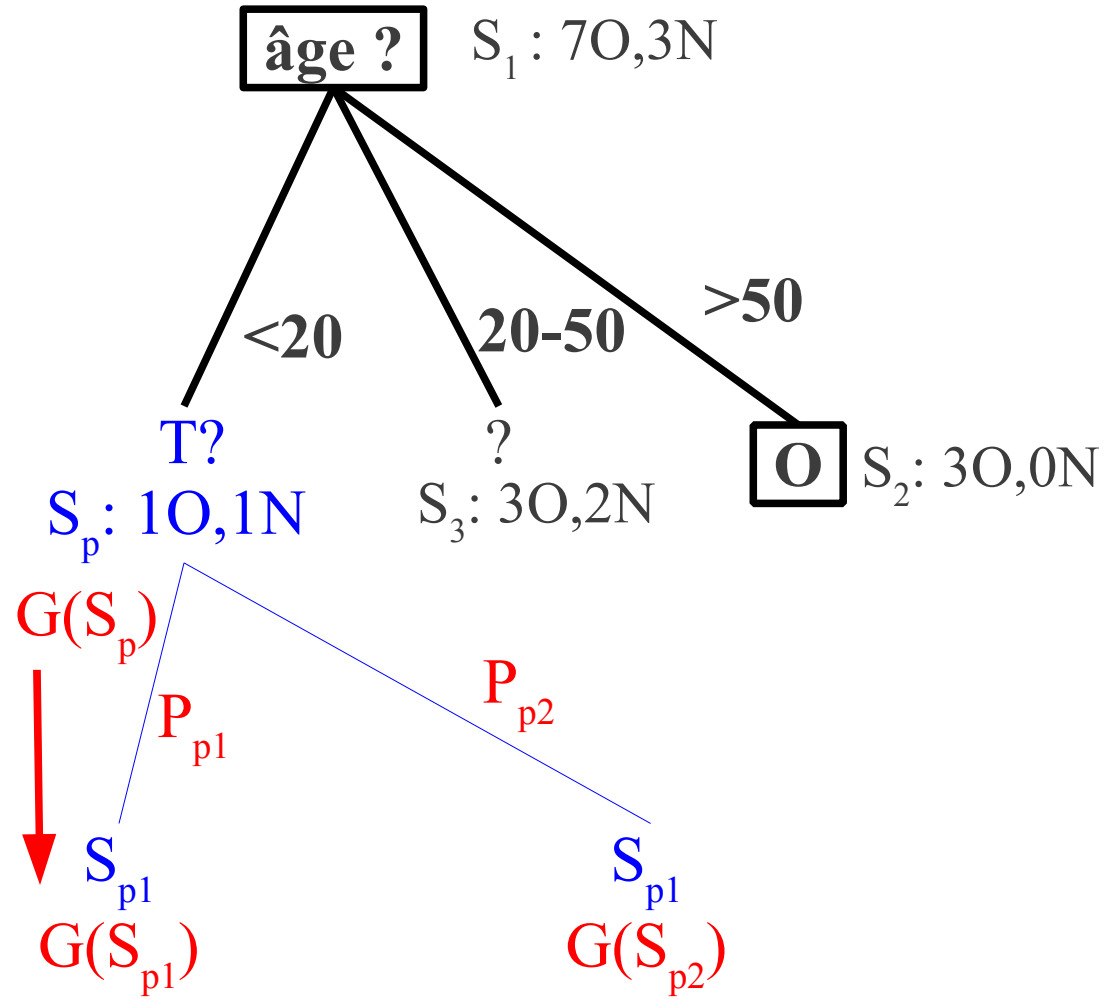
- Étape 1 : construction d'un petit arbre (rappel)
- Étape 2 : élagage / prévenir le surapprentissage

Construction de l'arbre

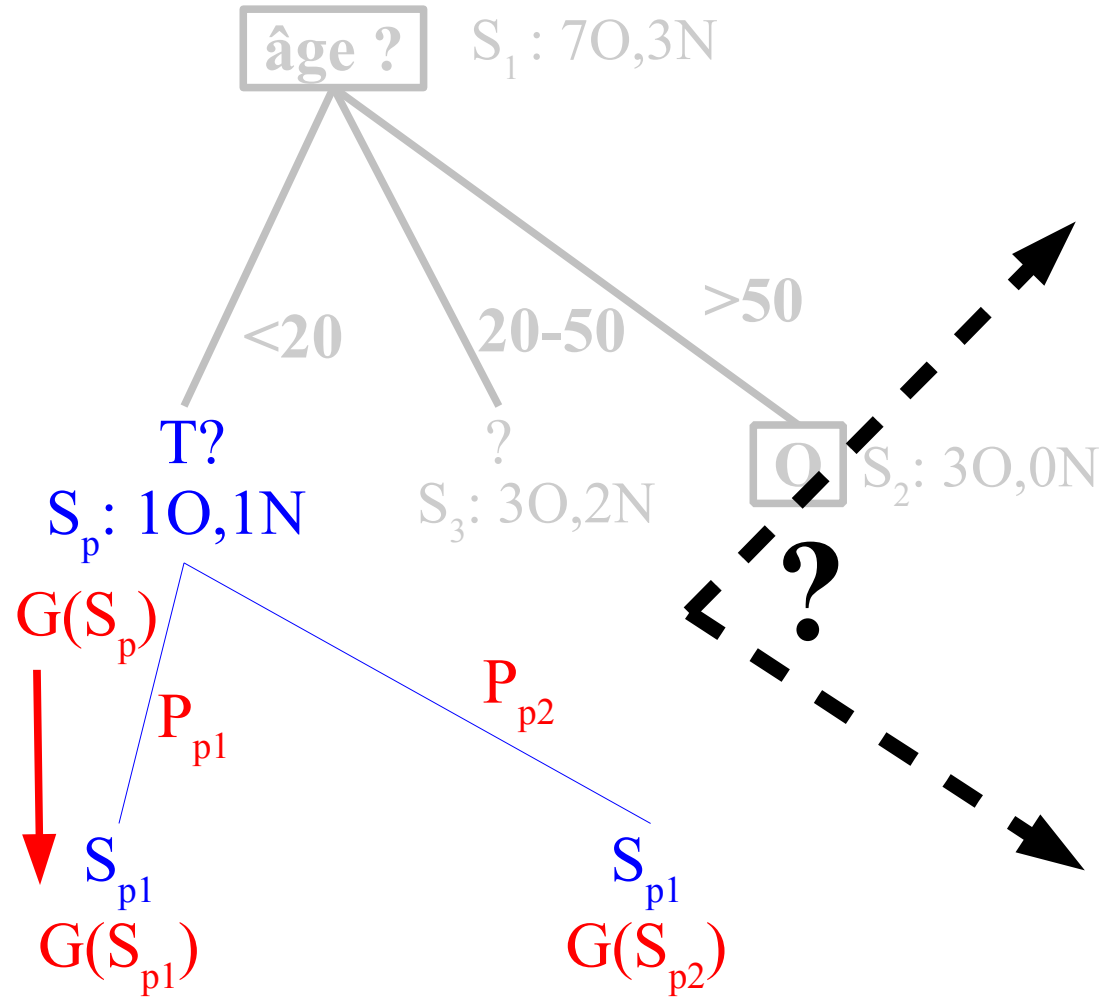
Principe :

- Objectif : classer efficacement = disperser les classes.
- Méthode top-down : depuis la racine d'un arbre vide.
- À chaque itération :
  - construction locale d'un test/nœud qui disperse au max.
  - ou construction d'une feuille
- Critère de sélection d'un test : maximiser le gain de dispersion
  = dispersion à l'entrée – dispersion à la sortie du test/noeud
- Indices de dispersion : Gini, entropie

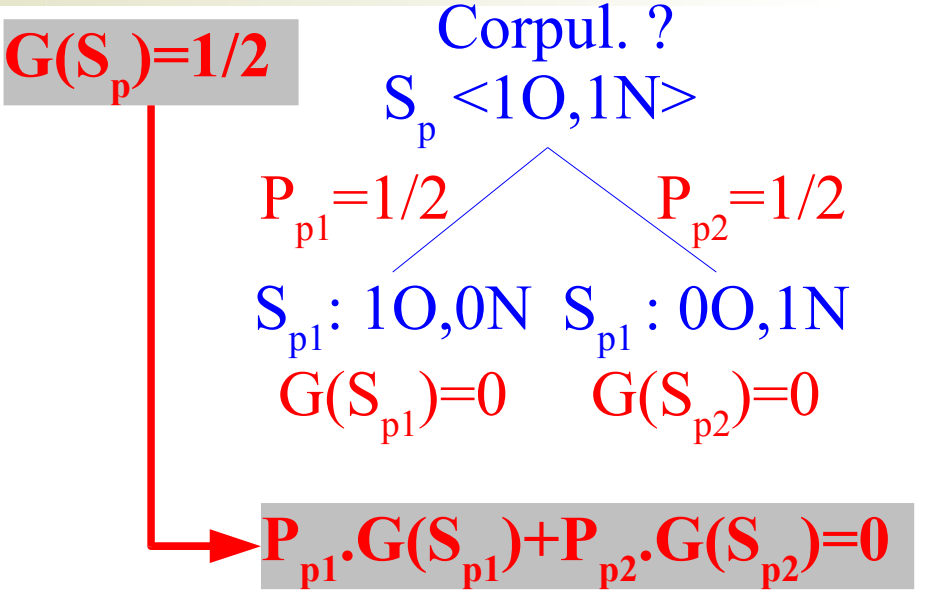
Maximiser le gain de dispersion

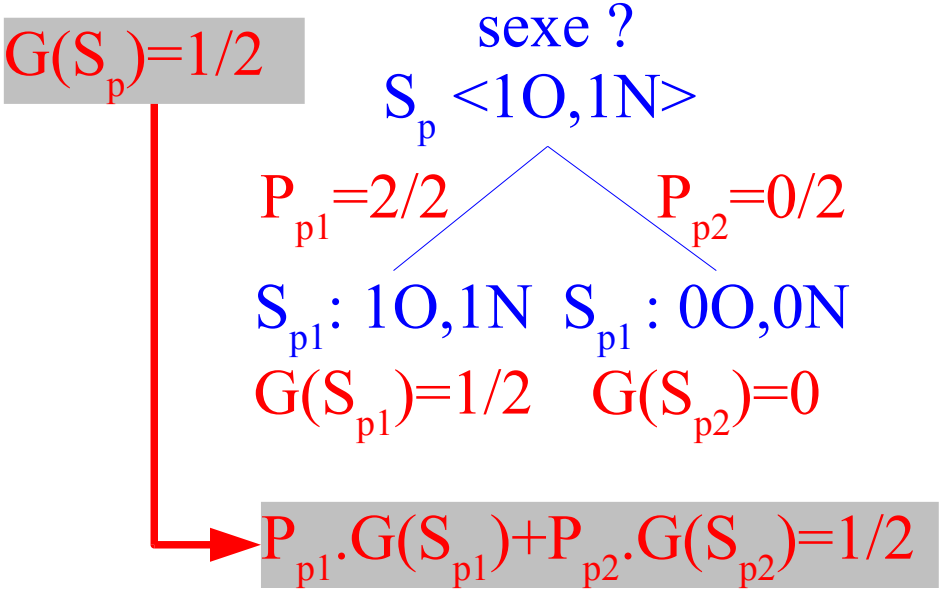


Maximiser le gain de dispersion



?





Exercices 2-3 p.23-24

n	$\log_2(n)$
1	0
2	1
3	1.58
4	2
5	2.32
6	2.58
7	2.81
8	3
9	3.17
10	3.32

Exercice 2 : corrigé en cours
Exercice 3 : à faire chez soi

Exercice

(source : Cornuéjols & Miclet)

Construire l'arbre de décision en utilisant le critère de Gini ou l'entropie.

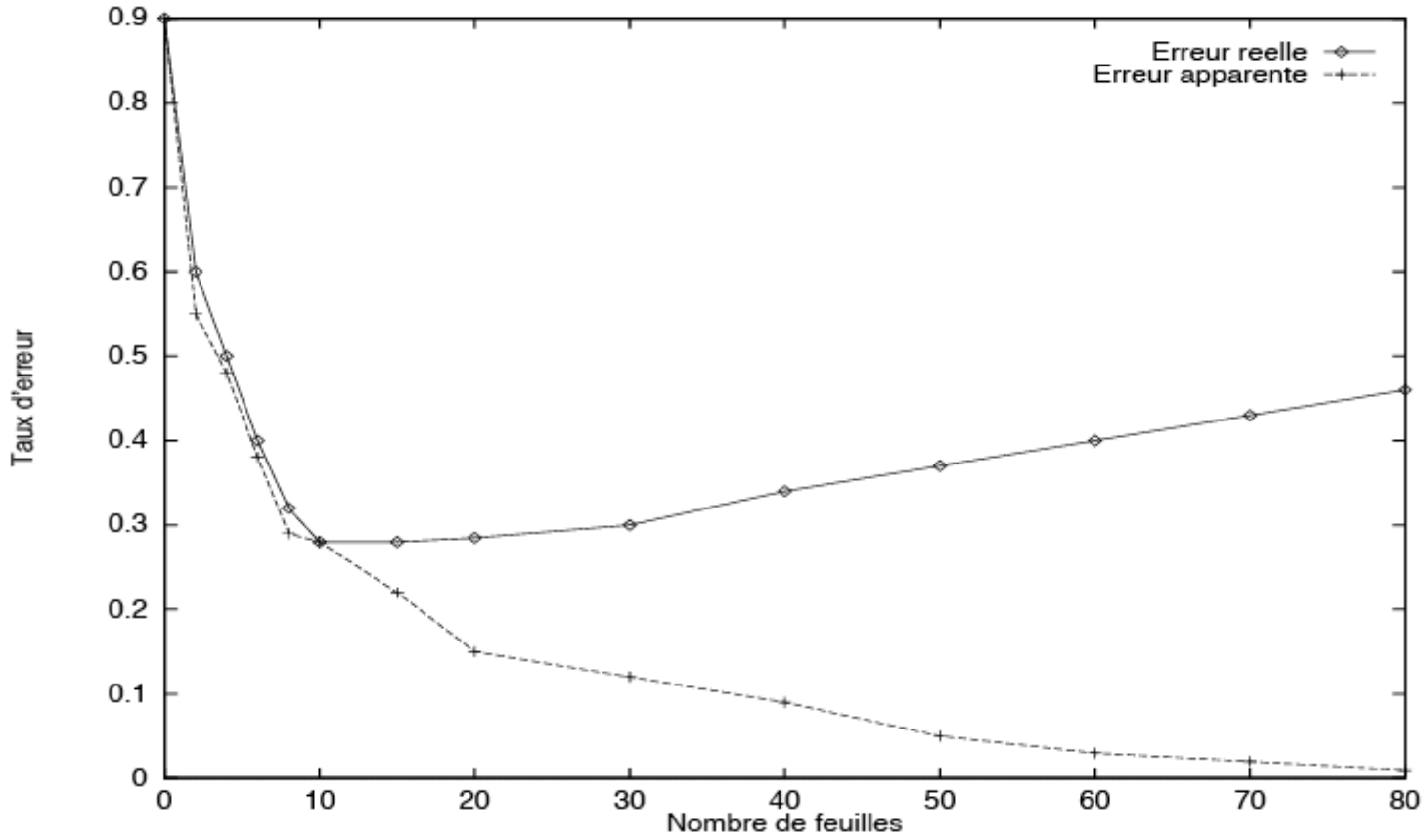
n	Devoirs finis ?	Maman de bonne humeur ?	Beau temps ?	Goûter pris ?	Peut aller jouer ?
1	V	F	V	F	O
2	F	V	F	V	O
3	V	V	V	F	O
4	V	F	V	V	O
5	F	V	V	V	N
6	F	V	F	F	N
7	V	F	F	V	N
8	V	V	F	F	N

Bilan de l'étape 1 (construction)

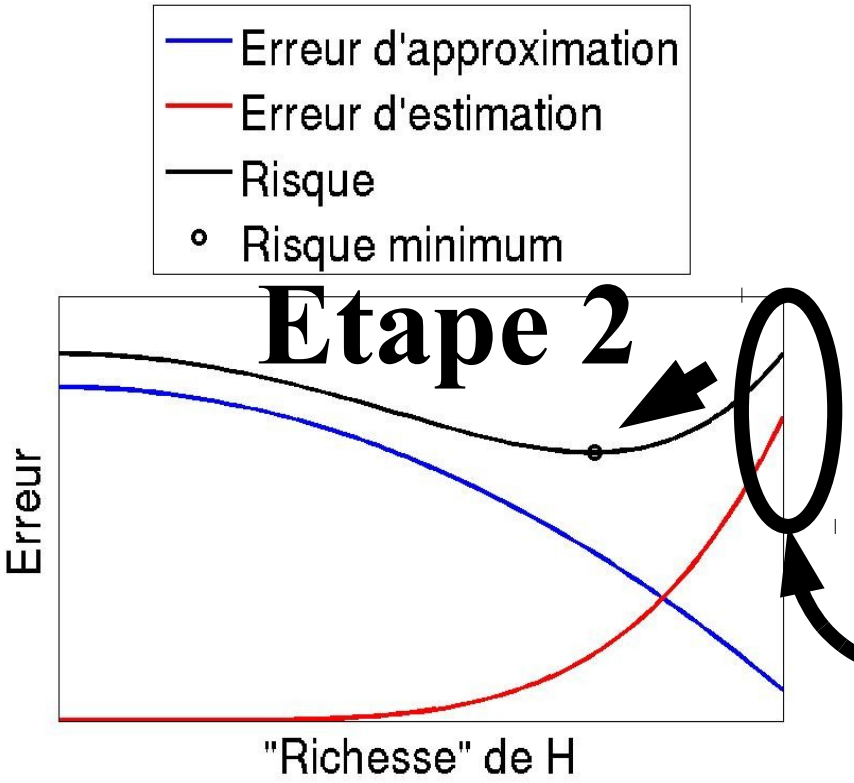
- L'algorithme a construit
  - un arbre adapté aux données
    = minimisation (approximative) du risque empirique
  - de « petite » taille parmi tous ceux possibles
- **Mais :**
  - risque empirique atteint : très faible, voire nul
  - en effet, nous nous sommes contentés de suivre l'intuition de minimisation du risque empirique
- Quid du risque réel ?
  L'arbre a-t-il de bonnes propriétés de **généralisation** ?

Erreur empirique vs. réelle

- Un arbre peut avoir une erreur apparente nulle mais une erreur réelle importante, c'est-à-dire être bien adapté à l'échantillon mais avoir un pouvoir de prédiction faible.
- Problème de surapprentissage



Quid du compromis biais-variance ?



risque empirique (quasi-)nul

= H très riche

(H : « tous les arbres possibles »)

= grande erreur d'estimation

= sur-apprentissage

Étape 2

- Objectif : éviter le surapprentissage
- Principe : réduire la richesse de H

  = limiter le nombre de nœuds/feuilles
- Moyen : algorithme + ensemble de validation

Eviter le sur-apprentissage

2 stratégies possibles

- Early-stopping : on utilise un ensemble de validation pour arrêter la construction de l'arbre quand l'estimation de l'erreur ne diminue plus.
- Élagage : on construit l'arbre en entier, puis on l'élague

  Avantage de l'élagage : on n'élague pas forcément dans l'ordre inverse de construction

Élagage d'arbre de décision (CART)

- Supposons qu'on a construit un arbre T_0.
  $\alpha = \Delta R_{emp}(S) / |T_p|-1$ où $\Delta R_{emp}(S)$ est le nombre d'erreurs supplémentaires que commet l'arbre de décision sur S lorsqu'on l'élague à la position p et où $|T_p|-1$ mesure le nombre de feuilles supprimées.
- T_{i+1} est obtenu en élaguant T_i en un nœud en lequel α est minimal. Soit $T_0, \ldots, T_i, \ldots, T_t$ la suite obtenue, T_t étant réduit à une feuille. On sélectionne l'arbre T_i dont le nombre d'erreurs calculé sur un ensemble de validation S_{valid} est minimal.

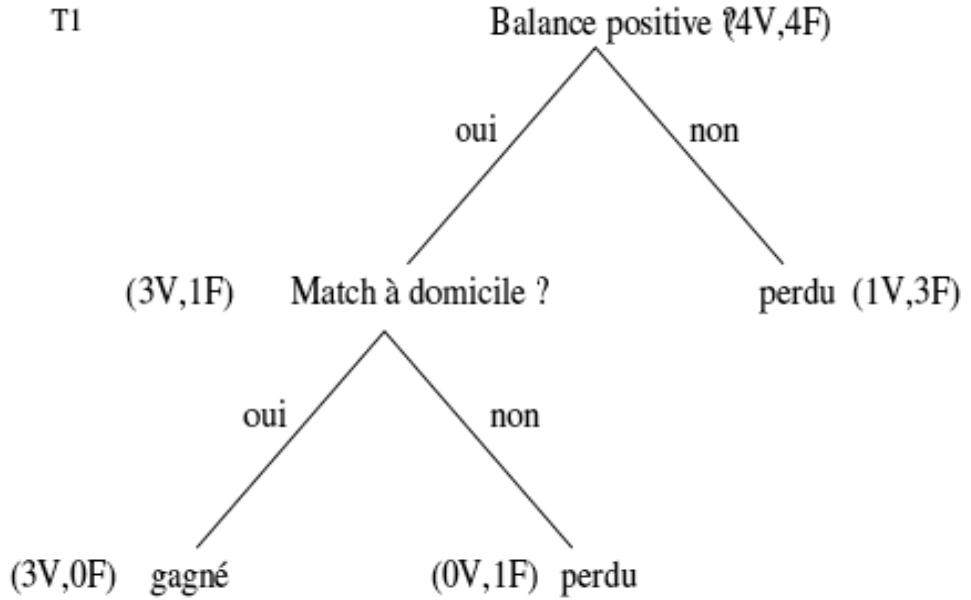
Retours à l'exemple

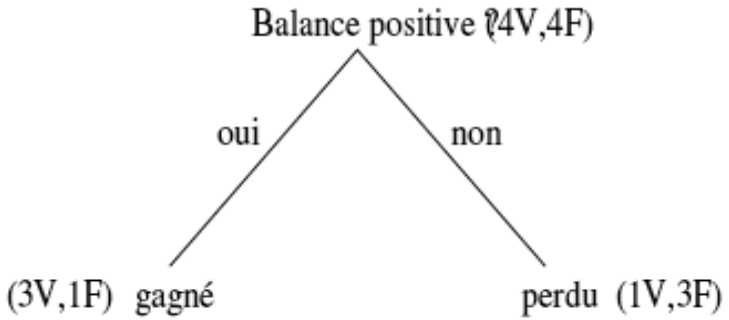
- On dispose de l'ensemble de validation suivant :

Match à domicile ?	Balance positive ?	Mauvaises conditions climatiques ?	Match précédent gagné ?	Match gagné
V	V	V	F	V
F	V	V	F	V
F	F	F	V	F
V	F	V	F	F
V	F	V	F	V

L'arbre T0 est l'arbre construit précédemment.
T1 est l'arbre obtenu en élaguant à partir de la position 2
T2 est obtenu en élaguant à partir de la position 1.
T3 est réduit à une feuille, portant la classe gagné.
L'algorithme d'élagage retournera alors l'arbre T2.

Résultat de CART sur l'exemple





T3 Gagné (4V,4F)

Exercice

Apprentissage sur n<9

Validation sur n>8

Construire l'arbre de décision en utilisant le critère de Gini et élaguer

n	T1	T2	T3	Classe
1	**0**	**V**	**N**	**A**
2	**1**	**V**	**I**	**A**
3	**0**	**F**	**O**	**B**
4	**1**	**V**	**N**	**A**
5	**1**	**V**	**O**	**A**
6	**1**	**V**	**I**	**B**
7	**0**	**F**	**O**	**B**
8	**0**	**V**	**I**	**A**
9	0	F	N	B
10	1	F	N	A
11	1	F	O	A
12	1	F	I	A
13	0	V	O	B

Exercice

Early stopping pour les 2 exercices précédents

(à faire chez soi)

Complément sur les attributs

- Les propriétés vues sur les attributs binaires s'étendent aux attributs n-aires
- Attributs discrets : il est possible (si on veut des arbres binaire par exemple) de regrouper a priori des valeurs des attributs.
- Attribut continus : processus de discrétisation a(souvent à l'aide d'inégalités).
- Attributs à valeurs manquantes :
  - En classement : prendre la branche majoritaire
  - En apprentissage : donner un valeur suivant la distribution (locale ou globale sur l'échantillon)

Pour être complet

- On peut facilement introduire une matrice de coût de prédictions erronées.
- Des attributs n-aires peuvent prendre un grand nombre de valeurs : il est possible de les pénaliser pour retarder leur apparition dans l'arbre.
- Instabilité : Un des inconvénients principaux des méthodes d'apprentissage par arbres de décision est leur *instabilité*. Sur des données réelles, un attribut est choisi plutôt qu'un autre se joue à peu de chose. Or le choix d'un attribut-test, surtout s'il est près de la racine, influence grandement le reste de la construction. Ces algorithmes ont une *variance importante*.