

Arbres de décision

Ingénierie des connaissances (Master 2 ISC)

Introduction

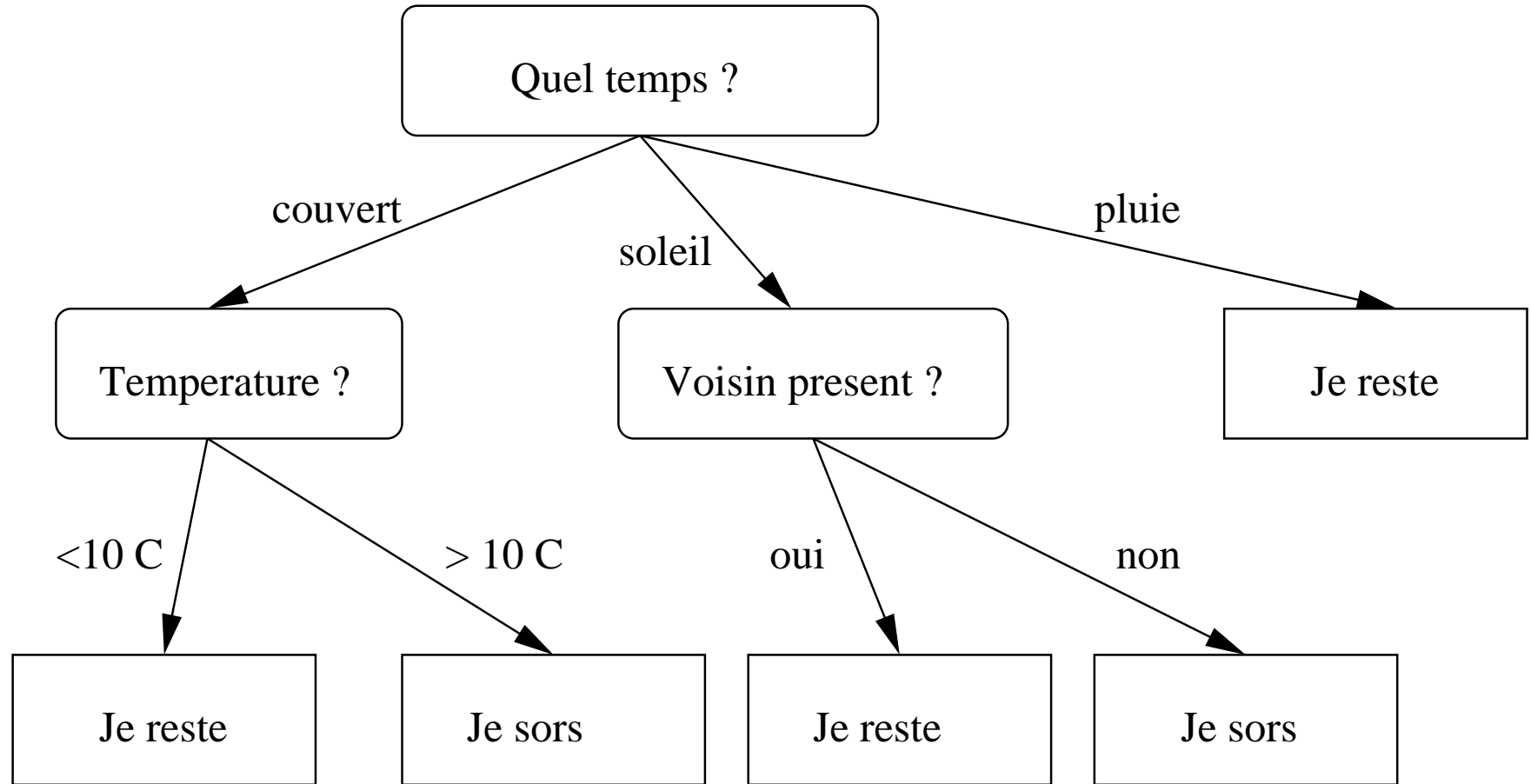
Pour résoudre un problème de **classification**, on peut utiliser des **règles de production** (exemple : diagnostique médical avec MYCIN).

Les **arbres de décision** sont une représentation commode de *fonctions de classifications*, moins puissants que les règles de production mais plus facile à utiliser.

Un **arbre de décision** permet de classer un objet à l'aide de **questions** : chaque nœud de l'arbre représente une question, chaque lien est une réponse à la question, et chaque feuille est une classe.

Exemple d'arbre de décision

Vais-je sortir mon chien ?



Un système déductif fait tout aussi bien l'affaire.

Avantage des arbres de décision

1. Facile à comprendre et à utiliser.
2. Nombre de tests limités par le nombre d'attributs (de *question*).
3. **Construction** efficace (mais technique) à l'aide d'apprentissage par optimisation (pour obtenir un arbre **petit** et « **correct** »).

On va particulièrement s'intéresser au troisième point.

Il n'est pas impossible d'apprendre à la place des règles de production, mais c'est plus délicat.

Position du problème

Pour notre apprentissage, on dispose d'un ensemble d'exemples $E = (\mathbf{x}, c)$. Chaque exemple est constitué :

1. d'un ensemble \mathbf{x} de *réponses* aux questions (possibles) de l'arbre. On appelle ces réponses (avec les questions qui y sont associés) des **attributs**.
2. de la classe c de l'exemple, qui appartient à l'ensemble des classes possibles \mathcal{C} .

Principe de construction

Construction récursive, en découpant successivement l'ensemble d'exemples E .

1. Si tous les exemples sont dans une seule classe, on place une feuille de cette classe.
2. Sinon, on choisit une question (la plus discriminante possible), on découpe l'ensemble d'exemples suivant cette question. Pour chaque nouvel ensemble, on construit un sous-arbre de décision.

Note : il est classique d'*élaguer* l'arbre ensuite à l'aide d'un ensemble d'exemples supplémentaire.

Algorithme (cas binaire)

On suppose que toutes les questions ont comme réponse *oui* ou *non*.

Procédure Construit_arbre(X)

debut

 Si tous les points de X sont dans la même classe alors

 Créer une feuille de cette classe

 sinon

 choisir le meilleur *sélecteur* (la meilleure question)

 pour créer un nœud. Séparer X suivant ce sélecteur en X_d et X_g .

 Construit_arbre(X_d)

 Construit_arbre(X_g)

 finsi

Fin

Exemple 2

Un écolier peut-il aller jouer dehors ? Voilà ce qu'il a observé :

Devoir finis ?	Mère de bonne humeur ?	Fait beau ?	Gouter pris ?	Réponse
oui	non	oui	non	oui
non	oui	non	oui	oui
oui	oui	oui	non	oui
oui	non	oui	oui	oui
non	oui	oui	oui	non
non	oui	non	non	non
oui	non	non	oui	non
oui	oui	non	non	non

Choix de la question

On dispose de 4 questions (DF, MBH, FB, GP). Chaque question sépare les réponses positives (4 en tout) et négatives (4 en tout) en deux ensembles.

Par exemple, avec la question « Devoirs finis ? », on trouve trois réponses positives et deux réponses négatives pour OUI, et une réponse négative et deux réponses positives pour NON.

L'idée est de maximiser la séparation globale. Les formules utilisées dérivent de la **théorie de l'information** : on cherche à maximiser le gain d'information.

Information : éléments techniques (1)

Les calculs dérivent des probabilités.

L'**entropie** d'une variable ω pouvant prendre les valeurs $\{\omega_i\}_{i \in I}$ avec des probabilités $p(\omega_i)$ est définie comme :

$$H(\omega) = - \sum_{i \in I} p(\omega_i) \log(p(\omega_i))$$

L'entropie note l'*incertitude* sur la valeur de la variable.

Par exemple, initialement l'entropie de "Je peux aller jouer" est (en considérant le logarithme à base 2), $H(c) = 1$. Si on sait que les devoirs ne sont pas finis, elle devient $H(c) = 0,8$. La *décroissance* de l'entropie note une diminution de l'incertitude, donc une augmentation de l'information.

Information : éléments techniques (2)

Dans le cas de deux variables, ω pouvant prendre les valeurs $\{\omega_i\}$, et a pouvant prendre les valeurs $\{a_j\}$, on définit l'entropie de ω **conditionnée par a** comme :

$$H(\omega|a) = - \sum_{i,j} p(\omega_i \wedge a_j) \log p(\omega_i|a_j)$$

(note : lorsque $p(\omega_i|a_j) = 0$, $p(\omega_i \wedge a_j) = 0$ donc le terme doit être considéré comme nul).

Cette entropie représente une incertitude sur ω indépendamment de a . Elle est d'autant plus faible que a permet de différencier les valeurs de ω . Ainsi pour $a = \omega$, elle vaut 0 : connaître ω lève l'incertitude sur ω .

Calcul des entropies conditionnées

Pour chaque question :

1. Pour DF, $H = -\frac{3}{8} * \log \frac{3}{5} - \frac{2}{8} * \log \frac{2}{5} - \frac{1}{8} * \log \frac{1}{3} - \frac{2}{8} * \log \frac{2}{3} = 0,95$.
2. Pour FB, $H = -2 * \frac{3}{8} * \log \frac{3}{4} - 2 * \frac{1}{8} * \log \frac{1}{4} = 0,81$.
3. Pour MBH, $H = -\frac{2}{8} * \log \frac{2}{4} - \frac{2}{8} * \log \frac{2}{4} - \frac{3}{8} * \log \frac{3}{4} - \frac{1}{8} * \log \frac{1}{4} = 0,91$.
4. Pour GP, $H = 1$.

On choisit alors le test qui minimise l'entropie, soit FB, et on recommence.

Bilan

Introduction sommaire sur l'apprentissage d'arbres de décision. Appliqué avec succès dans divers systèmes experts (et autres). Exemple d'*apprentissage par optimisation* : le problème est moins dans la découverte d'un arbre que dans la mise en place d'un « petit » arbre.

La complexité du choix d'un attribut est linéaire dans le produit du nombre d'attributs et de la taille de l'ensemble d'apprentissage \Rightarrow tout à fait correct.

Extensions :

- ▶ Pour des attributs à plusieurs réponses (couleurs...), on peut soit traiter toutes les réponses à la fois, soit revenir à des questions oui/non.
- ▶ pour les attributs continus (numériques), on peut tester d'éventuels « seuils » dans les valeurs (exemple pour un poids : moins de 50 g, plus de 500 g, etc...).
- ▶ étendu pour traiter des données « bruitées » ou contradictoires.