Chapitre 2

Les arbres de décision classiques et flous

Présentation *: Ce chapitre comporte deux parties : dans un premier temps, on présente succinctement les techniques de création d'un arbre de décision classique son élagage, et quelques définitions comme l'entropie et le gain d'information et l'algorithme de développement de l'arbre comme l'ID3 ; la deuxième partie présente l'arbre de décision flou comme une extension de l'arbre de décision classique.*

1.10 Arbre de décision classique :

Un arbre de décision est un outil d'aide à la décision et à l'exploration de données. Il permet de modéliser simplement, graphiquement et rapidement un phénomène mesuré plus ou moins complexe. Sa lisibilité, sa rapidité d'exécution et le peu d'hypothèses nécessaires à priori expliquent sa vaste utilisation dans différents applications et reconnaissance de forme. Il est composé de :

- Une racine, qui est point de départ.
- des nœuds, ou sont réalisés des tests
- des feuilles, qui contiennent l'information cherchée.

Le parcours d'un arbre se fait, par tests successifs, de la racine à une feuille et constitue une règle de décision **[9]**.

1.10.1 Construction d'un arbre de décision :

1.10.1.1 Bases de règles :

Un arbre est un système de règles de type particulier : Il y a autant de règles que de feuilles et autant de tests que de nœuds avec des différences **[10]** :

- Les règles sont mutuellement exclusives.
- Une règle et une seule est entièrement évaluée.
- Les tests sur les attributs sont réalisés dans un ordre donné.

1.10.1.2 Prétraitements :

Nous attachons un test à chaque nœud et une réponse à chaque feuille ce qui définit la spécification des arbres de décision. Avec des avantages comme :

- ✓ Suite de tests mono variés.
- ✓ Pas de problème de combinaison de variables, de différentes natures.

- ✓ Hiérarchisation de variables intégrée.
- ✓ Rapidité d'exécution.
- ✓ Génération de règles logiques simples de classification.

1.10.1.3 Algorithme de l'arbre de décision:

- Lire le vecteur d'observation,'V'
- Se mettre à la racine
- Tant que (non feuille) faire
- N= testNoeud(V)
- Se positionner sur le fils N fait
- Ecrire la réponse

1.10.1.4 Phase d'apprentissage :

a) Entropie :

Le concept d'entropie est utilisé pour ordonner la liste des descripteurs avec le respect des ensembles de données, et des classes. L'entropie fournie une définition des descripteurs les plus significatifs [10].

Le calcul de l'entropie est donné par la formule mathématique suivante :

$$I(p) = \sum_{i=1}^{n} P(c_i) \times \log_2 P(c_i) \qquad (2\text{-}1)$$

Où $P(c_i)$ est la probabilité que la classe c_i soit correcte. La quantité $\log_2 P(c_i)$ est la quantité d'information que l'on donne quand la classe est la valeur attendue de ce contenu de l'information. Cette valeur attendue est la mesure d'entropie de l'ensemble.

On note $|X|$ cardinal de l'ensemble X. Si un ensemble T d'enregistrements partitionnés par la valeur de l'attribut catégorique en classes disjointes {C1, C2,…,Ck}, alors l'information nécessaire pour reconnaître la classe d'un élément de T est associée à l'entropie I(p) ,où P est la distribution de la probabilité de la partition (C1,C2, …,Ck) :

$$p = \frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, \ldots, \frac{|C_k|}{|T|} \qquad (2\text{-}2)$$

Si l'on partitionne notre ensemble T en valeurs disjointes, alors l'information nécessaire pour identifier la classe d'un élément de T devient :

$$Info(X,T) = \sum_{i=0}^{m} \frac{|T_i|}{|T|} \times Info(T_i) \qquad (2\text{-}3)$$

Où Ti est un sous ensemble de A pour la valeur i.

De là nous pouvons définir ce qu'est le gain, qui représente le gain obtenu dû au partitionnement par l'attribut X :

$$Gain(X, T) = Info(X) - Info(X,T) \qquad (2\text{-}4)$$

b) Test d'arrêt :

Les tests d'arrêt permettent de contrôler la croissance de l'arbre, En arrêtant éventuellement la création de nouveaux nœuds, même si tous les attributs n'ont pas été évalués.

Deux cas ne posent pas de problème :

- Toutes les observations associées à un nœud sont de la même classe,
- Tous les attributs ont été évalués.

Alors le nœud considéré est un nœud terminal.

Dans les autres cas, on définit des critères paramétrables :

- Une classe contient une proportion d'observations de la même classe supérieure à un seuil donné,
- Les sous-ensembles d'observations sortant d'un nœud sont trop petits, ce qui évite d'explorer des branches comportant trop peu d'éléments.

1.10.2 Elagage d'un arbre de décision :

L'objectif est de construire un arbre de taille réduite ayant à peu prés les mêmes performances que l'arbre initial.

- L'arbre initial est construit avec un ensemble d'apprentissage et se fait sur un ensemble de test.
- L'élagage consiste à remplacer un sous-arbre par une feuille, tant que les capacités de généralisation ne se dégradent pas **[9]**.

1.10.2.1 Le pré-élagage :

La première stratégie utilisable pour éviter un sur ajustement massif des arbres de décision consiste à proposer des critères d'arrêt lors de la phase d'expansion. C'est le principe du pré-élagage.

Nous considérons par exemple qu'une segmentation n'est plus nécessaire lorsque le groupe est d'effectif trop faible ; ou encore, lorsque la pureté d'un sommet a atteint un niveau suffisant, nous considérons qu'il n'est plus nécessaire de le segmenter ; autre critère souvent rencontré dans ce cadre, l'utilisation d'un test statistique pour évaluer si la segmentation introduit un apport d'information significatif quant à la prédiction des valeurs de la variable à prédire **[9]**.

1.10.2.2 Le post-élagage :

La seconde stratégie consiste à construire l'arbre en deux temps : produire l'arbre le plus pur possible dans une phase d'expansion en utilisant une première fraction de l'échantillon de données (échantillon d'apprentissage) une marche arrière pour réduire l'arbre, c'est la phase de post-élagage, en s'appuyant sur une autre fraction des données de manière à optimiser les performances de l'arbre.

Selon les logiciels, cette seconde portion des données est désignée par le terme d'échantillon de validation ou échantillon de test.

1.10.3 Algorithme de création de l'arbre :

1.10.3.1 Algorithme ID3 (Induction of Decision Tree):

a) Présentation :

J.Ross Quilan de l'université de Syndney, est le créateur de l'algorithme **ID3** Ce fut en 1975. Construire un arbre de décision d'un ensemble fixe d'exemples. L'arbre résultant est employé pour classifier de futurs exemples. L'exemple a plusieurs attributs et appartient à une classe (comme oui ou non). Les nœuds de feuille de l'arbre de décision contiennent le nom de classe contrairement aux nœuds de décision. Le nœud de décision est un essai d'attribut avec chaque branchement (à un autre arbre de décision) étant une valeur possible de l'attribut. Gain de l'information des utilisations ID3 pour l'aider à décider quel attribut entre dans un nœud de décision.

b) Contraintes de l'ID3:

Les exemples utilisés par ID3 doivent respecter certaines contraintes, tel que :

- Etre fixes.
- Les mêmes attributs doivent décrire chaque variable, et avoir un nombre fixe de valeurs.
- Un exemple d'attributs doit être déjà définit.

- Des classes bien délimitées.
- ID3 met en jeux deux concepts majeurs :
- L'entropie : concept permettant de trouver les paramètres les plus significatifs.
- Les arbres de décision : efficaces et intuitifs permettent d'organiser les descripteurs qui peuvent être utilisés comme fonction prédictive.

1.10.4 Autres algorithmes :

1.10.4.1 Algorithme C4.5 :

L'algorithme C4.5 a été élaboré par Quinlan en 1993, cet algorithme n'est en fait qu'une amélioration de ID3.

Cette méthode utilise un critère plus élaboré : « le gain ratio » dont le but est de limiter la prolifération de l'arbre en pénalisant les variables qui ont beaucoup de modalités. Contrairement à ID3, C4.5 est parfaitement réalisable dans des applications industrielles.

1.10.4.2 Algorithme CART (Classification and Regression Tree) :

Cette méthode permet d'inférer des arbres de décision binaires, i.e. tous les tests étiquetant les nœuds de décision sont binaires. Le langage de représentation est constitué d'un certain nombre d'attributs. Ces attributs peuvent être binaires, qualitatifs (à valeurs dans un ensemble fini de modalités) ou continus (à valeurs réelles). Le nombre de tests à explorer va dépendre de la nature des attributs. A un attribut binaire correspond un test binaire. A un attribut qualitatif ayant n modalités, on peut associer autant de tests qu'il y a de partitions en deux classes, soit 2^{n-1} tests binaires possibles. Enfin, dans le cas d'attributs continus, il y a une infinité de tests envisageables. Dans ce cas, on découpe l'ensemble des valeurs possibles en segments, ce découpage peut être fait par un expert ou fait de façon automatique **[9]**.

1.11 Arbre de décision flou :

1.11.1 Concepts de la logique floue:

Dans la logique classique, les variables gérées sont Booléennes. C'est à dire qu'elles ne prennent que deux valeurs 0 ou 1.

La logique floue a pour but de raisonner à partir de connaissances imparfaites qui opposent une résistance à la logique classique. Pour cela la logique floue se propose de remplacer les variables booléennes par des variables flous.

1.11.1.1 Historique:

Depuis longtemps l'homme cherche à maîtriser les incertitudes et les imperfections inhérentes à sa nature. La première réelle manifestation de la volonté de formaliser la prise en compte des connaissances incertaines fut le développement de la théorie des probabilités à partir du XVII siècle. Mais les probabilités ne peuvent maîtriser les incertitudes psychologiques et linguistiques. On a donc assisté aux développements des théories de probabilité subjective (dans les années 50) puis de l'évidence (dans les années 60).

Puis la Logique Floue est apparue en 1965 à Berkeley dans le laboratoire de Lotfi Zadeh [4] avec la théorie des sous-ensembles flous puis en 1978 avec la théorie des possibilités. Ces deux théories constituent aujourd'hui ce que l'on appelle Logique Floue.

La Logique Floue permet la formalisation des imprécisions dues à une connaissance globale d'un système très complexe et l'expression du comportement d'un système par des mots.

Elle permet donc la standardisation de la description d'un système et du traitement de données aussi bien numériques qu'exprimées symboliquement par des qualifications linguistiques **[9]**.

1.11.1.2 Sous-ensembles flous :

Un sous-ensemble floue F est défini sur un ensemble de valeur, le référentielle U. Il est caractérisé par une fonction d'appartenance **[9]**:

$$\mu : x \in U \rightarrow \mu(x) \in [0,1] \tag{2-5}$$

Qui quantifie le degré d'appartenance de chaque élément de U à F.

Exemple : Evaluation de la température de l'eau d'un récipient par les mots

Froide : F Tiède : T Chaude : C

En logique classique :

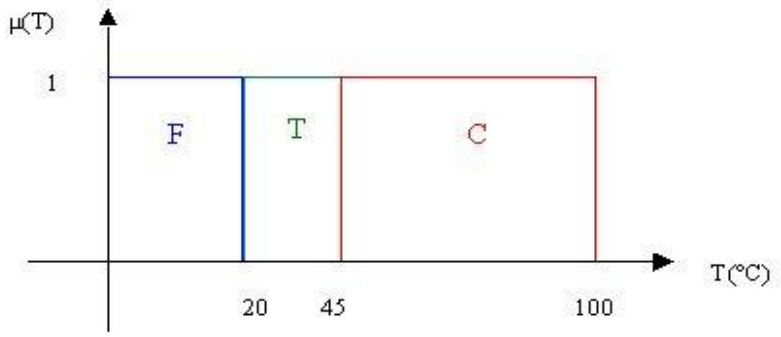


Figure 0.1 Logique classique

En logique floue :

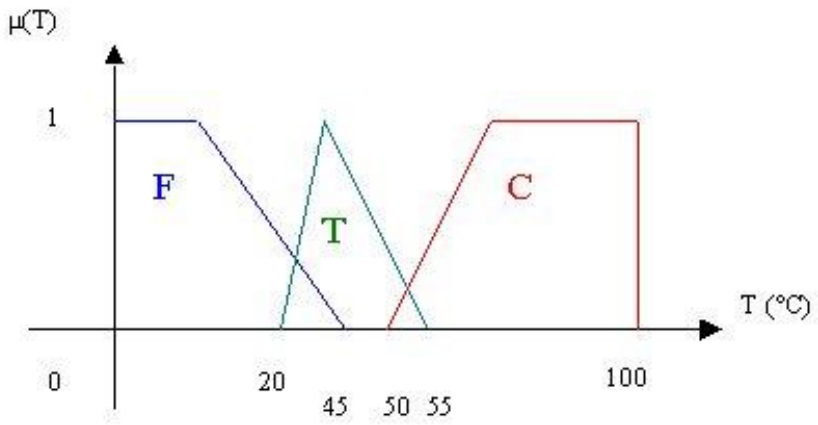


Figure 0.2 Logique floue

On voit que la logique classique ne peut utiliser que le 0 et le 1 ainsi l'eau est d'abord totalement froide puis tiède et enfin chaude. En dessous nous pouvons observer la représentation graphique de trois fonctions d'appartenance Froid, Tiède et Chaud. Ces fonctions nous permettent de superposer sur des plages de températures données les qualificatifs froid et tiède ainsi que tiède et chaud. On se rapproche donc du raisonnement humain.

1.11.1.3 Opérations de base sur les sous-ensembles flous :

La théorie mathématique sur les sous-ensembles flous définit de nombreuses opérations sur ces sous-ensembles et sur les fonctions d'appartenances qui rendent ces notions utilisables. Nous ne présentons ici que les opérations de base de cette théorie.

Si A et B sont deux sous-ensembles flous et μ (A) et μ (B) leur fonction d'appartenance, on définit :

- Le complémentaire de A, par la fonction d'appartenance : μ (A)=1-μ (A)

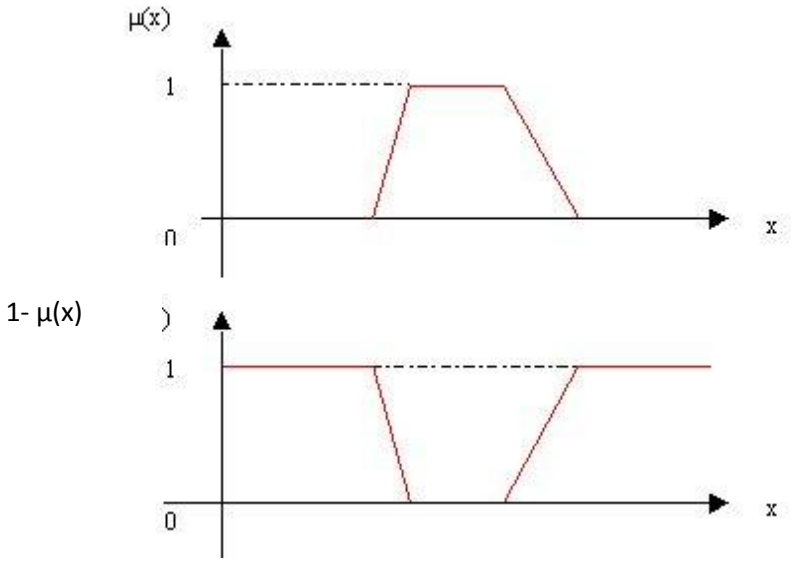


Figure 0.3 Fonction d'appartenances

- Le sous-ensemble A et B, par la fonction d'appartenance :

$$\mu (A \cap B)=\min (\mu(A), \mu(B)) \qquad (2\text{-}6)$$

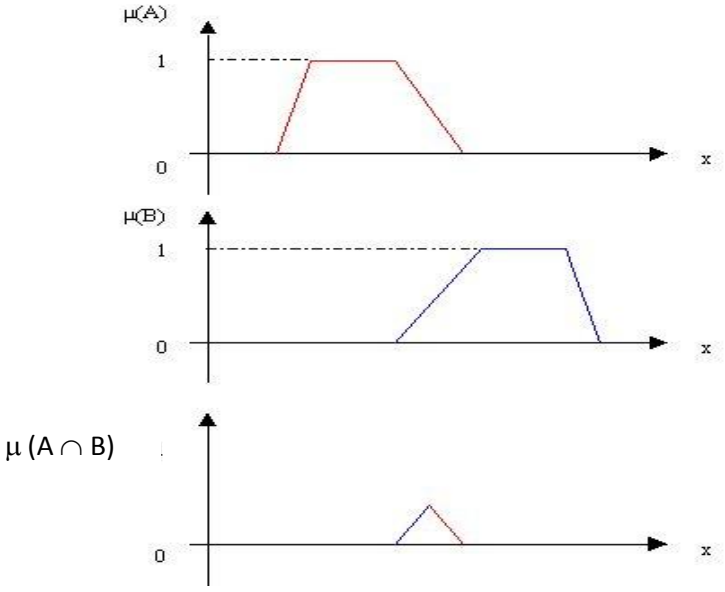


Figure 0.4 Intersection des fonctions d'appartenances

- Le sous-ensemble A ou B, A ∪ B, par la fonction d'appartenance :

$$\mu(A \cup B)=\max(\mu(A), \mu(B)) \qquad (2\text{-}7)$$

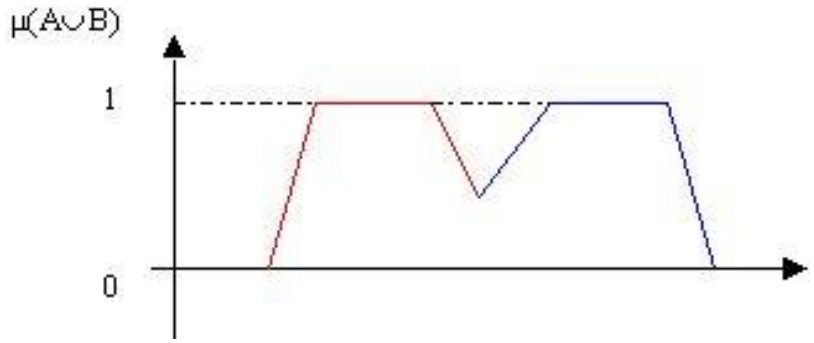


Figure 0.5 Union des fonctions d'appartenances

Ces définitions sont celles qui sont les plus communément utilisées mais parfois, pour certains cas, d'autres sont plus appropriées. Par exemple, l'intersection peut être définie par le produit des fonctions d'appartenance et l'union par la moyenne arithmétique des fonctions d'appartenance. Ces différentes techniques de calcul engendrent une énorme capacité d'adaptation des raisonnements flous.

1.11.2 Principes d'un arbre de décision flou :

La fouille de données consiste à extraire des connaissances à partir de données hétérogènes. Quand les données sont numériques et/ou symboliques, les arbres se décision constituent un bon outil. Ils permettent d'extraire des informations sous la forme de règles du type :

"Si... alors…".les arbres de décision flous constituent une extension des arbres de décision classiques du fait qu'ils permettent de contourner la contrainte de seuil rigide pendant le test au niveau des nœud **[9]**.

1.11.2.1 Construction des arbres de décision flous :

Les arbres de décision flous généralisent les arbres de décision classiques et sont beaucoup mieux adaptés pour le traitement de données numériques continues. Le parcours de la racine à une feuille constitue une règle floue. Lors de la présentation d'un exemple, différentes règles sont activées et un mécanisme d'inférence permet de composer ces règles pour obtenir la réponse de l'arbre.

Les arbres de décision ou de régression flous permettent de manipuler des variables continues sans introduire des seuils arbitraires de décision booléenne, où des tests en tout-ou-rien sont réalisés à chaque nœud. Si par exemple dans un cas de test de taux de glycémie chez un diabétique : 'le taux est inférieur à 1g, la réponse sera vrai pour les deux observations τ_0 = 0g et τ_1 = 0,999g ; la réponse sera faux pour l'observation τ_3 = 1g. Pour une différence minime et souvent non significative entre deux observations, les tests booléens peuvent ainsi conduire à des conclusions très différentes.

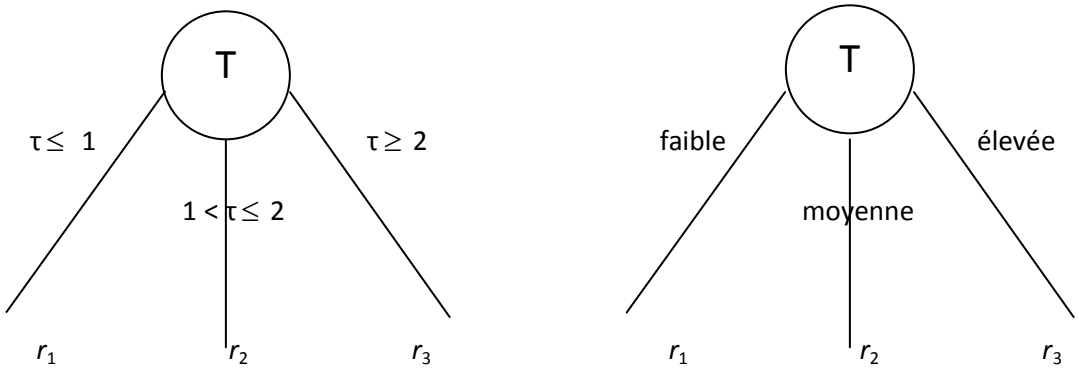
Il existe plusieurs implémentations pour les arbres de décision flous. Celle que nous avons choisie peut être considérée comme une extension d'ID3 (logiciel d'implémentation des arbres) :

Chaque nœud est associé à une variable dont le domaine de variation est découpé par une partition floue forte. Les tests réalisés à un nœud donnent alors les degrés d'appartenance aux différents sous-ensembles flous et les branches issues du nœud sont activées par ces degrés d'appartenance **[9]**.

En appelant r_i l'activation de la branche ***i***, on a (Figure 02.0.6) :

$$r_i \in \{0,1\} \text{ et } \sum_i r_i = 1 \quad \text{pour un arbre booléen} \qquad (2\text{-}8)$$

$$r_i \in [0,1] \text{ et } \sum_i r_i = 1 \quad \text{pour un arbre flou} \qquad (2\text{-}9)$$



τ : le taux de glycémie chez un diabétique

Figure 02.0.7 Arbre booléen (à gauche) et arbre flou

1.11.3 Caractéristiques principales d'un arbre de décision flou:

Dans un arbre de décision flou la branche forme la partie prémisse, tandis que la valeur associée à une feuille forme la conclusion [11]. Soit c_N cette conclusion et $\alpha_N(\mathbf{X_i})$ la valeur de la règle correspondante pour un vecteur d'entrée $X_i = (x_{i1},\ldots,x_{iM})$ donné. Chaque attribut, x_{ij}, ayant des modalités floues, avec $\mu_{A_j}(X_{i\,j})$ est le degré d'appartenance de $x_{i\,j}$ au sous-ensemble flou Aj.

On appelle aussi :

- $\alpha_N(X_i)$ le degré d'appartenance du vecteur $\mathbf{x_i}$ au nœud N calculé de la racine au nœud.
- $\alpha_N(X_i)$ est calculé à partir de tous les degrés d'appartenance le long de la branche concernée :

$$\alpha_N(xi) = \begin{cases} 1 \text{ si Nest la recine} \\ \alpha \mathrm{M(xi)} \wedge \mu \mathrm{Ai}(x_{ij}) \text{ sinon} \end{cases} \quad (2\text{-}10)$$

Si N est la racine, alors $\alpha_N(x_i)=1$ pour tout *i*. Si N est un nœud ou une feuille, alors $\alpha_N(x_i)$ est le produit de tous les degrés d'appartenance sur la branche, de la racine à N

Avec les notations de la (Figure 0.8) ou M est le parent de N ,X_{ij} la variable attachée à M et N et A est un sous-ensemble flou sur le domaine de X_{ij}.

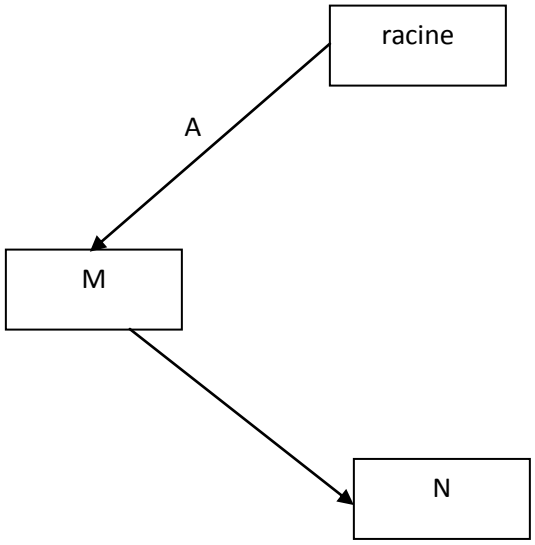


Figure 0.9 Appartenance au nœud

La sortie inférée par un ADDF est :

$$\text{ADDF}(x) = \frac{\sum_F \alpha_F(x) c_F}{\sum_F \alpha_F(x)} \qquad (2\text{-}11)$$

La conclusion associée à une feuille, C_F est calculée par :

$$C_F = \frac{\sum_{j\in J} \alpha_F(x_j) y_i}{\sum_{i\in J}^{N} \alpha_F(x_j)} \qquad (2\text{-}12)$$

J : ensemble des couples (xj,yj) pour lesquels la valeur de vérité de la feuille est supérieure à un seuil de donné.

1.11.4 Induction d'un arbre de décision flou :

On parle d'induction car la construction permet de passer du particulier (données de l'ensemble d'apprentissage) au général (les règles extraites des données). Cette construction successive sur l'ensemble d'apprentissage en sous-ensembles d'homogénéité croissante. Considérons le problème général suivant. Soit $f(x_1,\ldots,x_n)$ $y=f(x_1,\ldots,x_n)$ une relation d'entrée/sortie inconnue, les entrées x_1 à x_n étant des variables explicatives potentielles et y la sortie. Parmi les entrées, certaines sont importantes, d'autres redondantes, d'autres enfin inutiles. L'induction d'un ADDF consiste à :

- Hiérarchiser les variables d'entrée en fonction de leur importance.
- Évaluer l'utilité de prendre en compte ou non certaines entrées, soit en ligne, soit hors ligne par élagage.

Il faut, pour cela, un ensemble d'apprentissage,

$$E = \{ (x_i, y_i) ; x_i = (x_{i1},\ldots,x_{iM}), y_i \in R \text{ , pour } i = 1 \text{ à } P \} , \qquad (2\text{-}13)$$

où les x_{ij} , pour j = 1 à M, sont des variables linguistiques possédant m_j fonctions d'appartenance, $(A_{jk})^{mk}_{j=1}$.

Les fonctions d'appartenance forment des partitions floues triangulaires sur les domaines d'entrée. Sans perte de généralité, la T-norme utilisée est le produit :

$$\mathrm{ET}(x, y) = x * y. \qquad (2\text{-}14)$$

ces choix apportent la propriété suivante :

$$\sum_F \alpha_F(x) \equiv 1 \qquad (2\text{-}15)$$

Et l'équation (2-11) devient :

$$\text{ADDF(x)} = \sum_{F} \alpha_F(x) c_F \qquad (2\text{-}16)$$

La construction automatique d'un arbre nécessite des mesures comme l'entropie et le gain d'information.

Dans un problème de classification, dans le cas idéal, les vecteurs d'apprentissage associés à un nœud terminal (une feuille) appartiennent à la même classe.

On dit alors que la feuille est "pure". Ce n'est évidemment pas toujours possible et l'objectif de l'induction vise à créer des feuilles avec un degré de mélange minimum.

Le partitionnement est réalisé à chaque nœud par des tests portant sur une variable : il faut donc choisir le meilleur test, l'entropie permet de faire le bon choix de la classe.

- la notion de représentation de la classe k au nœud N pour une modalité A_j de la variable traitée, joue un rôle central. Elle est définie par :

$$r\,(k, j, N) = \sum_{i=1}^{P} \mu_k(x_i) \wedge \mu_{Aj}(x_{ij}) \wedge \alpha_N(x_i) \qquad (2\text{-}17)$$

On en déduit les paramètres P_k et w_j utilisés pour le calcul du gain :

$$P_k = \frac{\sum_j r(k, j, N)}{\sum_k \sum_j r(k, j, N)} = \frac{\sum_j r(k, j, N)}{\sum_{i=1}^{P} \alpha_N(X_i)} \qquad (2\text{-}18)$$

$$w_j = \frac{\sum_k r(k, j, N)}{\sum_k \sum_j r(k, j, N)} = \frac{\sum_k r(k, j, N)}{\sum_{i=1}^{P} \alpha_N(x_i)} \qquad (2\text{-}19)$$

Le gain d'information apporté par un attribut X au nœud N est :

$$G(X, N)=I(N)-Info(X,N) \quad (2\text{-}20)$$

Avec

$I(N)=-\sum_k P_k \log P_k$ Information au nœud N (entropie) (2-21)

$Info(X, N)=\sum_j w_j I(X_j)$ Information apportée par X au nœud N (2-22)

1.11.5 Optimisation de l'arbre de décision flou:

Le défaut majeur des arbres de décision flous est qu'il faut fixer à priori le nombre de fonctions d'appartenances par entrée et les placer plus ou moins empiriquement sur chaque domaine.

Ce qui diminue la performance de l'arbre car ils interviennent dans le calcul de la valeur des différents paramètres, tel que le gain d'information. La méthode proposée est d'augmenter le nombre de fonctions d'appartenance selon une heuristique et trouver le nouveau placement optimal algorithme de Solis et Wetts [12].

1.12 Résumé :

Nous avons présenté dans ce chapitre les principes des arbres de décision classiques ensuite la technique des arbres de décision flous.

Nous avons défini quelques notions essentielles comme l'entropie et le gain d'information qui jouent un rôle très important dans la construction de l'arbre de décision flou. Dans le chapitre suivant nous allons implémenter un classifieur des arythmies cardiaques en se basant sur le principe de cette technique.