Rech. Intell. et Coopération

Apprentissage de règles et arbres de décision

M2 IA&RF / IRR

Sept.-Nov. 2015

Introduction

Un agent dans une certaine situation / un certain état s doit prendre une décision

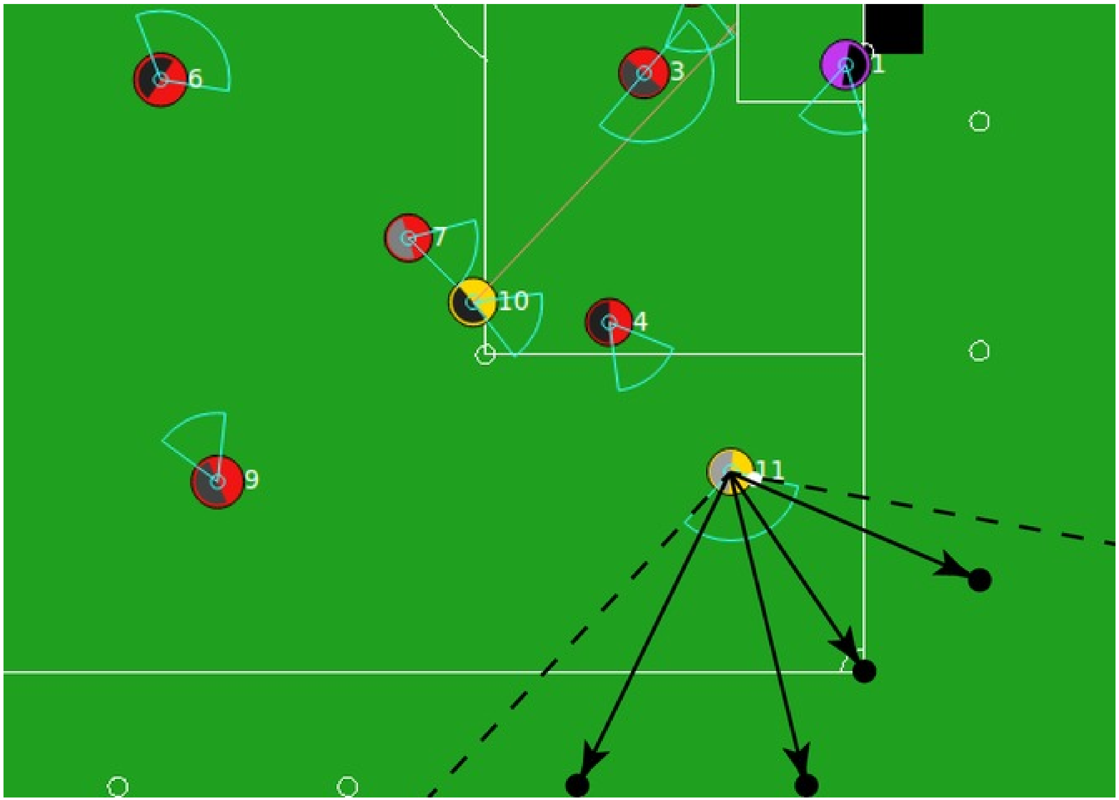
- plusieurs décision possibles, quelle est la « bonne » ?
- dépend de l'état / de la situation

Introduction : Exemples

Un joueur artificiel pour un jeu video

⇒ Que faire du ballon ? (Passe, tir, dribble . . .)

Dépends de la position du joueur sur le terrain, des positions de ses adversaires. . .



2D Simulation Leage – Robocup

Introduction : Exemples

Un banquier devant décider d'accorder ou non un prêt
$\Rightarrow$ dépends du demandeur : âge, profession, statut marital, biens immobiliers, salaire, montant du prêt, . . .

Introduction : Exemples

Un banquier devant décider d'accorder ou non un prêt
$\Rightarrow$ dépends du demandeur : âge, profession, statut marital, biens immobiliers, salaire, montant du prêt, . . .

Un médecin devant poser un diagnostique
$\Rightarrow$ dépend du résultats d'observations / tests / examens : fièvre, douleur, toux, rhinorrhée, . . .

Introduction

- Décision complexes : dépendent de nombreux facteurs
- Comment représenter le modèle de décision ?
  on veut un modèle compréhensible / interprétable par un humain
  $\Rightarrow$ règles / arbres de décision
- Comment *apprendre automatiquement* le modèle de décisio

Règles et arbres de décision

- Situations / états décrits par n attributs $X_1, X_2, \ldots, X_n$
- pour chaque attribut X_i : $\underline{X_i} = \mathsf{dom}(X_i)$ = ensemble des valeurs possibles pour X_i

$\Rightarrow$ l'ensemble des situations / états possibles est

$$\mathcal{S} = \prod_{X_i \in \mathcal{X}} \mathsf{dom}(X_i)$$

- un ensemble de décisions / *classes* possibles $\mathcal{Y} = \{y_1, \ldots\}$
- on veut représenter / apprendre une fonction $c : \mathcal{S} \longrightarrow \mathcal{Y}$

Règles et arbres de décision : Règles conjonctives

De la forme si $X_{i_1} = v_1$ et $X_{i_2} = v_2$ et $\ldots X_{i_k} = v_k$ alors y_j

Par exemple:

Règles et arbres de décision : Règles conjonctives

De la forme si $X_{i_1} = v_1$ et $X_{i_2} = v_2$ et $\ldots X_{i_k} = v_k$ alors y_j

Par exemple:

si position = devant_but $\wedge$ pos_gardien = a_droite $\wedge \ldots$ alors tirer_a_gauche

Règles et arbres de décision : Règles conjonctives

De la forme si $X_{i_1} = v_1$ et $X_{i_2} = v_2$ et $\ldots X_{i_k} = v_k$ alors y_j

Par exemple:

si position $=$ devant_but $\wedge$ pos_gardien $=$ a_droite $\wedge \ldots$ alors tirer_a_gauche

si age $\leq 40 \wedge$ proprietaire $=$ oui $\wedge \ldots$ alors pret_ok

Règles et arbres de décision : Règles conjonctives

De la forme si $X_{i_1} = v_1$ et $X_{i_2} = v_2$ et $\ldots X_{i_k} = v_k$ alors y_j

Par exemple:

si position $=$ devant_but $\wedge$ pos_gardien $=$ a_droite $\wedge \ldots$ alors tirer_a_gauche

si age $\leq 40 \wedge$ proprietaire $=$ oui $\wedge \ldots$ alors pret_ok

Mais:

- si age > 40 ?
- si pos_gardien $\neq$ a_gauche ?

$\Rightarrow$ une seule règle ne suffit pas

Règles et arbres de décision : Arbres

Un arbre de décision pour $X_1, \ldots, X_n$ est un arbre dont :

- chaque nœud interne est étiqueté par un test portant sur un ou plusieurs attributs;
- les arêtes sous les fils d'un nœud interne donné sont étiquetées par des réponses possibles mutuellement exclusives au test porté par ce nœud;
- les feuilles sont étiquetées par des décisions / classes de $\mathcal{Y}$.

Règles et arbres de décision : Arbres

Un arbre de décision pour $X_1, \ldots, X_n$ est un arbre dont :

- chaque nœud interne est étiqueté par un test portant sur un ou plusieurs attributs;
- les arêtes sous les fils d'un nœud interne donné sont étiquetés par des réponses possibles mutuellement exclusives au test porté par ce nœud;
- les feuilles sont étiquetées par des décisions / classes de $\mathcal{Y}$.

Un arbre de décision $\mathcal{A}$ représente une fonction $h_{\mathcal{A}} : \mathcal{S} \to \mathcal{Y}$, définie de la manière suivante :

> pour $x \in \mathcal{S}$, s'il existe une branche de l'arbre dont x vérifie toutes les conditions, alors $h_{\mathcal{A}}(x)$ est la valeur portée par la feuille correspondante.

Règles et arbres de décision : Arbres

Un arbre de décision pour $X_1, \ldots, X_n$ est un arbre dont :

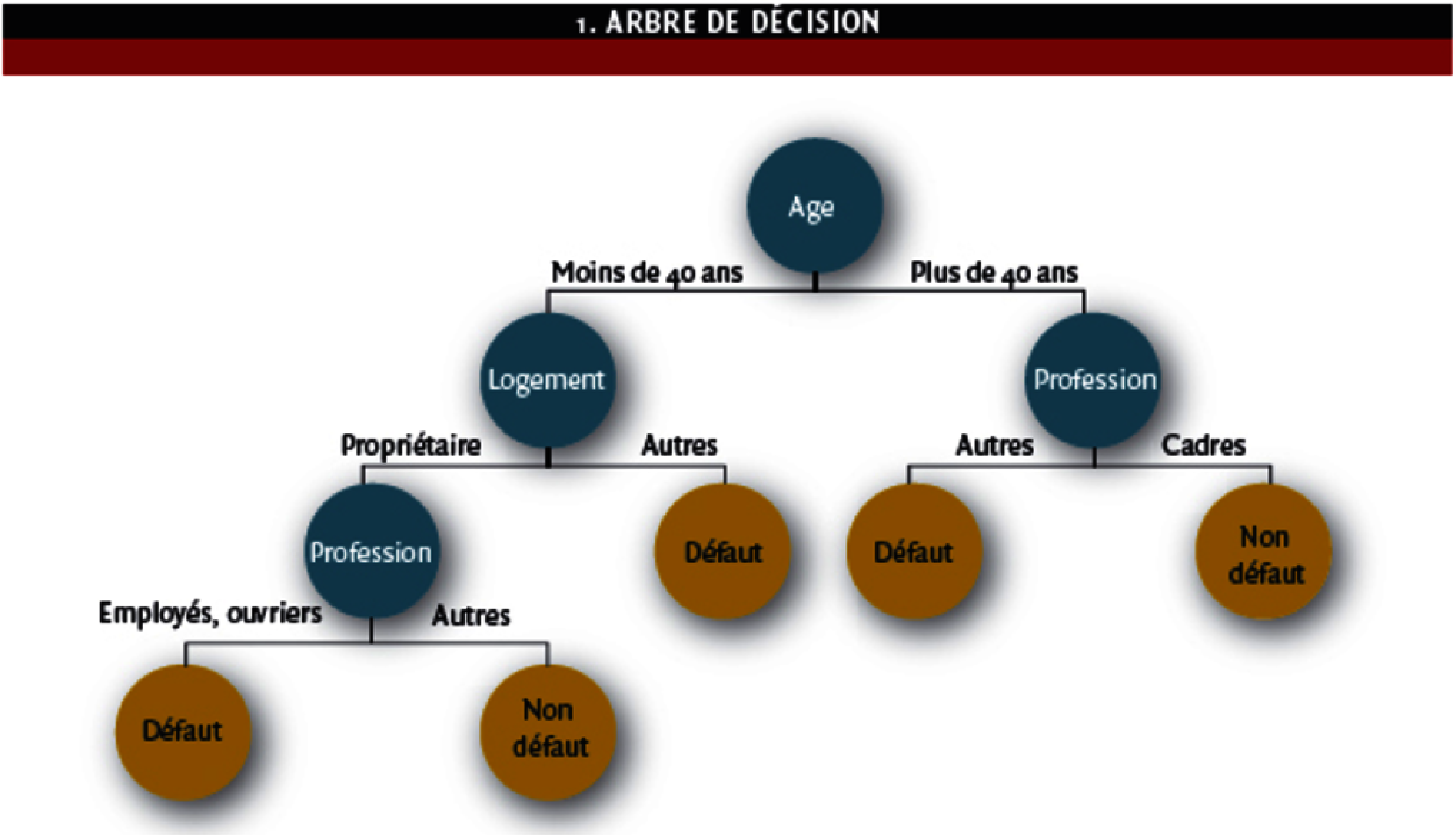
- chaque nœud interne est étiqueté par un test portant sur un ou plusieurs attributs;
- les arêtes sous les fils d'un nœud interne donné sont étiquetés par des réponses possibles mutuellement exclusives au test porté par ce nœud;
- les feuilles sont étiquetées par des décisions / classes de $\mathcal{Y}$.

Un arbre de décision $\mathcal{A}$ représente une fonction $h_\mathcal{A} : \mathcal{S} \to \mathcal{Y}$, définie de la manière suivante :

> pour $x \in \mathcal{S}$, s'il existe une branche de l'arbre dont x vérifie toutes les conditions, alors $h_\mathcal{A}(x)$ est la valeur portée par la feuille correspondante.

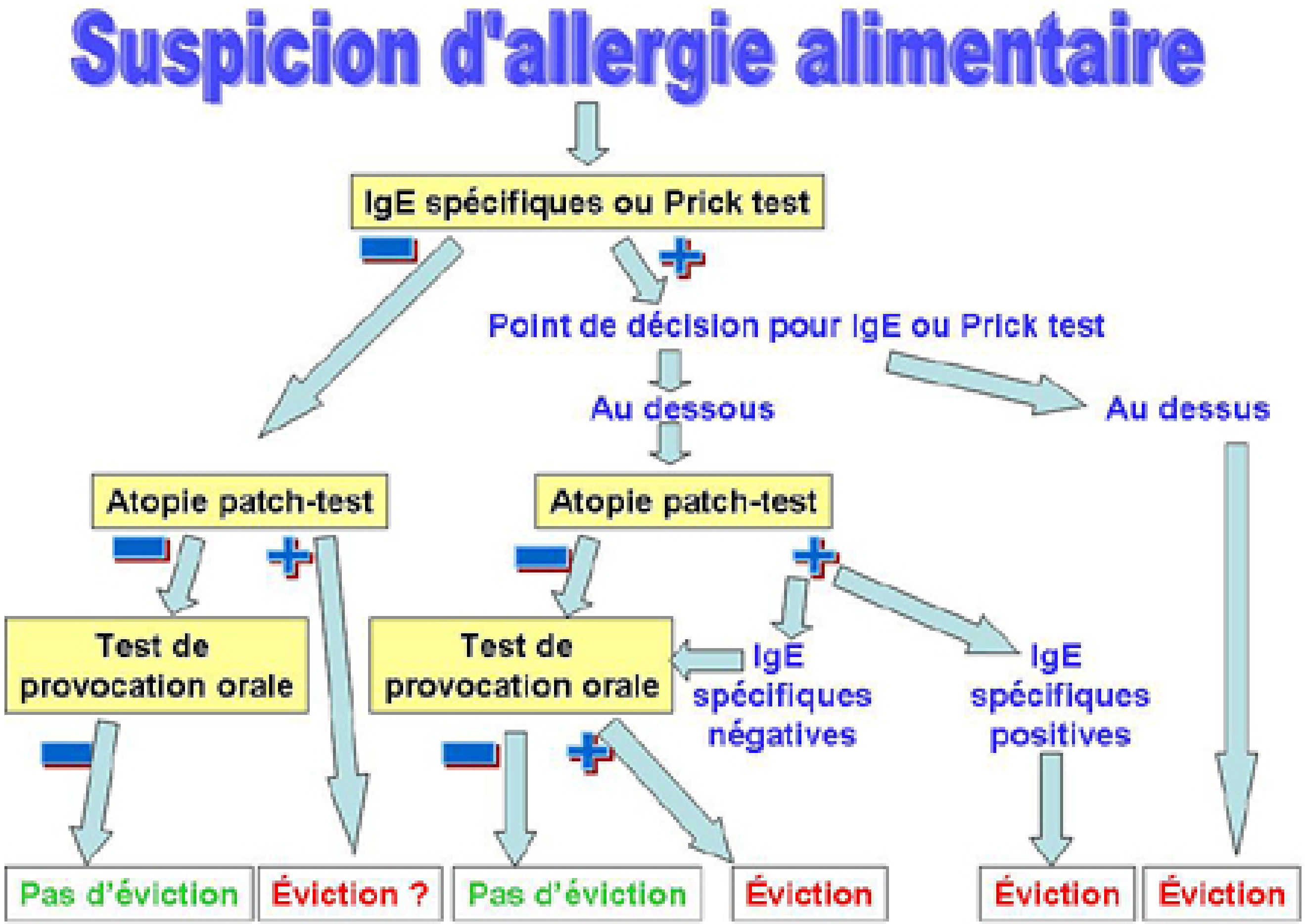
Intérêt des arbres de décision: facile à lire / interpréter ; représentation compacte

Règles et arbres de décision : Arbres



revue – bancaire.fr– 28 Octobre 2014

Règles et arbres de décision : Arbres



Règles et arbres de décision : Arbres

Propriétés

- À chaque branche d'un arbre de décision correspond une règle conjonctive.
  $\Rightarrow$ on peut traduire un arbre en un ensemble de règles
- Toute fonction $\mathcal{S} \longrightarrow \mathcal{Y}$ peut être représentée par un / plusieurs arbre(s) de décision
  au pire, une branche par état / situation
  $\Rightarrow$ taille potentiellement exponentielle

Apprentissage supervisé

Plutôt que demander à un expert de construire l'enemble de règles ou l'arbre de décision, on peut :

- lui demander la bonne décision dans certaines situations
- *induire* un arbre / un ensemble de règles

Apprentissage supervisé

Plutôt que demander à un expert de construire l'enemble de règles ou l'arbre de décision, on peut :

- lui demander la bonne décision dans certaines situations
- *induire* un arbre / un ensemble de règles

On appelle *exemple* une paire $(x, y) \in \mathcal{S} \times \mathcal{Y}$:

- x représente un état / une situation ;
- y est la décision correspondante

Apprentissage supervisé

Plutôt que demander à un expert de construire l'enemble de règles ou l'arbre de décision, on peut :

- lui demander la bonne décision dans certaines situations
- *induire* un arbre / un ensemble de règles

On appelle *exemple* une paire $(x, y) \in \mathcal{S} \times \mathcal{Y}$:

- x représente un état / une situation ;
- y est la décision correspondante

Étant donné un ensemble $\mathcal{E}$ d'exemples, on cherche $h : \mathcal{S} \longrightarrow \mathcal{Y}$ telle que $y = h(x)$ pour tout $(x, y) \in \mathcal{E}$.

Apprentissage supervisé : Une règle

Pour la suite, on distingue une valeur $y_0 \in \mathcal{Y}$;
pour simplifier la présentation, on note $y_0 = +1$.
$\Rightarrow$ on cherche à apprendre des règles de la forme

$$\text{si } X_{i_1} = v_1 \text{ et } X_{i_2} = v_2 \text{ et } \ldots X_{i_k} = v_k \text{ alors } +1$$

qui permettront de prédire si $c(x) = +1$ ou $c(x) \neq +1$.

Apprentissage supervisé : Une règle

Notations et terminologie :

- Pour une règle $r =$
« si $X_{i_1} = v_1$ et $X_{i_2} = v_2$ et $\ldots X_{i_k} = v_k$ alors $+1$ » on note :

$$\text{Cond}(r) = \text{« } X_{i_1} = v_1 \text{ et } X_{i_2} = v_2 \text{ et } \ldots X_{i_k} = v_k \text{ »}$$

- Un exemple $(x, y) \in \mathcal{E}$ est: *positif* si $y = +1$; *négatif* sinon ;
- La règle
« si $X_{i_1} = v_1$ et $X_{i_2} = v_2$ et $\ldots X_{i_k} = v_k$ alors $+1$ » *couvre* un exemple (x, y) si $X_{i_j}(x) = v_{i_j}$ pour $j \in 1 \ldots k$.

Étant donné un ensemble d'exemples $\mathcal{E}$, on cherche donc à calculer une règle qui :

- couvre autant d'exemples positifs que possible ; et
- couvre aussi peu d'exemples négatifs que possible.

Apprentissage supervisé : Une règle

Principe :

- On part de la règle trivial « Si vrai alors $+1$ » ;
- cette règle couvre trop d'exemples négatifs (elles les couvre tous) ;
- donc on essaie d'ajouter petit à petit des conditions élément / atomes de la forme $A_i = v_i$, pour exclure les exemples négatifs, en excluant le moins possible d'exemples positifs.

L'algorithme :

(a) Cond $\leftarrow$ « rien » ;
$E^+ \leftarrow \{x \mid (x, +1) \in \mathcal{E}\}$;
$E^- \leftarrow \{x \mid (x, -1) \in \mathcal{E}\}$;

(b) Tant que $E^- \neq \emptyset$ faire :

  i. $\alpha \leftarrow$ le « meilleur » atome ;

  ii. Cond $\leftarrow$ Cond et α ;

  iii. $E^- \leftarrow E^- - \{x \mid x$ ne vérifie pas Cond$\}$;
  $E^+ \leftarrow E^+ - \{x \mid x$ ne vérifie pas Cond$\}$;

(c) retourner $r =$ « Si Cond alors $+1$ ».

Apprentissage supervisé : Une règle

Un exemple On a enregistré sur 14 jours si des voiliers étaient sortis sur la mer, ainsi que l'ensoleillement ou non ces jours-là, la température, le taux d'humidité, le vent :

Apprentissage supervisé : Une règle

	ciel	temp.	humid.	vent	bateaux ?
1	soleil	chaude	haute	non	non
2	soleil	chaude	haute	oui	non
3	couvert	chaude	haute	non	oui
4	pluie	moy.	haute	non	oui
5	pluie	froide	normal	non	oui
6	pluie	froide	normal	oui	non
7	couvert	froide	normal	oui	oui
8	soleil	moy.	haute	non	non
9	soleil	froide	normal	non	oui
10	pluie	moy.	normal	non	oui
11	soleil	moy.	normal	oui	oui
12	couvert	moy.	haute	oui	oui
13	couvert	chaude	normal	non	oui
14	pluie	moy.	haute	oui	non

$\Rightarrow$ peut-on prédire, pour un autre jour, en fonction du soleil, de la température, du taux d'humidité, du vent, si on verra

Apprentissage supervisé : Une règle

ou non des bateaux ?

$\Rightarrow$ peut-on « découvrir »/*induire*, d'après ce tableau, des *règles* de prédiction, par exemple :

$$\text{si ciel} = \text{soleil et temp} = \text{chaude et humid.} = \text{haute}$$
$$\text{alors non}$$

Apprentissage supervisé : Une règle

Choix d'un « bon » atome On cherche, parmi tous les $X_i = v_i$ possibles, un atome qui :

- exclue autant d'exemples négatifs que possible ; et
- exclue aussi peu d'exemples positifs que possible.

Pour chaque atome $\alpha =$ « $X_i = v_i$ », on note

- $p(\alpha) = |\{x \in E^+ \mid X_i(x) = v_i\}|$
  $=$ le nombre d'exemples de E^+ « couverts » par α ;
- $n(\alpha) = |\{x \in E^- \mid X_i(x) = v_i\}|$
  $=$ le nombre d'exemples de E^- « couverts » par α ;

$\Rightarrow$ on veut maximiser $p(\alpha)$ tout en minimisant $n(\alpha)$.

Exemples de critères de maximisation possibles :

- $p(\alpha) - n(\alpha)$
- $p(\alpha) \ / \ (p(\alpha) + n(\alpha))$
- $p(\alpha) \times (\ \log(p(\alpha)/(p(\alpha) + n(\alpha))) - c\ )$ (*gain d'information*)

(Voir par exemple [**?**])

Apprentissage supervisé : Un ensemble de règles

En général, ça ne suffit pas d'apprendre une règle :
la règle retournée par l'algorithme ci-dessus ne couvre pas d'exemples négatif, mais ne couvre sans doute pas tous les exemples positifs.

On va donc cherche à apprendre d'autres règles, qui permette d'apprendre d'autres exemples : on insère l'algorithme ci-dessus dans un boucle qui fait apprendre des règles jusqu'à ce que tous les exemples positifs soient couverts.

Apprentissage supervisé : Un ensemble de règles

Algorithme de couverture séquentielle

1. $h \leftarrow \emptyset$; $\mathcal{E}^+ \leftarrow \{x \mid (x, +1) \in \mathcal{E}\}$; $\mathcal{E}^- \leftarrow \{x \mid (x, -1) \in \mathcal{E}\}$;
2. Tant que $\mathcal{E}^+ \neq \emptyset$ faire:
   (a) Cond $\leftarrow$ « rien » ; $E^- \leftarrow \mathcal{E}^-$; $E^+ \leftarrow \mathcal{E}^+$;
   (b) Tant que $E^- \neq \emptyset$ faire :
      i. $\alpha \leftarrow$ le « meilleur » atome ;
      ii. Cond $\leftarrow$ Cond et α ;
      iii. $E^- \leftarrow E^- - \{x \mid x$ ne vérifie pas Cond$\}$;
      $E^+ \leftarrow E^+ - \{x \mid x$ ne vérifie pas Cond$\}$;
   (c) $h \leftarrow h \cup \{$« Si Cond alors $+1$ »$\}$;
   (d) $\mathcal{E}^+ \leftarrow \mathcal{E}^+ - \{x \in \mathcal{E}^+ \mid x$ non couvert par $h\} = \mathcal{E}^+ - E^+$;
3. Retourner h.

(On dit qu'un exemple (x, y) est *couvert* par un ensemble de règles h s'il y a dans h au moins une règle qui couvre (x, y).)

Terminaison de l'algorithme

- Etant donnés $\mathcal{E}^+$ et $\mathcal{E}^-$ tels que $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$ et $\mathcal{E}^+ \neq \emptyset$, on peut toujours trouver un ensemble de règles qui classe

Apprentissage supervisé : Un ensemble de règles

bien:

$$h(x) = \begin{cases} +1 \text{ si } x \in \mathcal{E}^+ \\ -1 \text{ sinon} \end{cases}$$

- Etant donnés E^+ et E^-, tels que $E^+ \cap E^- = \emptyset$ et $E^+ \neq \emptyset$, et une condition Cond vérifiée par tout $x \in E^+$, on peut toujours trouver une condition Cond' telle que Cond(x) et Cond$'(x)$ faux pour tout $x \in E^-$: on choisit un $x_0 \in E^+$, et on pose Cond$'(x) =$ vrai si $x = x_0$.

Apprentissage supervisé : Arbres de décision

Un algorithme :

Entrée : un ensemble $\mathcal{E} \subseteq \underline{\mathcal{X}} \times \mathcal{Y}$;

1. si pour tout $(x, y), (x', y') \in \mathcal{E}$ on a $y = y'$: retourner cette valeur ;
2. $A \leftarrow$ le « meilleur » attribut de $\mathcal{X}$;
3. créer un nœud étiqueté par A;
4. pour chaque $v \in \underline{A}$ faire:
   (a) créer une branche étiquetée par v;
   (b) y attacher un arbre de décision pour $\mathcal{E}_v = \{(x, y) \in \mathcal{E} \mid A(x) = v\}$

Apprentissage supervisé : Arbres de décision

Choix du « meilleur » attribut On va essayer de trouver l'attribut amenant une *entropie* minimale.

Étant donnés un ensemble d'exemples $\mathcal{E}$, un attribut A et une valeur $v \in \mathsf{dom}(A)$:

- on note : $\mathcal{E}_v = \{(x, y') \in \mathcal{E} \mid A(x) = v\}$;
  pour chaque $y \in \mathcal{Y}$:
  $p_v^y = |\{(x, y') \in \mathcal{E} \mid A(x) = v \text{ et } y' = y\}|/|\mathcal{E}_v|$;
  p_v^y est la proportion d'exemples qui ont l'étiquette y parmi ceux qui ont la valeur v pour l'attribut A dans $\mathcal{E}$.
- Alors on définit l'*entropie de* $\mathcal{E}$ pour l'attribut A et la valeur v ainsi :

$$\mathsf{Entropie}(\mathcal{E}, A, v) = -\sum_{y \in \mathcal{Y}} p_v^y \log p_v^y$$

Propriétés de l'entropie :

- s'il existe une valeur y telle que $p_v^y = 1$:
  alors $\mathsf{Entropie}(\mathcal{E}, A, v) = 0$;
  sinon $\mathsf{Entropie}(\mathcal{E}, A, v) > 0$;
- la valeur maximale est atteinte lorsque $p_v^y = 1/|\mathcal{Y}|$ pour toute $y \in \mathcal{Y}$.

Remarque : en théorie de l'information, l'entropie mesure l'incertitude d'une variable aléatoire. Si X peut prendre deux valeurs 0 et 1 avec une probabilité 1/2 pour chacune, l'incertitud sur la valeur qui peut être prise est maximale. Si par contre la probabilité d'avoir 1 est par exemple $0,9$, l'incertitude est très faible (c'est très probable d'avoir 1).

On va donc définir le gain ainsi:

$$G(\mathcal{E}, A) = \mathsf{Entropie}(\mathcal{E}) - \sum_{v \in \mathsf{dom}(A)} \frac{|\mathcal{E}_v|}{|\mathcal{E}|} \mathsf{Entropie}(\mathcal{E}, A, v)$$

Le second terme est la moyenne des entropies des ensembles obtenus si on branche sur l'attribut A, pondérée par la taille de chacun de ces ensembles.

Apprentissage supervisé : Arbres de décision

Remarque: cette expression du gain favorise les attributs qui ont beaucoup de valeurs possibles. Par exemple, un attribut qui a une valeur différente pour chaque exemple donne un gain maximal, car l'entropie de chaque classe est nulle.

Pour remédier à cela, on peut diviser le gain par le terme suivant:

$$\sum_{v \in \mathsf{dom}(A)} \frac{|\mathcal{E}_v|}{|\mathcal{E}|} \log_2 \frac{|\mathcal{E}_v|}{|\mathcal{E}|}$$

Apprentissage supervisé : Attributs numériques

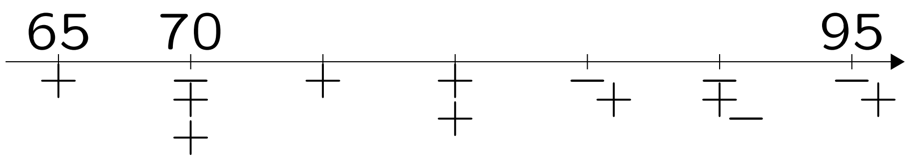
ciel	temp.	humid.	vent	bateaux ?
soleil	85	85	non	non
soleil	80	90	oui	non
couvert	83	86	non	oui
pluie	70	96	non	oui
pluie	68	80	non	oui
pluie	65	70	oui	non
couvert	64	65	oui	oui
soleil	72	95	non	non
soleil	69	70	non	oui
pluie	75	80	non	oui
soleil	75	70	oui	oui
couvert	72	90	oui	oui
couvert	81	75	non	oui
pluie	71	91	oui	non

Apprentissage supervisé : Attributs numériques

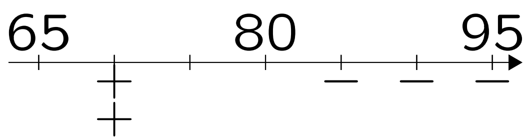
Checher une bonne valeur de séparation

- On cherche un valeur α pour construire deux branches définies par $A > \alpha$ et $A < \alpha$.
- On teste les points possibles « entre deux classes »
- On garde celui qui donne la meilleure entropie
- On peut refaire d'autres branchements sur le même attribut, mais avec d'autres valeurs, plus bas dans l'arbre.

Sur l'exemple :



Si on ne considère que les exemples où ciel = soleil:



On peut prendre comme atomes humid $\geq 77,5$ et humid $< 77,5$.

Surapprentissage

Il y a en général risque de surapprentissage, si les exemples sont bruités, ou si on a peu d'exemples (auquel cas un attribut peut sembler bien séparer les exemples, alors qu'il n'est en réalité pas du tout lié au concept à apprendre).

Pour pallier à cette difficulté, on peut décider d'arrêter l'algorith avant d'avoir atteint des feuilles pures. On décide alors d'une précision minimale qu'on veut pour l'arbre, ou pour chaque feuille.

On peut garder une partie des exemples à part, afin de vérifier sur ces exemples la validité de l'hypothèse apprise. On peut aussi faire de la validation croisée: on apprend une hypothèse sur une première moitié des exemples, puis on la teste sur l'autre moitié. On apprend ensuite une seconde hypothèse sur cette dernière moitié des exemples, puis on la teste sur la première moitié: on garde la meilleur des deux hypothèses.

Élagage

Une autre manière de pallier au surapprentissage consiste à élaguer les règles/arbres appris.

Dans le cas d'arbres de décision, on peut essayer de transformer certains sous-arbres en feuilles: on affecte à cette feuille la classification la plus fréquente dans l'ensemble de validation. On n'effectue la coupe que si l'arbre obtenue n'est pas moins bon que l'arbre courant sur l'ensemble de validation.

Dans le cas de règles, on peut essayer de supprimer des atomes dans les règles.

Exercices

Exercice 1 Une agence de voyages vend des vacances à la mer sur son site internet `vacances.mer`. Afin de faciliter le choix des clients fidèles en ne proposant que des offres correspondant à leurs habitudes, le site mémorise les choix effectués chaque année, ainsi qu'une variable indiquant si le client a été satisfait ou non: le but est d'apprendre quel type de vacances satisfait un client donné.

Pour la famille Ploucs, le tableau ci-contre montre les températu moyennes de l'eau et de l'air à midi pour les vacances choisies lors des 10 années de 1998 à 2007, ainsi que le prix, et indique si les Ploucs ont été satisfaits de leur vacances $(+)$ ou non $(-)$.

On veut apprendre un arbre de décision permettant de sélection les lieux de acances en fonction des valeurs des attributs Eau, Air et Prix.

Exercices

Année	Satisfaction	Eau	Air	Prix
1998	-1	m	b	b
1999	-1	m	h	m
2000	$+1$	m	m	m
2001	-1	h	m	h
2002	-1	m	m	m
2003	-1	m	b	m
2004	$+1$	m	m	h
2005	$+1$	m	m	b
2006	-1	h	m	b
2007	-1	m	h	h

Exercices

Question 1.1 Lequel de ces trois attributs semble le mieux pour classer les 10 exemples du tableau ci-dessus ?

Question 1.2 Calculez un arbre de décision de hauteur 2 permettant de classer du mieux possible les exemples, et donnez l'ensemble de règles correspondant à cet arbre.

Question 1.3 Quel est le risque empirique associé à cet arbre (la proportion d'exemples mal classés parmi les exemples utilisés pour l'apprentissage) ? Pourrait-on obtenir un arbre de hauteur quelconque de risque empirique moindre ?