

Apprentissage logique inductif

Les arbres de décision

Exercice: construire un arbre de décision

Exemple: cartes "singulières"

- Jeu de cartes, chacune désignée par $[r, c]$, son rang et sa couleur et certaines cartes sont "singulières"

- Connaissance de fond KB
- $$((r=1) \vee \dots \vee (r=10)) \Leftrightarrow \text{NUM}(r)$$
- $$((r=\text{Valet}) \vee (r=\text{Dame}) \vee (r=\text{Roi})) \Leftrightarrow \text{FACE}(r)$$
- $$((c=\text{PI}) \vee (c=\text{TR})) \Leftrightarrow \text{NOIR}(c)$$
- $$((c=\text{CA}) \vee (c=\text{CO})) \Leftrightarrow \text{ROUGE}(c)$$

- Ensemble d'apprentissage Δ :
- $$\text{SINGULIER}([4, \text{TR}]) \wedge \text{SINGULIER}([7, \text{TR}]) \wedge \text{SINGULIER}([2, \text{PI}]) \wedge \neg \text{SINGULIER}([5, \text{CO}]) \wedge \neg \text{SINGULIER}([\text{Valet}, \text{PI}])$$

PI: Pique, TR: Trèfle, CA: Carreau, CO: Coeur

Enjeux

- KB une base de connaissances "de fond" (background knowledge)
- Δ ensemble d'apprentissage (connaissance observée) qui n'est pas logiquement déductible de KB

- Inférence inductive:
trouver une hypothèse inductive h telle que

KB et h impliquent Δ

$h = \Delta$ est une solution triviale,
mais sans aucun intérêt

Exemple: cartes "singulières"

- Jeu de cartes, chacune désignée par $[r, c]$, son rang et sa couleur et certaines cartes sont "singulières"

- Connaissance de fond KB
- $$((r=1) \vee \dots \vee (r=10)) \Leftrightarrow \text{NUM}(r)$$
- $$((r=\text{Valet}) \vee (r=\text{Dame}) \vee (r=\text{Roi})) \Leftrightarrow \text{FACE}(r)$$
- $$((c=\text{PI}) \vee (c=\text{TR})) \Leftrightarrow \text{NOIR}(c)$$
- $$((c=\text{CA}) \vee (c=\text{CO})) \Leftrightarrow \text{ROUGE}(c)$$

Il y a plusieurs hypothèses inductives possibles

- Ensemble d'apprentissage Δ :
- $$\text{SINGULIER}([4, \text{TR}]) \wedge \text{SINGULIER}([7, \text{TR}]) \wedge \text{SINGULIER}([2, \text{PI}]) \wedge \neg \text{SINGULIER}([5, \text{CO}]) \wedge \neg \text{SINGULIER}([\text{Valet}, \text{PI}])$$

- Hypothèse inductive possible:
 $h \equiv (\text{NUM}(r) \wedge \text{NOIR}(c) \Leftrightarrow \text{SINGULIER}([r, c]))$

Apprendre un prédictat

Soient:

- E un ensemble d'objets,
- CONCEPT(x) un prédictat à apprendre, où x est un objet de E, et qui prend la valeur Vrai ou Faux (e.g. SINGULIER).

Exemple:

CONCEPT décrit la précondition d'une action, par ex., Dépiler(C,A)

- E est l'ensemble des états
- CONCEPT(x) \Leftrightarrow

PinceVideex, BLOC(C) ex, BLOC(A) ex,
LIBRE(C) ex, SUR(C,A) ex

Apprendre CONCEPT constitue une étape vers l'apprentissage de l'action

Apprendre un prédictat

Soient:

- E un ensemble d'objets,
- CONCEPT(x) un prédictat à apprendre, où x est un objet de E, et qui prend la valeur Vrai ou Faux (e.g. SINGULIER).

- A(x), B(x), ... des prédictats observables (e.g. NUM, ROUGE),

- ensemble d'apprentissage: valeurs de CONCEPT pour certaines combinaisons de valeurs des prédictats observables,

Un ensemble d'apprentissage

Ex. #	A	B	C	D	E	CONCEPT
1	True	True	False	True	False	False
2	True	False	False	False	False	False
3	False	False	True	True	True	True
4	True	True	True	False	True	False
5	False	True	True	False	True	True
6	True	True	False	True	True	False
7	False	False	True	False	True	False
8	True	False	True	False	True	True
9	False	False	False	True	True	False
10	True	True	True	True	False	True

Noter que l'ensemble d'apprentissage ne dit rien sur la pertinence d'un prédictat observable A, ..., E.

Apprendre un prédictat

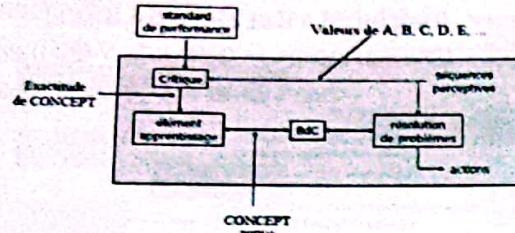
Soient:

- E un ensemble d'objets,
- CONCEPT(x) un prédictat à apprendre, où x est un objet de E, et qui prend la valeur Vrai ou Faux (e.g. SINGULIER),
- A(x), B(x), ... des prédictats observables (e.g. NUM, ROUGE),
- un ensemble d'apprentissage formé des valeurs de CONCEPT pour certaines combinaisons de valeurs des prédictats observables,
- Trouver une représentation de CONCEPT sous la forme:

$$\text{CONCEPT}(x) \Leftrightarrow S(A, B, \dots)$$

où S(A,B,...) est une expression construite à partir des prédictats observables, ex:
 $\text{CONCEPT}(x) \Leftrightarrow A(x) \wedge (\neg B(x) \vee C(x))$

Apprendre un prédictat



Apprendre le concept "Arche"

Ces objets sont des arches:
(exemples positifs)



Ceux-ci n'en sont pas:
(exemples négatifs)



$\text{ARCHE}(x) \Leftrightarrow \text{Pour-Partie}(x,b1) \wedge \text{Pour-Partie}(x,b2) \wedge \text{Pour-Partie}(x,b3) \wedge \text{Est-Un}(b1,\text{BRIQUE}) \wedge \text{Est-Un}(b2,\text{BRIQUE}) \wedge \neg\text{Contact}(b1,b2) \wedge (\text{Est-Un}(b3,\text{BRIQUE}) \vee \text{Est-Un}(b3,\text{COIN})) \wedge \text{Supporté-Par}(b3,b1) \wedge \text{Supporté-Par}(b3,b2)$

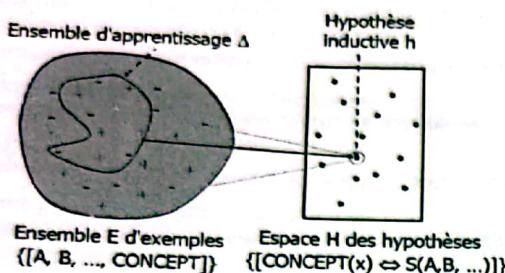
Ensemble d'exemples

- Un exemple est formé des valeurs de CONCEPT et des prédictats observables pour un objet donné x,
- un exemple est positif si CONCEPT est Vrai, sinon il est négatif,
- l'ensemble E formé de tous les exemples est appelé l'ensemble d'exemples,
- l'ensemble d'apprentissage est un sous-ensemble de E.

Espace des hypothèses

- Une hypothèse est n'importe quelle expression h de la forme:
 $\text{CONCEPT}(x) \Leftrightarrow S(A,B, \dots)$
où $S(A,B,\dots)$ est une expression construite à partir des prédictats observables,
- l'ensemble de toutes les hypothèses possibles est appelé l'espace des hypothèses et il est dénoté par H ,
- une hypothèse h coïncide avec un exemple si elle donne la bonne valeur de CONCEPT.

Schéma d'un apprentissage inductif



Dimension de l'espace H

- Si on a n prédictats observables,
- on aura alors 2^n entrées dans la table de vérité.
- en l'absence de toute restriction (biais), il y a ...
 2^n hypothèses parmi lesquelles choisir.
- si $n = 6 \rightarrow 2 \times 10^{19}$ hypothèses !!!

Hypothèses inductives multiples

- Jeu de cartes, chacune désignée par $[r, c]$, son rang et sa couleur et certaines cartes sont "singulières"
- Connaissance de fond KB

$$\begin{aligned} & \{([r=1] \vee \dots \vee [r=10]) \Leftrightarrow \text{NUM}(r) \\ & ([r=\text{Valet}] \vee [r=\text{Dame}] \vee [r=\text{Roi}]) \Leftrightarrow \text{FACE}(r) \\ & ([c=\text{PT}] \vee [c=\text{TR}]) \Leftrightarrow \text{NOIR}(c) \\ & ([c=\text{CA}] \vee [c=\text{CO}]) \Leftrightarrow \text{ROUGE}(c) \end{aligned}$$
- Ensemble d'apprentissage Δ :

$$\begin{aligned} & \text{SINGULIER}([4, \text{TR}]) \wedge \text{SINGULIER}([7, \text{TR}]) \wedge \text{SINGULIER}([2, \text{PI}]) \wedge \\ & \neg \text{SINGULIER}([5, \text{CO}]) \wedge \neg \text{SINGULIER}([\text{Valet}, \text{PI}]) \end{aligned}$$

$$h_1 \equiv \text{NUM}(x) \wedge \text{NOIR}(x) \Leftrightarrow \text{SINGULIER}(x)$$

$$h_2 \equiv \text{NOIR}([r, c]) \wedge \neg(r=\text{Valet}) \Leftrightarrow \text{SINGULIER}([r, c])$$

$$h_3 \equiv ([r, c]=[4, \text{TR}]) \vee ([r, c]=[7, \text{TR}]) \vee [r, c]=[2, \text{PI}] \Leftrightarrow \text{SINGULIER}([r, c])$$

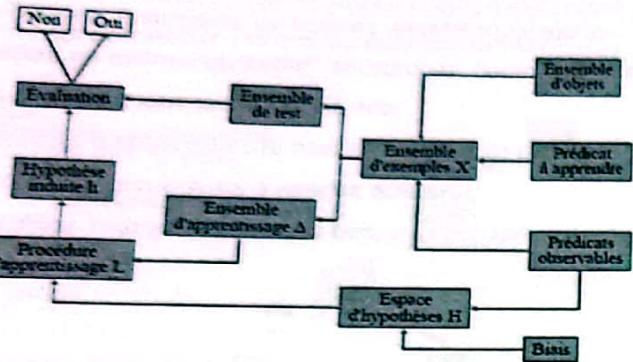
$h_3 \equiv \neg([r, c]=[5, \text{CO}]) \vee \neg([r, c]=[Valet, \text{PI}]) \Leftrightarrow \text{SINGULIER}([r, c])$
sont compatibles avec tous les exemples de l'ensemble d'apprentissage

Il faut un système de préférence –biais– pour comparer les hypothèses entre elles

Biais "Keep-It-Simple" (KIS)

- Motivation
 - si une hypothèse est trop complexe, il peut ne pas être valable de l'apprendre,
 - il y a beaucoup moins d'hypothèses simples que d'hypothèses complexes, par conséquent l'espace d'hypothèses correspondant est plus petit.
 - Exemples:
 - utiliser moins de prédictats observables que suggéré par l'ensemble d'apprentissage,
 - contraindre le prédictat appris à ne contenir que des prédictats observables de "haut niveau" et/ou avoir une syntaxe simple (e.g. une conjonction de littéraux).
- Si le biais limite les expressions à n'être que des conjonctions de k prédictats pris parmi les n prédictats observables ($k << n$), alors la dimension de H est $O(n^k)$.

En mettant tout ensemble



Méthodes d'apprentissage de prédicats

- Représentation par arbre de décision
- Algorithme d'apprentissage par arbre de décision
 - Algorithmes de Quinlan (ID3, C4.5, C5.0)
 - Algorithme de Breiman (CART: Classification And Regression Tree)
- Entropie et gain en information, indice Gini

Les arbres de décision

- Un arbre de décision est un arbre au sens informatique
- Les noeuds sont repérés par des positions $\in \{1, \dots, p\}^*$, où p est l'arité maximale des noeuds.
- Les noeuds internes sont les noeuds de décision.
- Un noeud de décision est étiqueté par un test qui peut être appliqué à chaque description d'un individu d'une population.
- Chaque test examine la valeur d'un unique attribut.
- Dans les arbres de décision binaires on omet les labels des arcs.
- Les feuilles sont étiquetées par une classe.

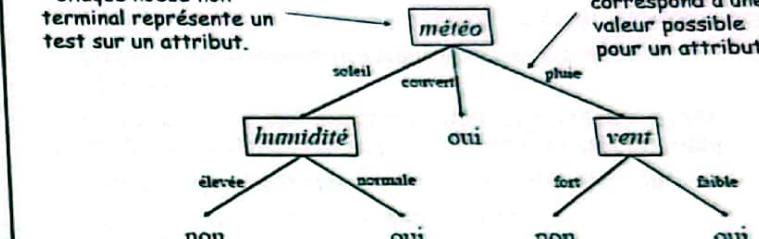
Les arbres de décision

- À chaque arbre, on associe naturellement une procédure de classification.
- À chaque description complète est associée une seule feuille de l'arbre.
- La procédure de classification représentée par un arbre correspond à des règles de décision.
- Exemple :

Si Température < 37,5 ET gorge irritée ALORS malade
Si Température < 37,5 ET gorge non irritée ALORS bien portant
Si Température ≥ 37,5 ALORS malade

Un arbre de décision

- Chaque noeud non-terminal représente un test sur un attribut.
- Chaque branche correspond à une valeur possible pour un attribut.



- Chaque noeud terminal correspond à une classification possible.

météo, humidité et vent sont les attributs décrivant les exemples.

Quand utiliser un arbre de décision ?

Lorsque:

- les exemples sont décrits par des paires <attributs, valeurs>,
- la fonction à apprendre est à valeurs discrètes,
- les exemples d'apprentissage sont incomplets ou bruités,
- pour des applications réelles telles que:
 - analyse de risques (finances, assurances, géologiques, etc..),
 - diagnostics (médicaux, de pannes, surveillance, etc..),
 - systèmes d'aide à la décision,
 - etc..

Applications des arbres de décision

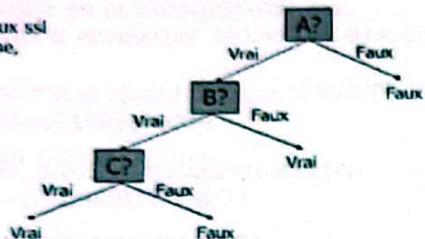
- Octroi de crédits (American Express),
- Diagnostic médical,
- Classification et gestion automatiques d'avions dans les systèmes de contrôle du trafic aérien,
- Contrôle de processus complexes
 - BP utilise un système expert construit sur un arbre de décision pour contrôler le séparation gaz-pétrole sur les plates-formes offshore (Michie 1986),
- Pilotage automatique: avions et hélicoptères (Sammut 1992)
 - systèmes de pilotage (par simulateur) aux performances meilleures que celles de l'expert qui a fourni les données d'apprentissage,
- Génération automatique de règles dans les systèmes experts.

Un prédicat vu comme arbre de décision

Le prédicat $\text{CONCEPT}(x) \Leftrightarrow A(x) \wedge (\neg B(x) \vee C(x))$ peut être représenté par l'arbre de décision suivant:

Exemple:
un champignon est vénéneux ssi il est jaune et petit, ou jaune, gros et tacheté

- x est un champignon
- $\text{CONCEPT} = \text{Vénéneux}$
- $A = \text{Jaune}$
- $B = \text{Gros}$
- $C = \text{Tacheté}$

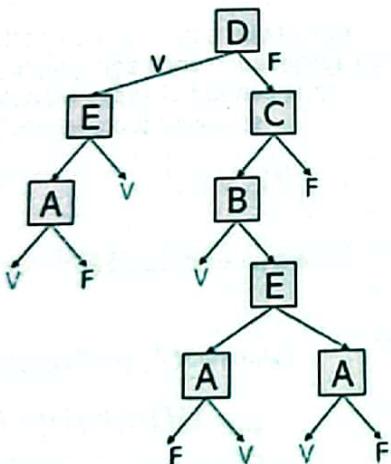


Ensemble d'apprentissage

Ex. n°	A	B	C	D	E	CONCEPT
1	Faux	Faux	Vrai	Faux	Vrai	Faux
2	Faux	Vrai	Faux	Faux	Faux	Faux
3	Faux	Vrai	Vrai	Faux	Vrai	Vrai
4	Faux	Faux	Vrai	Faux	Faux	Faux
5	Faux	Faux	Faux	Vrai	Faux	Vrai
6	Vrai	Faux	Vrai	Faux	Faux	Vrai
7	Vrai	Faux	Faux	Vrai	Faux	Vrai
8	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai
9	Vrai	Vrai	Vrai	Faux	Vrai	Vrai
10	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai
11	Vrai	Vrai	Faux	Faux	Vrai	Faux
12	Vrai	Vrai	Faux	Faux	Vrai	Vrai
13	Vrai	Faux	Vrai	Vrai	Vrai	Vrai

Un arbre de décision possible

Ex. n°	A	B	C	D	E	CONCEPT
1	Faux	Faux	Vrai	Faux	Vrai	Faux
2	Faux	Vrai	Faux	Faux	Faux	Faux
3	Faux	Vrai	Vrai	Faux	Faux	Faux
4	Faux	Faux	Vrai	Faux	Faux	Faux
5	Faux	Faux	Faux	Vrai	Faux	Faux
6	Vrai	Faux	Vrai	Faux	Vrai	Vrai
7	Vrai	Faux	Faux	Vrai	Vrai	Vrai
8	Vrai	Vrai	Vrai	Faux	Vrai	Vrai
9	Vrai	Vrai	Vrai	Faux	Faux	Vrai
10	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai
11	Vrai	Vrai	Faux	Faux	Vrai	Faux
12	Vrai	Vrai	Faux	Faux	Vrai	Vrai
13	Vrai	Faux	Vrai	Vrai	Vrai	Vrai

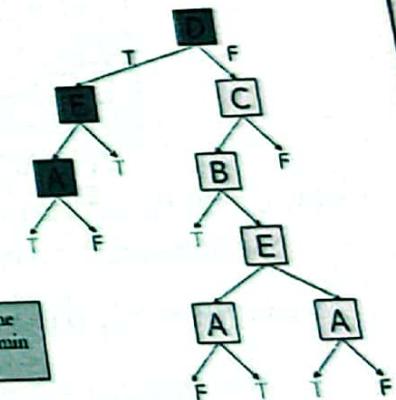


$$\text{CONCEPT} \Leftrightarrow (D \wedge (\neg E \vee A)) \vee (C \wedge (B \vee ((E \wedge \neg A) \vee A)))$$

Un arbre de décision possible

Ex. n°	A	B	C	D	E	CONCEPT
1	Faux	Vrai	Faux	Faux	Vrai	Faux
2	Faux	Vrai	Faux	Faux	Faux	Faux
3	Faux	Vrai	Vrai	Faux	Faux	Faux
4	Faux	Falses	Vrai	Faux	Vrai	Faux
5	Faux	Falses	Falses	Vrai	Vrai	Vrai
6	Vrai	Falses	Vrai	Falses	Falses	Vrai
7	Vrai	Falses	Falses	Vrai	Vrai	Vrai
8	Vrai	Vrais	Vrai	Falses	Vrai	Vrai
9	Vrai	Vrais	Vrai	Falses	Falses	Vrai
10	Vrai	Vrais	Vrai	Vrai	Vrai	Vrai
11	Vrai	Vrai	Vrai	Vrai	Falses	Vrai
12	Vrai	Vrai	Vrai	Falses	Vrai	Vrai
13	Vrai	Falses	Vrai	Vrai	Vrai	Vrai

Pour les expressions booléennes, une ligne de la table de vérité correspond à un chemin dans l'arbre de la racine à une feuille.



Algorithmes d'apprentissage d'arbres de décision

Plusieurs algorithmes : CART [Breiman84], C4.5[Quinlan94].

- Algorithmes en deux étapes :

- ▶ Construction d'un petit arbre de décision compatible
- ▶ Elagage de l'arbre

- Première étape :

- ▶ Idée principale : Diviser récursivement et le plus efficacement possible l'échantillon d'apprentissage par des tests définis à l'aide des attributs jusqu'à obtenir des sous-échantillons ne contenant (presque) que des exemples appartenant à une même classe.

- ▶ Méthodes de construction Top-Down, gloutonnes et récursives.

Pour commencer

Induction de haut en bas d'un arbre de décision

La distribution de l'ensemble d'apprentissage est:

Vrai: 6, 7, 8, 9, 10, 13
Faux: 1, 2, 3, 4, 5, 11, 12

Ex. #	A	B	C	D	E	CONCEPT
1	False	False	True	False	True	Faux
2	False	True	False	False	False	Faux
3	False	True	True	True	True	Vrai
4	False	False	True	False	False	Faux
5	False	False	False	True	True	Faux
6	True	False	True	False	False	Vrai
7	True	False	False	True	False	Vrai
8	True	False	True	False	True	Vrai
9	True	True	True	False	True	Vrai
10	True	True	True	True	True	Vrai
11	True	True	False	False	False	Faux
12	True	True	False	False	True	Faux
13	True	False	True	True	True	Vrai

Pour commencer

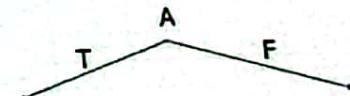
La distribution de l'ensemble d'apprentissage est:

Vrai: 6, 7, 8, 9, 10, 13
Faux: 1, 2, 3, 4, 5, 11, 12

Sans tester aucun prédictat observable, on pourrait dire que CONCEPT est Faux (règle de la majorité) avec une probabilité estimée de l'erreur $P(E) = 6/13$

Si l'on ne veut inclure qu'un seul prédictat observable dans l'arbre de décision, quel prédictat devrait-on tester pour minimiser la probabilité d'erreur (càd le # d'exemples de l'ensemble d'apprentissage mal classés)? → algorithme « greedy »

Supposons que ce soit A



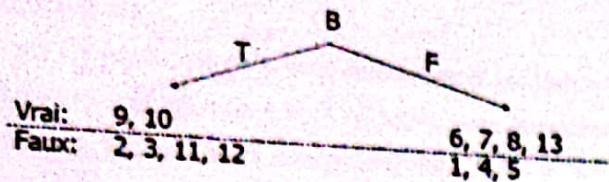
Vrai: 6, 7, 8, 9, 10, 13
Faux: 11, 12 1, 2, 3, 4, 5

Si on teste uniquement A, on rapportera que CONCEPT est Vrai si A est Vrai (règle de la majorité) et Faux autrement

La probabilité estimée de l'erreur est:
 $P(E) = (8/13)x(2/8) + (5/13)x0 = 2/13$

→ il y a 2 exemples de l'ensemble d'apprentissage mal classés

Supposons que ce soit B

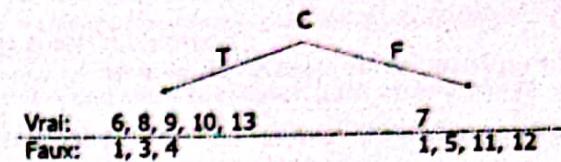


Si on teste uniquement B, on rapportera que CONCEPT est Vrai si B est Vrai et Faux autrement

La probabilité estimée de l'erreur est:

$$P(E) = (6/13) \times (4/6) + (7/13) \times (4/7) = 8/13$$

Supposons que ce soit C



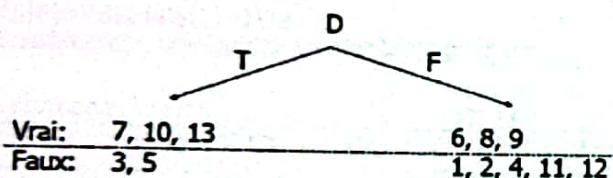
Si on teste uniquement C, on rapportera que CONCEPT est Vrai si C est Vrai et Faux autrement

La probabilité estimée de l'erreur est:

$$P(E) = (8/13) \times (3/8) + (5/13) \times (1/5) = 4/13$$

→ il y a 4 exemples de l'ensemble d'apprentissage mal classés

Supposons que ce soit D



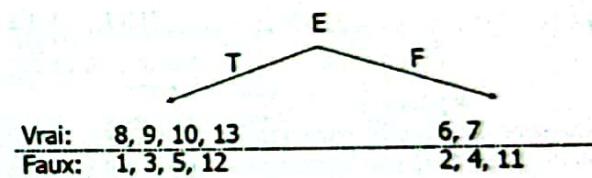
Si on teste uniquement D, on rapportera que CONCEPT est Vrai si D est Vrai et Faux autrement

La probabilité estimée de l'erreur est:

$$P(E) = (5/13) \times (2/5) + (8/13) \times (3/8) = 5/13$$

→ il y a 5 exemples de l'ensemble d'apprentissage mal classés

Supposons que ce soit E



Si on teste uniquement E, on rapportera que CONCEPT est Faux indépendamment de l'issue

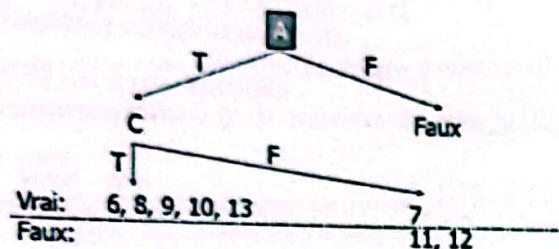
La probabilité estimée de l'erreur est inchangée:

$$P(E) = (8/13) \times (4/8) + (5/13) \times (2/5) = 6/13$$

→ il y a 6 exemples de l'ensemble d'apprentissage mal classés

Le meilleur prédicat à tester est donc A

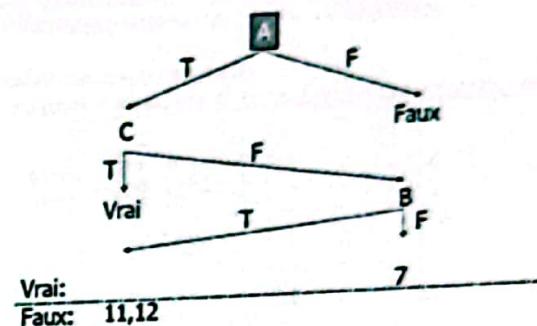
Choix du deuxième prédictat



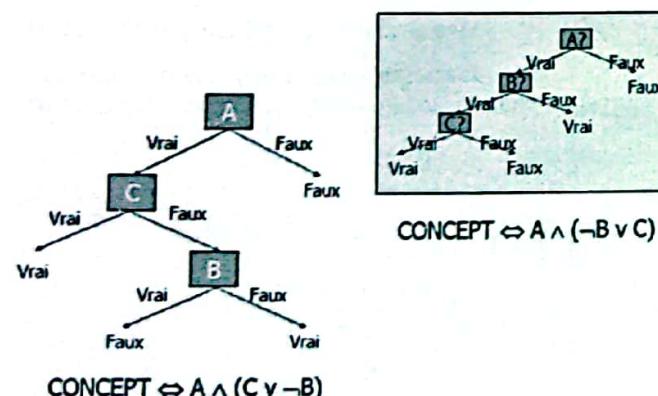
La règle de la majorité donne une probabilité d'erreur
 $P(E|A) = 1/8$ et $P(E)=1/13$

→ il y a 1 exemple de l'ensemble d'apprentissage mal classé

Choix du troisième prédictat



Arbre final



Algorithme DTL (Decision Tree Learning)

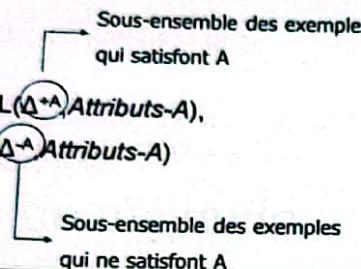
- entrées:
 - Δ : un ensemble d'exemples étiquetés positifs ou négatifs.
 - Attributs: les attributs décrivant les exemples.
- sortie:
 - un arbre de décision qui classe correctement les exemples d'apprentissage

Algorithme DTL

$DTL(\Delta, Attributs)$

1. Si tous les exemples de Δ sont positifs alors retourner Vrai
2. Si tous les exemples de Δ sont négatifs alors retourner Faux
3. Si Attributs est vide alors retourner Echec
4. $A \leftarrow$ attribut le plus discriminant de Attributs
5. Retourner l'arbre dont:
 - la racine est A ,
 - la branche gauche est $DTL(\Delta^A, Attributs-A)$,
 - la branche droite est $DTL(\Delta^{-A}, Attributs-A)$

En présence de bruit sur l'ensemble d'apprentissage, on peut retourner une règle de majorité, au lieu d'échec



Propriétés de DTL

- L'espace des hypothèses est complet: toutes les fonctions-objectifs à valeurs discrètes et pour les attributs donnés y sont contenues.
- DTL produit une seule hypothèse (càd. un seul arbre).
- C'est un algorithme de type "greedy", il peut éventuellement se trouver "bloqué" dans des minima locaux:
 - mais le choix de l'attribut le plus discriminant par calcul du gain en information s'il ne produit pas l'arbre de décision le plus compact donne de bons résultats en pratique.
- Le choix des attributs les plus discriminants se faisant selon leurs valeurs d'entropie, DTL est donc résistant au bruit.
- Le biais inductif de DTL est difficile à formaliser: on observe une préférence systématique pour l'arbre le plus court.

Difficultés

- Données manquantes
 - exemples avec des attributs manquants.
- Attributs trop spécifiques
 - un attribut tel que le nom d'une personne maximise le gain en information, mais est inefficace en terme de généralisation.
- Attributs à valeurs continues
 - la numérisation (discrétilisation) est une solution possible; faut-il numériser avant ou pendant la construction de l'arbre, que faire si l'attribut n'est pas retenu lors de la construction?
- Données bruitées
 - erreurs de saisie, capteurs défectueux, phénomènes stochastiques, etc.

Construction des arbres de décision

- Étant donné un échantillon S et des classes $\{1, \dots, c\}$, on veut construire un arbre t
- À chaque position p de t correspond un sous-ensemble de S qui contient les éléments de S qui satisfont les tests de la racine jusqu'à p .
- On définit pour chaque p :
 - $N(p)$: le cardinal de l'ensemble des exemples associé à p
 - $N(k/p)$: le cardinal de l'ensemble des exemples associé à p de classe k
 - $P(k/p) = N(k/p)/N(p)$: la proportion d'éléments de classe k à

Algorithmes d'apprentissage d'arbres de décision

On a besoin de trois opérateurs permettant de :

- Décider si un noeud est terminal
- Si un noeud n'est pas terminal, lui associer un test
- Si un noeud est terminal, lui affecter une classe

Algorithme générique :

arbre \leftarrow arbre vide ; noeud_courant \leftarrow racine

Répéter

Décider si le noeud courant est terminal

Si le noeud est terminal alors lui affecter une classe

Sinon sélectionner un test et créer autant de noeuds fils qu'il y a de réponses au test

Passer au noeud suivant (si il existe)

Jusqu'à obtenir un arbre de décision

8

Mesurer le degré de mélange des exemples

- Un test est intéressant s'il permet une bonne discrimination
- Une fonction qui mesure le degré de mélange des exemples doit
 - prendre son maximum lorsque les exemples sont équirépartis (ici par exemple (4,4)).
 - prendre son minimum lorsque les exemples sont dans une même classe ((0,8) ou (8,0)).
- On veut une fonction qui minimise le degré de mélange

Les trois opérateurs (en général)

- Un noeud est terminal lorsque :
 - (presque) tous les exemples correspondant à ce noeud sont dans la même classe, ou
 - il n'y a plus d'attribut non utilisé dans la branche correspondante.
- On attribue à un noeud terminal la classe majoritaire (en cas de conflit, on peut choisir la classe majoritaire dans l'échantillon, ou en choisir une au hasard)
- On sélectionne le test qui fait le plus progresser la classification des données d'apprentissage. Comment mesurer cette progression ? CART utilise l'*indice de Gini* et C4.5 utilise la notion d'*entropie*

9

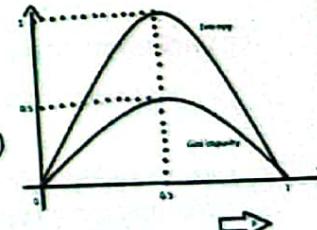
Fonctions mesurant le degré de mélange

- Fonctions minimisant le degré de mélange

$$\text{Entropie}(p) = -\sum_{k=1}^c P(k/p) \times \log(P(k/p))$$

$$\text{Gini}(p) = 1 - \sum_{k=1}^c P(k/p)^2$$

$$= 2 \sum_{k < k'} P(k/p)P(k'/p)$$



Choisir un test

- Soit f la fonction choisie (Gini ou Entropie)
- On définit le gain pour un test T et une position p
 - $Gain(p, T) = f(S_p) - \sum_{j=1}^n (P_j) \cdot f(S_{p_j})$
 - où n est l'arité du test T , S_p est l'échantillon associé à p et P_j est la proportion des éléments de S_p qui satisfont la $j^{\text{ème}}$ branche de T
- Sélectionner l'attribut dont le gain est maximum correspond à une stratégie gloutonne: rechercher le test qui fait progresser la classification

Conditions d'arrêt de développement d'un arbre-hyperparamètres

- La profondeur de l'arbre atteint une limite prédéfinie (on ne peut plus utiliser d'autres variables pour séparer davantage)
- Le nombre de noeud atteint une valeur maximale prédéfinie
- L'effectif (nombre d'échantillons) de chaque feuille atteint un seuil prédéfini
- ...

Quantité d'information et entropie (rappel)

• quantité d'information

$$I = \log_2 \frac{1}{p} = -\log_2 p$$

I = quantité d'information apportée par l'événement informatif e

p = probabilité de l'événement informatif e

• entropie

$$H = \sum_i p_i \log \frac{1}{p_i} = -\sum_i p_i \log p_i$$

H = entropie de la "source" des événements informatifs e_i

p_i = probabilité de l'événement informatif e_i

Choix de l'attribut le plus discriminant

On utilise la notion d'entropie (cf. théorie de l'information) pour déterminer quel est l'attribut le plus discriminant.

- Entropie (S) = $-p \log p - n \log n$
 - S = ensemble d'apprentissage,
 - p = proportion d'exemples positifs dans S ,
 - n = proportion d'exemples négatifs dans S ($n = 1-p$)
- Entropie de l'ensemble d'apprentissage
- Entropie (S) = nombre de bits nécessaires pour coder la classification (+ ou -) des éléments de S choisis aléatoirement avec un code optimal (càd. le plus court).
- La théorie de l'information dit que le code de longueur optimale pour un message de probabilité p nécessite $-\log_2 p$ bits.

Choisir un test

- Soit f la fonction choisie (Gini ou Entropie)
- On définit le gain pour un test T et une position p
 - $\text{Gain}(p, T) = f(S_p) - \sum_{j=1}^n (P_j) \cdot f(S_{p_j})$
 - où n est l'arité du test T , S_p est l'échantillon associé à p et P_j est la proportion des éléments de S_p qui satisfont la $j^{\text{ème}}$ branche de T
- Sélectionner l'attribut dont le gain est maximum correspond à une stratégie gloutonne: rechercher le test qui fait progresser la classification

Conditions d'arrêt de développement d'un arbre-hyperparamètres

- La profondeur de l'arbre atteint une limite prédéfinie (on ne peut plus utiliser d'autres variables pour séparer davantage)
- Le nombre de noeud atteint une valeur maximale prédéfinie
- L'effectif (nombre d'échantillons) de chaque feuille atteint un seuil prédéfini
- ...

Quantité d'information et entropie (rappel)

• quantité d'information

$$I = \log_2 \frac{1}{p} = -\log_2 p$$

I = quantité d'information apportée par l'événement informatif e

p = probabilité de l'événement informatif e

• entropie

$$H = \sum_i p_i \log \frac{1}{p_i} = -\sum_i p_i \log p_i$$

H = entropie de la "source" des événements informatifs e_i

p_i = probabilité de l'événement informatif e_i

Choix de l'attribut le plus discriminant

On utilise la notion d'entropie (cf. théorie de l'information) pour déterminer quel est l'attribut le plus discriminant.

- Entropie (S) = $-p \log p - n \log n$
 - S = ensemble d'apprentissage,
 - p = proportion d'exemples positifs dans S ,
 - n = proportion d'exemples négatifs dans S ($n = 1-p$)
- Entropie de l'ensemble d'apprentissage
 - Entropie (S) = nombre de bits nécessaires pour coder la classification (+ ou -) des éléments de S choisis aléatoirement avec un code optimal (càd. le plus court).
 - La théorie de l'information dit que le code de longueur optimale pour un message de probabilité p nécessite $-\log_2 p$ bits.

Choix de l'attribut le plus discriminant

- donc l'entropie de S est donnée par:

$$p(-\log_2 p) + n(-\log_2 n)$$

- l'entropie des 14 exemples (9 +, 5 -) de l'ensemble d'apprentissage "joue-t-on au tennis?" est de 0.94

jour	météo	humidité	température	vent	humidité +
1	-	-	-	-	-
2	-	-	-	-	-
3	-	-	-	-	-
4	-	-	-	-	-
5	-	-	-	-	-
6	-	-	-	-	-
7	-	-	-	-	-
8	-	-	-	-	-
9	-	-	-	-	-
10	-	-	-	-	-
11	-	-	-	-	-
12	-	-	-	-	-
13	-	-	-	-	-
14	-	-	-	-	-

Chaque jour (exemple) est décrit par des attributs:

- météo, température, humidité, vent

le concept (la fonction) à apprendre est:

$$f : \text{jour} \rightarrow \{\text{oui, non}\}$$

Entropie pour des problèmes à classes multiples

- Si la fonction-objectif prend c valeurs ($c > 2$), on peut calculer l'entropie de l'ensemble d'apprentissage comme:

$$\text{Entropie}(S) = \sum_{i=1}^c -p_i \log p_i$$

où p_i est la proportion d'éléments de S appartenant à la classe i

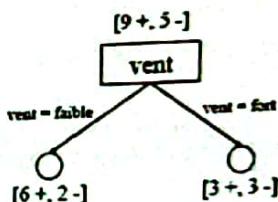
Gain en information

- $\text{Gain}(S, A)$ est la réduction d'entropie attendue suite à la partition de l'ensemble d'apprentissage par rapport au test sur l'attribut A .

$$\text{Gain}(S, A) = \text{Entropie}(S) - \sum_{v \in \text{valeurs}(A)} \frac{|S_v|}{|S|} \text{Entropie}(S_v)$$

- $\text{Gain}(S, A)$ représente le nombre de bits économisés dans le codage de la valeur de la fonction-objectif pour un membre quelconque de S en connaissant sa valeur pour l'attribut A .

Gain en information - exemple

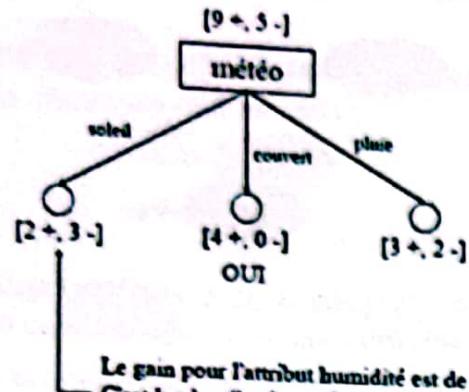


$$\begin{aligned} \text{Gain}(S, \text{vent}) &= 0.94 - (8/14) \text{Entropie}(S_{\text{faible}}) - (6/14) \text{Entropie}(S_{\text{fort}}) \\ &= 0.94 - (8/14) \times 0.811 - (6/14) \times 1.00 \\ &= 0.048 \end{aligned}$$

de la même manière:

$$\text{Gain}(S, \text{humidité}) = 0.151 \quad \text{et} \quad \text{Gain}(S, \text{météo}) = 0.246$$

Construction de l'arbre de décision



Le gain pour l'attribut humidité est de 0.97.
C'est le plus élevé parmi tous les attributs restants,
donc une partition selon "humidité" se fera ici.

Indice de Gini

Temps couvert
4+, 0-

$$I = 1 - \left[\left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 \right] = 0$$

Temps ensoleillé
4+, 6-

$$I = 1 - \left[\left(\frac{4}{10}\right)^2 - \left(\frac{6}{10}\right)^2 \right] = 0,48$$

Plus l'index de Gini est bas, plus le nœud de test est pur

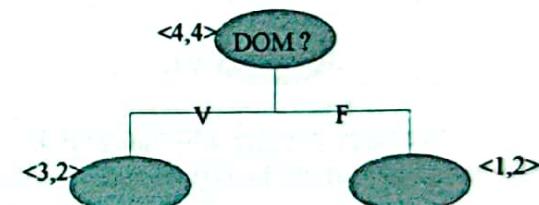
Exemple

- Pronostic de matchs

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

Exemple

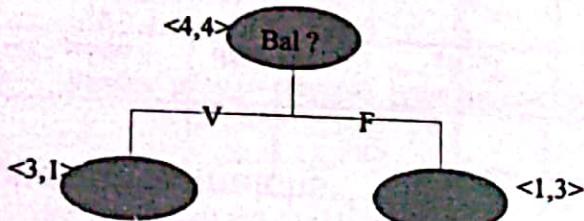
- Avec le critère de Gini et en désignant les attributs descriptifs *Dom*, *Bal*, *MCC* et *MPG* nous avons :



$$\begin{aligned} \text{Gain}(\epsilon, \text{Dom}) &= \text{Gini}(S) - (5/8\text{Gini}(S_1) + 3/8 \text{Gini}(S_2)) \\ &= \text{Gini}(S) - 2*5/8*3/5*2/5 - 2*3/8*1/3*2/3 \\ &= \text{Gini}(S) - 7/15 \end{aligned}$$

Exemple

- Avec le critère de Gini et en désignant les attributs descriptifs *Dom*, *Bal*, *MCC* et *MPG* nous avons :

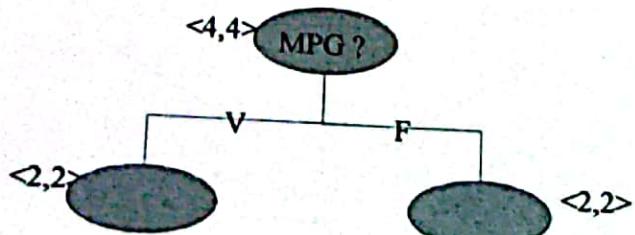


$$\begin{aligned} \text{Gain}(\epsilon, \text{Bal}) &= \text{Gini}(S) - (4/8\text{Gini}(S_1) + 4/8 \text{Gini}(S_2)) \\ &= \text{Gini}(S) - 2*4/8*3/4*1/4 - 2*4/8*1/4*3/4 \\ &= \text{Gini}(S) - 3/8 \end{aligned}$$

13

Exemple

- Avec le critère de Gini et en désignant les attributs descriptifs *Dom*, *Bal*, *MCC* et *MPG* nous avons :

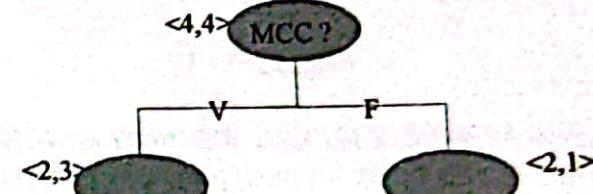


$$\begin{aligned} \text{Gain}(\epsilon, \text{MPG}) &= \text{Gini}(S) - (4/8\text{Gini}(S_1) + 4/8 \text{Gini}(S_2)) \\ &= \text{Gini}(S) - 2*4/8*2/4*2/4 - 2*4/8*2/4*2/4 \\ &= \text{Gini}(S) - 1/2 \end{aligned}$$

15

Exemple

- Avec le critère de Gini et en désignant les attributs descriptifs *Dom*, *Bal*, *MCC* et *MPG* nous avons :



$$\begin{aligned} \text{Gain}(\epsilon, \text{MCC}) &= \text{Gini}(S) - (5/8\text{Gini}(S_1) + 3/8 \text{Gini}(S_2)) \\ &= \text{Gini}(S) - 2*5/8*2/5*3/5 - 2*3/8*2/3*1/3 \\ &= \text{Gini}(S) - 7/15 \end{aligned}$$

14

Exemple

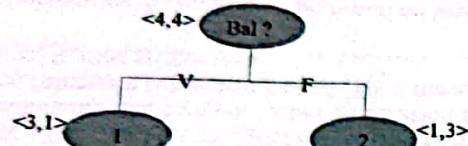
- Gains obtenus avec le critère de Gini:

$$\begin{aligned} \text{Gain}(\epsilon, \text{Dom}) &= \text{Gini}(S) - 0,46 \\ \text{Gain}(\epsilon, \text{Bal}) &= \text{Gini}(S) - 0,375 \\ \text{Gain}(\epsilon, \text{MCC}) &= \text{Gini}(S) - 0,46 \\ \text{Gain}(\epsilon, \text{MPG}) &= \text{Gini}(S) - 0,5 \end{aligned}$$

- Le gain maximal est obtenu pour le test **Balance positive (Bal)**.

Exemple

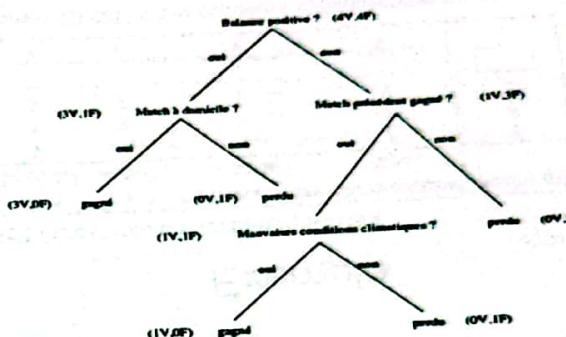
- En choisissant l'attribut **balance positive (Bal)** à la racine, l'arbre courant est alors :



- Pour poursuivre la construction de l'arbre, il faut répéter récursivement (et indépendamment) le calcul du gain en position 1 et en position 2 pour choisir les tests à ces niveaux.

17

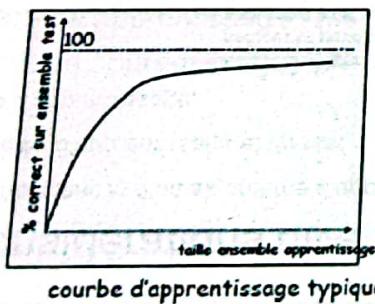
Exemple: Arbre final construit T0



18

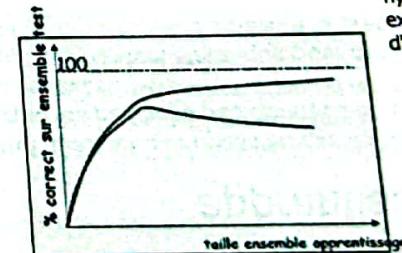
Considérations diverses

- Mesure des performances d'un algorithme d'apprentissage,
 - ensembles d'apprentissage (S) et de test (T),
Erreur réelle=nombre d'exemples mal-classés(T)/T
 - courbe d'apprentissage,



Considérations diverses

- Mesure des performances d'un algorithme d'apprentissage,
 - ensembles d'apprentissage et de test,
 - courbe d'apprentissage,
- Sur-apprentissage → Risque d'utiliser des attributs non pertinents pour générer une hypothèse en accord avec tous les exemples de l'ensemble d'apprentissage



Considérations diverses

- Mesure des performances d'un algorithme d'apprentissage,
 - ensembles d'apprentissage et de test,
 - courbe d'apprentissage,

- Sur-apprentissage
 - élagage de l'arbre
 - validation croisée

Risque d'utiliser des attributs non pertinents pour générer une hypothèse en accord avec tous les exemples de l'ensemble d'apprentissage

Terminer la récursion quand le gain d'information est trop faible

L'arbre de décision résultant peut ne pas classifier correctement tous les exemples de l'ensemble d'apprentissage

Comment corriger le sur-apprentissage

- Arrêt aussi tôt que possible: cesser de construire l'arbre lorsque les partitions des données ne sont plus statistiquement significatives,
- appliquer un test statistique pour déterminer si la partition produite par un noeud améliore la précision sur l'ensemble de test

- Élagage à posteriori: construire l'arbre entier et ensuite systématiquement supprimer les noeuds inutiles selon un test approprié, par exemple lorsque le gain en information est trop faible.

- Validation croisée: fragmenter l'ensemble d'apprentissage pour statistiquement mieux évaluer le moment d'arrêt.

Elagage d'arbre de décision (CART)

Soit T_0 l'arbre obtenu après apprentissage:

$\alpha = \Delta R_{emp}(S) / |T_p| - 1$ 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 $T_0, \dots, T_i, \dots, T_t$ la suite obtenue, T_t étant réduit à une feuille. On sélectionne l'arbre T_j dont le nombre d'erreurs calculé sur un ensemble de validation S_{val} est minimal.

Exemple

Soit 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	V

L'arbre T_0 est l'arbre construit précédemment:

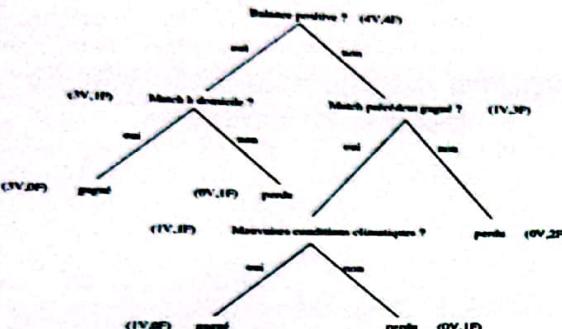
- T_1 est l'arbre obtenu en élaguant à partir de la position 2

- T_2 est obtenu en élaguant à partir de la position 1.

- T_3 est réduit à une feuille, portant la classe gagné.

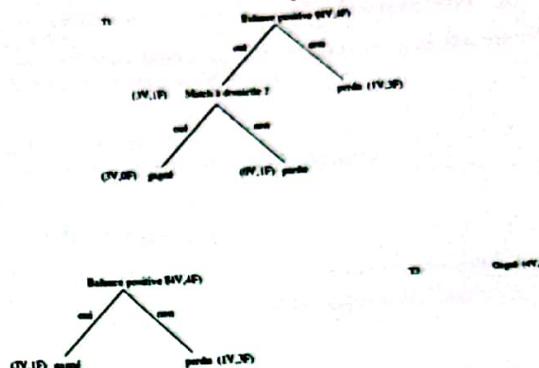
L'algorithme d'élagage retourne l'arbre T_2 .

Exemple: Arbre final construit T0



18

Exemple



74

- Performances- erreur de classification :
- Arbre t0 : 0 en apprentissage, 1/2 en test.
- Arbre t1 : 1/4 en apprentissage, 1/2 en test.
- Arbre t2 : 1/2 en apprentissage, 1/4 en test.
- Arbre t3 : 1/2 en apprentissage, 1/4 en test.

Implémentations des algo des arbres de décision

- ID3 – Inductive Decision Tree (Quinlan 1979)
 - arbre de discrimination (traite les variables qualitatives de type nominal = nom de personne, ou ordinal = rang) ;
 - critère d'homogénéité : entropie.

- C4.5 (Quinlan 1993)
 - amélioration de ID3 ;
 - arbre de régression (gère les variables continues) ;
 - critère d'homogénéité: entropie.
 - J48 (implémentation de C4.5 sous Weka).

- CART – Classification And Regression Tree (Breiman et al., 1984)
 - critère d'homogénéité: Gini.