Université Pierre et Marie Curie – Paris VI

HABILITATION À DIRIGER DES RECHERCHES

spécialité **Informatique**

présentée par

Christophe MARSALA

Sujet :

Apprentissage artificiel et raisonnement flou

soutenue le mardi 30 novembre 2010

devant le jury composé de

Mme Sylvie Galichet	rapporteur
M. Eyke Hüllermeier	rapporteur
M. Louis Wehenkel	rapporteur
Mme Bernadette Bouchon-Meunier	
Mme Giulianella Coletti	
M. Patrick Gallinari	
M. Mohammed Ramdani	

Résumé

Ce mémoire présente les travaux de recherches que j'ai menés, seul, en collaborant avec d'autres collègues ou en co-encadrant des thèses, depuis l'obtention de ma thèse d'université, en 1998. Le cadre principal de mes travaux est l'étude et la proposition de méthodes pour étendre des algorithmes d'apprentissage artificiel pour les doter d'une meilleure capacité de prise en compte des données imparfaites (numériques, imprécises, incertaines, ou incomplètes), et améliorer leur mise en œuvre dans un raisonnement flou.

Des premiers travaux de recherche m'ont conduit à étudier les mesures de discrimination, leurs propriétés et leur rôle pour la sélection et l'ordonnancement d'attributs flous dans les algorithmes d'apprentissage artificiel.

D'autres travaux de recherche m'ont amené à étudier d'autres apports possibles de l'apprentissage artificiel pour l'amélioration des mécanismes de raisonnement flou. Dans ce rapport, une étude est présentée sur les méthodes de construction de règles floues et sur l'utilisation de bases de règles incomplètes par un raisonnement flou par interpolation dans un cadre multi-prémisses.

D'autre part, un modèle de construction et d'utilisation de forêts d'arbres de décision flous est proposé pour pallier les déséquilibres des distributions des classes dans certains problèmes d'apprentissage.

Diverses applications sur des données complexes, dans le domaine des bases de données géographiques ou multidimensionnelles, dans le domaine médical, ou en video mining, qui m'ont permis de réaliser des études réelles de data mining, sont aussi détaillées.

Summary

In this report, a description of the researches I have conducted since the defense of my PhD thesis, in 1998, is done. The aim of these researches is the study and the proposal of methods to extend machine learning algorithms to enable them to take into account imperfect data (numerical, imprecise, fuzzy, uncertain, or incomplete) and to enhance their use in fuzzy reasoning.

First of all, a study on measures of discrimination is done that highlights their main properties and their role to select and to rank fuzzy attributes in a machine learning algorithm.

Secondly, a study of other contributions of machine learning to enhance fuzzy reasoning tools is presented. A comparison of two approaches of construction of a fuzzy rule base is done and a new method of interpolative reasoning with multi-premises is proposed. A new model to construct and to use a forest of fuzzy decision trees is proposed to handle class-unbalanced data sets with a machine learning algorithm.

Several real-world applications are described in the domain of geographical databases or multidimensional databases, in the medical domain, and in video mining. These applications enable us to tackle real-world data mining problems.

Table des matières

Introduction

Les informations du monde réel que nous avons à traiter en permanence sont essentiellement des données imparfaites, imprécises, incertaines, incomplètes. La mise en œuvre d'un système automatique de raisonnement se doit donc de prendre en compte et de gérer au mieux ce type d'informations.

La théorie des sous-ensembles flous de Zadeh a fait la preuve, depuis maintenant 45 ans, de sa validité et de son efficacité pour la prise en compte de ce type d'informations, à travers les méthodes théoriques et les applications pratiques qui ont été réalisées dans différents domaines. Ainsi, dans le domaine de l'apprentissage artificiel, la théorie des sous-ensembles flous a permis l'extension d'algorithmes d'apprentissage classiques ou la proposition de nouveaux algorithmes originaux pour une meilleure prise en compte des données issues du monde réel.

C'est dans ce cadre que se situent les recherches que j'ai effectuées depuis ma thèse de doctorat, soutenue en janvier 1998. Dans la continuation logique de ma thèse, je me suis attaché à étudier les apports que l'on peut proposer pour améliorer la prise en compte de données réelles par un algorithme d'apprentissage artificiel et leur mise en œuvre dans un raisonnement flou.

Dans le premier chapitre de ce mémoire, je présente les recherches que j'ai menées, seul ou conjointement avec d'autres collègues, sur les méthodes de sélection d'attributs flous en apprentissage artificiel. Dans ces travaux, j'ai étendu l'approche que j'avais proposée durant ma thèse pour obtenir un modèle d'étude des mesures plus général qui puisse s'appliquer à des mesures de discrimination floues. L'idée directrice de ces recherches est de répondre à la question qui se pose lorsque l'on souhaite étendre un algorithme d'apprentissage classique pour la prise en compte de données réelles et offrir un cadre d'étude des mesures qui puissent aider à étudier, valider et choisir une mesure de discrimination selon les propriétés qu'elle possède et l'application dans laquelle on souhaite l'utiliser. Ce domaine de recherche a représenté un point important de mes recherches car il a aussi été un des thèmes de deux des thèses que j'ai co-encadrées.

Dans le second chapitre, je présente les travaux que j'ai réalisés pour la construction de forêts d'arbres de décision flous. Ces travaux ont émergé de l'application de data mining dans le domaine de l'extraction de descripteurs sémantiques à partir de vidéos auquel j'ai participé. Dans cette application, il a été nécessaire de proposer un mécanisme d'apprentissage par arbres de décision flous plus complexe pour arriver à pallier les imperfections des données vidéos que l'on avait à traiter (bases avec des distributions de classes déséquilibrées en particulier). L'utilisation d'ensembles de classifieurs flous (des arbres de décision flous en l'occurrence), peu répandue encore à l'époque, a permis d'obtenir une meilleure robustesse pour la prise en compte de ce type de bases et d'offrir une augmentation notable de la capacité de généralisation des règles floues générées par l'apprentissage d'arbres.

Dans le troisième chapitre, je présente les travaux que j'ai réalisés avec d'autres collègues sur les modèles d'apprentissage et de raisonnement flous. Tout d'abord, je présente une étude de méthodes d'apprentissage de règles floues qui a pour cadre une comparaison de deux ap-

proches de génération de règles floues afin d'en faire ressortir les similitudes et d'en proposer des améliorations. Je présente ensuite une étude sur le raisonnement par interpolation qui permet la mise en œuvre de bases de règles floues incomplètes comme celles qui sont susceptibles d'être générées par apprentissage artificiel.

Le quatrième chapitre fait un tour d'horizon de diverses applications de data mining auxquelles j'ai participé et qui m'ont permis de mettre en œuvre des méthodes d'apprentissage artificiel et de raisonnement flou dans des cas d'étude concrets. Ces applications m'ont d'ailleurs souvent permis d'alimenter les travaux plus théoriques qui sont présentés dans les chapitres précédents, en permettant de faire ressortir des difficultés ou des nécessités dans la prise en compte de données réelles.

Dans le cinquième chapitre, une présentation des thèses que j'ai co-encadrées est faite. Dans ces travaux, les thèmes abordés ont toujours pour point commun le domaine de l'apprentissage artificiel et la prise en compte de données réelles, imparfaites, et imprécises. Avec le concours des doctorants qui les ont réalisées, j'ai ainsi participé à des travaux qui m'ont souvent permis de mettre en œuvre des recherches dans un cadre applicatif concret.

Finalement, je présente une conclusion et quelques perspectives de travaux que j'envisage pour la suite de mes recherches.

Chapitre 1

Sélection d'attributs flous en apprentissage

1.1 Introduction

L'apprentissage inductif repose sur l'idée de généralisation à partir d'une population donnée, une *base d'apprentissage*, $\mathcal{E} = \{e_1, ..., e_n\}$ contenant des individus caractéristiques d'une certaine classe C, afin d'obtenir une loi générale régissant l'occurrence des valeurs de la classe.

Chaque e_i de $\mathcal{E}$ est un *exemple* de la classe à apprendre, donné sous la forme d'une *description* associée à une valeur de C. En ce qui nous concerne, une description est un N-uplet de couples attribut-valeur $(A_j, e_i(A_j))$ où un attribut A_j est une caractéristique mesurable et observable de la classe, possédant la valeur $e_i(A_j)$ pour l'exemple e_i. Dans la base d'apprentissage, la valeur $e_i(C)$ de C, qui est la valeur de la classe associée, est connue. Par abus de langage, on appelle simplement *classe de l'exemple* e_i la valeur $e_i(C)$. Les attributs et la classe définissent, par leurs valeurs, des sous-ensembles flous sur $\mathcal{E}$ qui sont donc définis par leurs fonctions d'appartenance μ de $\mathcal{E}$, à valeurs dans $[0, 1]$.

De plus, on considère généralement que chaque exemple e de $\mathcal{E}$ est associé à une probabilité d'occurrence $p(e)$. En fait, les probabilités des exemples sont souvent estimées par leur fréquence dans la base d'apprentissage $\mathcal{E}$. Chaque exemple e est donc alors associé à la probabilité $p(e) = \frac{1}{|\mathcal{E}|}$.

Beaucoup d'algorithmes d'apprentissage inductif reposent sur l'évaluation du *pouvoir de discrimination* des attributs relativement à la classe. Autrement dit, ils recherchent à évaluer comment les valeurs de la classe peuvent être prédites par les valeurs d'un attribut. L'idée est de pouvoir ordonner les attributs afin de faire ressortir ceux dont les valeurs sont les plus corrélées aux valeurs de la classe. Ainsi, l'ordonnancement des attributs s'effectue en fonction de leur pouvoir de discrimination relativement à la classe. On souhaite au final obtenir l'attribut A_j qui est le plus fortement corrélé à la classe C, c'est-à-dire celui avec lequel les valeurs de la classe pourraient être le plus facilement déterminées à partir de ses propres valeurs.

Pour cela, une mesure est utilisée afin d'évaluer le pouvoir de discrimination d'un attribut relativement à la classe. Cette *mesure de discrimination* se doit de posséder un certain nombre de propriétés afin de garantir que l'ordonnancement des attributs ait un sens réel et soit le reflet correct de leur pouvoir de discrimination.

Définition 1 (mesure de discrimination) *Une* mesure de discrimination *est une fonction à valeur réelle qui évalue le pouvoir de discrimination d'un attribut vis-à-vis de la classe, et dont*

la valeur est :

- *minimale quand toute valeur de l'attribut permet de déterminer sans ambiguïté la valeur de la classe,*
- *maximale quand la connaissance de la valeur de l'attribut ne lève aucune ambiguïté sur la valeur de la classe.*

Dans ce chapitre, nous proposons un *modèle hiérarchique de fonctions* pour caractériser les propriétés requises pour une fonction utilisée comme une mesure de discrimination. Ce modèle est une extension dans le cadre de la théorie des sous-ensembles flous du modèle que nous avons présenté dans [79] et dans lequel les propriétés principales requises pour ce type de mesures ont été mises en évidence. Une extension de ce modèle a déjà été proposée par [36, 32] pour la prise en compte de valeurs floues dans la description des exemples, mais elle est conçue sur des hypothèses très restrictives pour le choix de la mesure de l'inclusion des sous-ensembles flous, ainsi que du choix de la t-norme utilisée pour définir leur intersection. La nouvelle extension du modèle hiérarchique que nous proposons ici est fondée sur de nouvelles hypothèses moins contraignantes pour le choix des opérateurs d'inclusion et d'intersection. Une synthèse des différences entre les modèles hiérarchiques existants est présentée dans la Table 1.1.

	Modèle [61]	Modèle [32]	Nouveau modèle
Valeur d'attribut	Floue	Floue	Floue
Valeur de classe	Précise	Floue	Floue
Description d'un exemple d'apprentissage	Précise	Floue	Floue
Inclusion $A \subseteq B$ iff $\forall x \in X$	Classique	$\mu_{A\cap B}(x) = \mu_A(x)$	$\mu_{A\cap B}(x) \leq \mu_A(x)$
Intersection $A \cap B$	Classique	Produit	Toute t-norme

Table 1.1 – Comparaison des modèles hiérarchiques

Ce chapitre se décompose comme suit. Un rappel sur les mesures de discrimination pour la construction d'arbres de décision est d'abord fait. Ensuite, notre extension dans le cadre flou du modèle hiérarchique de fonctions est décrite. Puis, une étude sur les mesures de discrimination les plus utilisées pour construire des arbres de décision flous est faite, afin de voir leur adéquation avec le modèle proposé. Finalement, nous proposons une deuxième étude pour mieux cerner les différences, en termes de résultats, de l'utilisation de mesures pour la construction d'arbres de décision.

On ne répétera pas ici tout ce qui a déjà été dit dans [79] mais on fera un simple rappel des points essentiels pour situer les autres éléments du reste du chapitre.

Les notations utilisées dans ce chapitre, ainsi qu'un rappel sur les probabilités d'événements flous sont donnés en Annexe 5.7.

1.2 Mesures de sélection d'attributs flous

Dans cette partie, un rappel est fait sur les mesures les plus couramment utilisées pour ordonner des attributs flous, en particulier lors de la construction d'arbres de décision flous.

Soit un univers $\mathcal{X} = \{x_1, ..., x_n\}$ pour lequel chaque élément x_i de $\mathcal{X}$ est associé à une probabilité (classique) d'occurrence $p(x_i)$, telle que $\sum_{i=1}^{n} p(x_i) = 1$. Dans ce qui suit, on considère $\mathcal{U} = \{U_1, U_2, ..., U_m\}$ et $\mathcal{V} = \{V_1, V_2, ..., V_l\}$, deux ensembles de sous-ensembles, éventuellement

flous, de $\mathcal{X}$, tels que $0 < \sum_{i=1}^{m} p^*(U_i) \leq 1$, et $0 < \sum_{j=1}^{l} p^*(V_j) \leq 1$. De plus, on suppose que pour tout $U \in \mathcal{U}$, $p^*(U) > 0$.

1.2.1 Entropie d'événements flous

L'*entropie d'événements flous*[1] est une extension de l'entropie de Shannon pour la prise en compte d'événements flous, obtenue en remplaçant les probabilités classiques par des probabilités d'événements flous.

L'entropie floue H^* de l'ensemble $\mathcal{U}$ est définie[2] par :

$$H^*(\mathcal{U}) \quad = \quad -\sum_{i=1}^{m} p^*(U_i) \log(p^*(U_i)) \tag{1.1}$$

L'*entropie conditionnelle d'événements flous* est, elle aussi, définie par extension de l'entropie de Shannon. L'entropie conditionnelle de $\mathcal{V}$ sachant $\mathcal{U}$ est définie par :

$$H_E^*(\mathcal{V}|\mathcal{U}) \quad = \quad -\sum_{i=1}^{m} p^*(U_i) \sum_{j=1}^{l} p^*(V_j|U_i) \log(p^*(V_j|U_i))$$

On remarque ici que, pour cette entropie conditionnelle, on utilise la notation conditionnelle pour les événements mais, en fait, H_E^* est une fonction à 2 variables. Dans ce qui suit, sauf indication contraire, pour toute fonction f, on notera indifféremment $f(\mathcal{V}|\mathcal{U})$ ou $f(\mathcal{U}, \mathcal{V})$.

On peut réécrire $H_E(\mathcal{V}|\mathcal{U})$ ainsi :

$$H_E(\mathcal{V}|\mathcal{U}) = \sum_{i=1}^{m} p^*(U_i) \cdot G_E(\mathcal{V}|U_i),$$

avec

$$G_E(\mathcal{V}|U_i) \quad = \quad -\sum_{j=1}^{l} p^*(V_j|U_i) \log p^*(V_j|U_i). \tag{1.2}$$

Cette forme sera intéressante par la suite.

1.2.2 Index flou de diversité de Gini

L'*index de diversité* a été introduit par Gini en inférence statistique, et il a été utilisé, par exemple, par Simpson en biologie [58]. Il a aussi été utilisé pour la construction d'arbres de décision [29].

Cet index peut s'étendre pour la prise en compte de valeurs floues de la même façon que l'entropie de Shannon. L'*index flou de diversité de Gini* $H_G(\mathcal{V}|\mathcal{U})$ est défini par :

$$H_G(\mathcal{V}|\mathcal{U}) = \sum_{i=1}^{m} p^*(U_i) \cdot G_G(\mathcal{V}|U_i), \tag{1.3}$$

[1] Aussi appelée souvent plus simplement *entropie floue*.

[2] Classiquement [3], une entropie est définie pour une distribution de probabilité donnée. Nous préférons une notation plus ensembliste (mais équivalente) pour notre contexte d'apprentissage inductif.

avec $G_G(\mathcal{V}|U_i) = 1 - \sum_{j=1}^{l} p^*(V_j|U_i)^2$.

Sous cette forme, on peut remarquer de suite (voir équation (1.2)) que $H_E(\mathcal{V}|\mathcal{U})$ et $H_G(\mathcal{V}|\mathcal{U})$ sont construites par la même agrégation d'une fonction $G(\mathcal{V}|U_i)$.

1.2.3 Mesure d'ambiguïté

La *mesure d'ambiguïté* a été introduite par [112] pour la construction d'arbres de décision flous. C'est une des rares mesures utilisées pour construire des arbres de décision flous qui ne soit pas une extension d'une mesure classique.

La mesure de l'ambiguïté $H_Y(\mathcal{V}|\mathcal{U})$ de $\mathcal{V}$ relativement à $\mathcal{U}$ est définie par :

$$H_Y(\mathcal{V}|\mathcal{U}) = \sum_{i=1}^{m} w(U_i) \cdot G_Y(\mathcal{V}|U_i) \tag{1.4}$$

avec

$$w(U_i) = \frac{M(U_i)}{\sum_{k=1}^{m} M(U_k)}$$

et avec M la mesure de cardinalité floue (sigma-count de [113]) :

$$M(U) = \sum_{x \in \mathcal{X}} \mu_U(x)$$

Si $\mathcal{U}$ est une partition floue de $\mathcal{X}$, on a, pour tout $x \in \mathcal{X}$, $\sum_{k=1}^{m} \mu_{U_k}(x) = 1$ et aussi $\sum_{k=1}^{m} M(U_k) = n$.

On a alors $w(U_i) = \frac{1}{n} \sum_{x \in \mathcal{X}} \mu_{U_i}(x)$, ce qui permet de retrouver la probabilité d'événements flous de Zadeh [113] si on considère que les éléments x de $\mathcal{X}$ sont équiprobables, soit donc $w(U_i) = p^*(U_i)$ car on a alors

$$w(U_i) = \sum_{x \in \mathcal{X}} \mu_{U_i}(x) p(x)$$

L'équation (1.4) peut donc s'écrire :

$$H_Y(\mathcal{V}|\mathcal{U}) = \sum_{i=1}^{m} p^*(U_i) \cdot G_Y(\mathcal{V}|U_i)$$

ce qui nous permet de mettre en évidence le fait que $H_E(\mathcal{V}|\mathcal{U})$, $H_G(\mathcal{V}|\mathcal{U})$ et $H_Y(\mathcal{V}|\mathcal{U})$ sont construites par la même agrégation d'une fonction $G(\mathcal{V}|U_i)$.

Pour revenir à la mesure d'ambiguïté, on a :

$$G_Y(\mathcal{V}|U_i) = g(\Pi(\mathcal{V}|U_i))$$

où g est une mesure de *non-specificité*, et $\Pi(\mathcal{V}|U_i) = \{\pi(V_j|U_i),\ j = 1, .., l\}$ est la distribution de possibilité de $\mathcal{V}$ conditionnellement à U_i. $\Pi(\mathcal{V}|U_i)$ est déterminée comme suit. Chaque $\pi(V_j|U_i)$ est définie à partir d'une mesure *d'inclusion floue* S :

$$\pi(V_j|U_i) = \frac{S(U_i, V_j)}{\max_{k=1,..,l} S(U_i, V_k)},$$

où l'inclusion floue S de la valeur U_i dans la valeur V_j est définie par :

$$S(U_i, V_j) = \frac{\sum_{x \in \mathcal{X}} \mu_{U_i \cap V_j}(x)}{\sum_{x \in \mathcal{X}} \mu_{U_i}(x)}$$

Toujours en considérant que les éléments de $\mathcal{X}$ sont équiprobables, il est alors aisé de voir que l'on a :

$$\begin{aligned} S(U_i, V_j) &= \frac{p^*(V_j \cap U_i)}{p^*(U_i)} \\ &= p^*(V_j|U_i) \end{aligned}$$

Pour U_i donné, on note $V_+^i = \text{argmax}_{V_k, k=1,...,l} S(U_i, V_k)$. Avec la définition de la mesure d'inclusion S que l'on a, on remarque de suite que V_+^i est donc la valeur de $\mathcal{V}$ qui possède le plus d'éléments communs avec U_i.

On peut alors écrire :

$$\pi(V_k|U_i) = \frac{\frac{p^*(V_k \cap U_i)}{p^*(U_i)}}{\frac{p^*(V_+^i \cap U_i)}{p^*(U_i)}} = \frac{p^*(V_k \cap U_i)}{p^*(V_+^i \cap U_i)}.$$

Dans ce qui suit, pour alléger les notations, on note $\pi(V_k|U_i)$ simplement π_k.

La mesure de non-spécificité g de la distribution de possibilité $\Pi = \{\pi_1, ..., \pi_m\}$ est alors définie par :

$$g(\Pi) = \sum_{j=1}^{l} (\pi_j^* - \pi_{j+1}^*) \log j \tag{1.5}$$

où $\Pi^* = \{\pi_1^*, ..., \pi_{l+1}^*\}$ avec $\pi_{l+1}^* = 0$ et $\{\pi_1^*, ..., \pi_l^*\}$ est obtenue par une permutation ρ de $\{\pi_1, ..., \pi_l\}$ de telle sorte que, pour tout $j = 1, .., l$, $\pi_j^* \geq \pi_{j+1}^*$, avec $\pi_j^* = \pi_{\rho(j)}$.

Par définition, on a $\pi_1^* = \frac{p^*(V_+^i \cap U_i)}{p^*(V_+^i \cap U_i)} = 1$ et :

$$\pi_j^* = \frac{p^*(V_{\rho(j)} \cap U_i)}{p^*(V_+^i \cap U_i)}.$$

De plus, $V_{\rho(1)} = V_+^i$, et par conséquent, l'équation (1.5) se réécrit :

$$\begin{aligned} g(\Pi) &= \sum_{j=1}^{l} \pi_j^* \log j - \pi_{j+1}^* \log j \\ &= \sum_{j=2}^{l} \pi_j^* (\log j - \log(j-1)) \end{aligned}$$

car $\pi_1^* \log(1) = 0$ et $\pi_{l+1}^* \log(l) = 0$.

On trouve alors

$$g(\Pi) = \sum_{j=2}^{l} \frac{p^*(V_{\rho(j)} \cap U_i)}{p^*(V_+^i \cap U_i)} \log(1 + \frac{1}{j-1})$$

Ainsi, on peut en conclure que l'équation (1.5) se réécrit :

$$G_Y(\mathcal{V}|U_i) = \sum_{j=2}^{l} \frac{p^*(V_{\rho(j)} \cap U_i)}{p^*(V_+^i \cap U_i)} \log(1 + \frac{1}{j-1})$$

On remarque ici que, dans l'expression de la mesure d'ambiguïté, l'ordre des valeurs de $\mathcal{V}$ est important, ce qui n'est pas le cas pour l'entropie floue ou l'index flou de Gini.

1.2.4 D'autres mesures

Il existe un grand nombre de mesures pour sélectionner des attributs en fonction de leur pouvoir pour discriminer des valeurs d'une classe. Nous n'en avons présenté que trois principales ici, qui sont surtout utilisées pour la construction des arbres de décision flous. Notre but est avant tout de présenter les mécanismes d'étude de telles mesures et non d'en faire une étude exhaustive.

Nous renvoyons à nos travaux conjoints avec Bernadette Bouchon-Meunier et Thanh Ha Dang, dans lesquels nous avons étudié d'autres mesures comme l'entropie généralisée de Daroczy, ou celle de Rényi.

1.3 Choisir une mesure de discrimination

Dans cette partie nous présentons un modèle hiérarchique de fonctions qui est une version étendue de celui que nous avions introduit dans [79].

La question principale qui se pose lorsque l'on a besoin de sélectionner des attributs concerne le choix de la mesure de discrimination adéquate, susceptible de rendre compte au mieux du pouvoir de discrimination d'un attribut vis-à-vis de la classe. N'importe quelle mesure ne peut être utilisée dans un tel but : il est nécessaire qu'elle puisse rendre compte de propriétés rendant cohérente son utilisation dans un tel processus. Dans le cadre classique, en théorie de l'information, l'entropie de Shannon, et les mesures entropiques, ont été largement étudiées et leurs propriétés sont bien connues (voir, par exemple, [3]).

En présence de données floues, un tel cadre d'étude n'existe pas pour les mesures floues de sélection d'attribut, et plus particulièrement pour l'utilisation de telles mesures dans un processus d'apprentissage. L'entropie floue, par exemple, est utilisée "naturellement", par extension de l'entropie de Shannon à l'aide des probabilités d'événements flous, mais sans, que pour autant, on puisse certifier que ses propriétés restent compatibles avec la mesure de la discrimination d'un attribut à valeurs floues relativement à une classe.

C'est pourquoi nous proposons ici un modèle permettant de valider le choix d'une mesure comme *bonne* mesure de discrimination. L'étude que nous proposons permet d'une part de mettre à jour la correction de l'utilisation d'une telle mesure floue pour construire des arbres, mais aussi, d'autre part, de mettre en évidence l'inadéquation de l'entropie floue pour un tel processus selon les choix d'opérateurs qui sont fait pour la définir.

Dans un tel processus, une bonne mesure se doit de posséder des propriétés qui lui confèrent une légitimité dans son utilisation pour évaluer le pouvoir discriminant d'un attribut vis-à-vis d'une classe. C'est donc en partant de ce que l'on souhaite mesurer que nous allons proposer une caractérisation des propriétés requises pour une bonne mesure de discrimination.

1.3.1 Comparaison d'ensembles d'exemples

Notre modèle repose sur une vision ensembliste du problème de la mesure du pouvoir de discrimination d'un attribut pour une classe. L'ensemble des exemples d'apprentissage $\mathcal{E}$ est un univers discret d'éléments sur lequel il est donc possible de définir des sous-ensembles flous.

Ainsi, une valeur v_{ji} d'un attribut A_j définit alors un sous-ensemble flou $\mathcal{E}_{v_{ji}}$ des exemples de $\mathcal{E}$. Ce sous-ensemble flou $\mathcal{E}_{v_{ji}}$ est défini par sa fonction d'appartenance μ_{ji} qui associe à chaque exemple de $\mathcal{E}$ le degré avec lequel il possède cette valeur v_{ji}. Si v_{ji} est une valeur symbolique, ou précise, μ_{ji} vaut donc soit 1, si l'exemple possède la v_{ji} pour cet attribut A_j, soit 0.

De la même façon, une valeur c_k de la classe C définit aussi un sous-ensemble flou $\mathcal{E}_{c_k}$ des exemples de $\mathcal{E}$. Ce sous-ensemble flou est défini par sa fonction d'appartenance μ_{c_k} qui associe à chaque exemple de $\mathcal{E}$ le degré avec lequel il possède cette valeur de classe.

De cette manière, A_j et C peuvent donc être vus chacun comme un ensemble de sous-ensembles flous sur l'ensemble des exemples, ils représentent tous les deux une partition particulière de l'ensemble $\mathcal{E}$.

Avec ce point de vue, le problème du choix d'un attribut parmi un ensemble d'attributs peut alors se voir de la façon suivante : *soit C une partition de $\mathcal{E}$ dite "de référence", et soit $\mathcal{A} = \{A_1, ..., A_K\}$ un ensemble de K partitions de $\mathcal{E}$; trouver la partition A_M de $\mathcal{A}$ qui soit la plus proche (ressemblante, similaire, adéquate, corrélée) de la partition de référence C.*

Cette proximité entre C et chaque A_j doit alors se mesurer en fonction des proximités existantes entre les sous-ensembles flous qui composent C et ceux qui composent A_j. En effet, c'est ici la partition qui crée la discrimination. Connaissant une valeur de la partition A_j, on souhaite pouvoir déterminer de façon la plus certaine possible, la valeur de la partition C correspondante.

Ainsi, nous proposons de décomposer la mesure du pouvoir de discrimination d'un attribut A_j pour C comme une agrégation des mesures du pouvoir de discrimination de chaque valeurs de A_j relativement à chaque valeur de C.

1.3.2 Modèle hiérarchique de fonctions

Le modèle hiérarchique de fonctions que nous proposons est défini à partir de trois types de fonctions à valeurs dans $\mathbb{R}$. Les notations utilisées dans cette partie sont rappelées en Annexe 5.7.

Définition 2 ($\mathcal{F}$-fonction) *Une* $\mathcal{F}$-fonction *est une fonction $F : I\!P[\mathcal{S}] \times I\!P[\mathcal{S}] \longrightarrow I\!R^+$ telle que, pour tout $(U, V) \in I\!P[\mathcal{S}] \times I\!P[\mathcal{S}]$, $U \neq \emptyset$ et V normalisé :*

i) $F(U, V) = 0$ *quand U est inclus dans le noyau de V (c'est-à-dire quand $U \subseteq {}^1V$).*

ii) $F(U, V)$ *est* **maximum** *quand $U \cap V = \emptyset$.*

iii) $F(U, V)$ *est* **strictement décroissante** *avec $U \cap V$.*
*(*i.e. *$U_1 \cap V \subset U_2 \cap V$ implique $F(U_1, V) > F(U_2, V)$, et $U \cap V_1 \subset U \cap V_2$ implique $F(U, V_1) > F(U, V_2)$).*

Définition 3 ($\mathcal{G}$-fonction) *Soit F une $\mathcal{F}$-fonction, et soit $g_k : \mathbb{R}^k \longrightarrow \mathbb{R}^+$, $k \in \mathbb{N}$, une suite de fonctions. Une* $\mathcal{G}$-fonction *est une fonction $G : \mathbb{P}[\mathcal{S}] \times \mathbb{P}[\mathbb{P}[\mathcal{S}]] \longrightarrow \mathbb{R}^+$ qui, pour toute partition $\mathcal{V} = \{V_1, ..., V_l\}$, associe à chaque $(U, \mathcal{V}) \in \mathbb{P}[\mathcal{S}] \times \mathbb{P}[\mathbb{P}[\mathcal{S}]]$ une valeur*

$$G(U, \mathcal{V}) = g_l(F(U, V_1), ..., F(U, V_l))$$

telle que :

iv) $G(U, \mathcal{V})$ soit **minimum** *quand il existe un unique V_j $(1 \leq j \leq l)$ tel que $F(U, V_j) = 0$ et pour tout V_k, $(1 \leq k \leq l,\ k \neq j)$, $F(U, V_k)$ est maximum.*

v) $G(U, \mathcal{V})$ soit **maximum** *quand $F(U, V_1) = ... = F(U, V_l)$.*

Définition 4 ($\mathcal{H}$-fonction) *Soit G une $\mathcal{G}$-fonction, et soit $h_k : (\mathbb{R}^+)^k \longrightarrow \mathbb{R}^+$, $k \in \mathbb{N}$, une suite de fonctions. Une* $\mathcal{H}$-fonction *est une fonction $H : \mathbb{P}[\mathbb{P}[\mathcal{S}]] \times \mathbb{P}[\mathbb{P}[\mathcal{S}]] \longrightarrow \mathbb{R}^+$, telle que, pour tout $(\mathcal{U}, \mathcal{V}) \in \mathbb{P}[\mathbb{P}[\mathcal{S}]] \times \mathbb{P}[\mathbb{P}[\mathcal{S}]]$, avec une partition $\mathcal{U} = \{U_1, ..., U_m\}$, $H(\mathcal{U}, \mathcal{V})$ est définie par*

$$H(\mathcal{U}, \mathcal{V}) = h_m(G(U_1, \mathcal{V}), ..., G(U_m, \mathcal{V}))$$

de sorte que :

vi) $H(\mathcal{U}, \mathcal{V})$ soit maximale quand pour tout i, $G(U_i, \mathcal{V})$ est maximal.

vii) $H(\mathcal{U}, \mathcal{V})$ soit minimale quand pour tout i, $G(U_i, \mathcal{V})$ est minimal.

1.3.3 Mesure de discrimination

Le modèle hiérarchique permet de définir maintenant plus formellement ce qu'est une mesure de discrimination dans le cadre de l'apprentissage inductif.

Définition 5 (Mesure de discrimination) *Soit $\mathbb{P}_C[\mathbb{P}[\mathcal{E}]]$, l'ensemble des partitions de $\mathcal{E}$ induites par la classe, et soit $\mathbb{P}_A[\mathbb{P}[\mathcal{E}]]$ l'ensemble des partitions de $\mathcal{E}$ induites par l'attribut A. On appelle* mesure de discrimination *toute $\mathcal{H}$-fonction $M : \mathbb{P}_A[\mathbb{P}[\mathcal{E}]] \times \mathbb{P}_C[\mathbb{P}[\mathcal{E}]] \longrightarrow \mathbb{R}^+$.*

En apprentissage inductif, on souhaite qu'une mesure de discrimination évalue le pouvoir de discrimination d'un attribut vis-à-vis de la classe, en prenant une valeur minimale quand toute valeur de l'attribut détermine sans ambiguïté la valeur de la classe, et en prenant une valeur maximale quand la connaissance de ses valeurs n'apporte aucune connaissance sur la valeur de la classe.

1.3.4 Application à des mesures existantes

Dans cette partie, nous allons voir que les mesures de discrimination les plus utilisées pour construire des arbres de décision flous, l'entropie d'événements flous, l'index flou de Gini, et la mesure d'ambiguïté de Yuan et Shaw, peuvent être considérées (sous certaines conditions dans le cas de l'entropie) comme de bonnes mesures de construction d'arbres.

Dans le cadre de notre hiérarchie, valider une fonction comme mesure de discrimination correcte revient donc à vérifier que la fonction en question est une $\mathcal{H}$-fonction, capable de se décomposer en $\mathcal{G}$-fonction et en $\mathcal{F}$-fonction.

Application à l'entropie floue

La $\mathcal{F}$-fonction $F : \mathbb{P}[\mathcal{E}] \times \mathbb{P}[\mathcal{E}] \longrightarrow \mathbb{R}^+$ pour l'entropie d'événements flous est définie par $F(U,V) = -\log\ p^*(V|U)$.

On montre dans ce qui suit que cette fonction F est bien une $\mathcal{F}$-fonction, quelle que soit la t-norme utilisée pour l'intersection de sous-ensembles flous.

- Si $U \subseteq {}^1V$, on a $\top(\mu_U(e), \mu_V(e)) = \mu_U(e)$, et donc $p^*(V|U) = \frac{\sum_{e\in\mathcal{E}} \top(\mu_U(e),\mu_V(e))}{\sum_{e\in\mathcal{E}} \mu_U(e)} = 1$. En conséquence, si $U \subseteq {}^1V$, alors $F(U,V) = 0$, et la propriété *i* est satisfaite.
- Si $U \cap V = \emptyset$ alors $p^*(V|U) = 0$. Donc $F(U,V)$ tend vers $+\infty$. En conséquence, $F(U,V)$ est maximum quand $U \cap V = \emptyset$ et la propriété *ii)* est satisfaite.
- Les t-normes étant monotones, il est facile de voir que F est continue et décroît avec $U \cap V$, la propriété *iii* est donc satisfaite.

À l'aide de la fonction F précédente, on peut définir la fonction G suivante.

Soit $\mathcal{V}$ une partition de $\mathcal{E}$ en l sous-ensembles flous V_k, on définit la fonction $G : \mathbb{P}[\mathcal{E}] \times \mathbb{P}[\mathbb{P}[\mathcal{E}]] \longrightarrow \mathbb{R}^+$ par :

$$G(U,\mathcal{V}) = \sum_{k=1}^{l} F(U,V_k)\ e^{-F(U,V_k)}.$$

Il est facile de vérifier qu'en utilisant la fonction F définie précédemment, on a $G(U,\mathcal{V}) = -\sum_{k=1}^{l} p^*(V_k|U) \log\ p^*(V_k|U)$, ce qui permet de retrouver G_E.

On doit vérifier maintenant que cette fonction G est bien une $\mathcal{G}$-fonction.

- Il est clair que $G(U,\mathcal{V})$ est minimum quand $F(U,V_j) = 0$ et que pour tout $k \neq j$, $F(U,V_k)$ est maximum (ce maximum tend vers $+\infty$). La propriété *iv* est donc vérifiée.
- Soit $F(U,V_1) = ... = F(U,V_l)$. Dans ce cas, $p^*(V_1|U) = ... = p^*(V_l|U)$. On a : $G(U,\mathcal{V}) = -\sum_{k=1}^{l} p^*(V_k|U) \log\ p^*(V_k|U)$. Et donc, si $\sum_{k=1}^{l} p^*(V_k|U) = 1$ alors $G(U,\mathcal{V})$ est une entropie et elle atteint son maximum pour $p^*(V_1|U) = ... = p^*(V_l|U)$. Cependant, $\sum_{k=1}^{l} p^*(V_k|U) = 1$ n'est pas vraie pour toute t-norme $\cap$. Et il nous faut donc décider ici laquelle utiliser :
  - avec la t-norme produit, on a

$$\sum_{k=1}^{l} p^*(V_k|U) = \frac{\sum_{e\in\mathcal{E}} \sum_{k=1}^{l} \mu_{V_k}(e)\mu_U(e)}{\sum_{e\in\mathcal{E}} \mu_U(e)}.$$

    Donc

$$\sum_{k=1}^{l} p^*(V_k|U) = \frac{\sum_{e\in\mathcal{E}} \mu_U(e) \sum_{k=1}^{l} \mu_{V_k}(e)}{\sum_{e\in\mathcal{E}} \mu_U(e)} = 1.$$

    Par conséquent, avec cette t-norme, la fonction G satisfait bien la propriété *v*.
  - avec la t-norme de Zadeh, le min, on ne peut pas garantir que $\sum_{k=1}^{l} p^*(V_k|U) = 1$. Ainsi, par exemple, pour $l = 2$, et avec $U \subseteq V_1$ et $U \subseteq V_2$, on a

$$p^*(V_1|U) = \frac{\sum_{e\in\mathcal{E}} \min(\mu_{V_1}(e), \mu_U(e))}{\sum_{e\in\mathcal{E}} \mu_U(e)} = 1$$

    et $p^*(V_2|U) = 1$, ce qui amène donc à $\sum_{k=1}^{2} p^*(V_k|U) \neq 1$. Par conséquent, on met en évidence un cas où $F(U,V_1) = F(U,V_2)$ implique $G(U,\mathcal{V}) = 0$ ce qui ne peut pas être considéré comme un maximum pour G. Ainsi, avec cette t-norme-ci, on montre qu'il existe au moins une possibilité que $G(U,\mathcal{V})$ ne soit pas maximum quand $F(U,V_1) =$

... = $F(U, V_n)$. Par conséquent, dans ce cas, G ne peut pas être considérée comme une $\mathcal{G}$-fonction car elle ne satisfait alors pas la propriété *v*.

La fonction G précédente peut être utilisée pour définir la fonction H suivante. Soit $\mathcal{U}$ une partition de $\mathcal{E}$ en m sous-ensembles flous U_j, $H : \mathbb{P}[\mathbb{P}[\mathcal{E}]] \times \mathbb{P}[\mathbb{P}[\mathcal{E}]] \longrightarrow \mathbb{R}^+$ est définie par

$$H(\mathcal{U}, \mathcal{V}) = \sum_{j=1}^{m} p(U_j) G(U_j, \mathcal{V}).$$

Si la fonction G est définie à l'aide de la t-norme produit alors G est une $\mathcal{G}$-fonction. De plus, la fonction H est alors une entropie et il facile de vérifier qu'elle satisfait les propriétés *vi* et *vii*.

Ainsi, on a mis en évidence une décomposition de l'entropie d'événements flous en trois niveaux de fonctions qui respectent les définitions des niveaux $\mathcal{F}$, $\mathcal{G}$, et $\mathcal{H}$ du modèle hiérarchique dans le cas où la t-norme produit est utilisée pour l'intersection de sous-ensembles flous. L'entropie est donc dans ce cas une mesure de discrimination.

On a pu aussi remarquer que, si l'on décide plutôt d'utiliser la t-norme de Zadeh, la fonction G ainsi construite n'est plus une $\mathcal{G}$-fonction et cela entraîne alors un comportement inadéquat de la fonction pour la mesure d'un pouvoir de discrimination.

Application à la mesure d'ambiguïté

Pour cette mesure, on définit la fonction $F : \mathbb{P}[\mathcal{E}] \times \mathbb{P}[\mathcal{E}] \longrightarrow \mathbb{R}^+$ comme : $F(U, V) = -\log\ p^*(V|U)$. C'est la même fonction que pour l'entropie d'événements flous précédente, et on a déjà montré que c'est bien une $\mathcal{F}$-fonction.

À partir de cette fonction F, on définit la fonction G suivante. Pour une partition $\mathcal{V}$ de $\mathcal{E}$ en l sous-ensembles flous V_k, la fonction $G : \mathbb{P}[\mathcal{E}] \times \mathbb{P}[\mathbb{P}[\mathcal{E}]] \longrightarrow \mathbb{R}^+$ est définie par (on utilise les notations introduites dans la Section 1.2.3, en particulier, on note $V_+ = V_{\rho(1)}$) :

$$G(U, \mathcal{V}) = \sum_{i=2}^{l} \frac{e^{-F(U, V_{\rho(i)})}}{p^*(V_+ \cap U)} \log(1 + \frac{1}{i-1})$$

Il est facile de voir que l'on retrouve bien G_Y en utilisant la fonction F donnée :

$$G(U, \mathcal{V}) = -\sum_{i=2}^{l} \frac{p^*(V_{\rho(i)} \cap U)}{p^*(V_+ \cap U)} \log(1 + \frac{1}{i-1}).$$

Maintenant, on vérifie que cette fonction G est une $\mathcal{G}$-fonction.

- On voit facilement que $G(U, \mathcal{V})$ est minimum quand il existe un unique V_j $(1 \leq j \leq l)$ tel que $F(U, V_j) = 0$ et que pour tout V_k, $(1 \leq k \leq l,\ k \neq j)$, $F(U, V_k)$ est maximum. La propriété *i* est donc satisfaite.
- Quand $F(U, V_1) = ... = F(U, V_l)$, on a $p^*(V_{\rho(i)} \cap U) = p^*(V_+ \cap U)$ pour tout i. Ainsi, $G_Y(U, \mathcal{V}) = \sum_{i=2}^{l} \log(1 + \frac{1}{i-1})$ et $G_Y(U, \mathcal{V}) = \log(l)$. Cette valeur est le maximum pour G : comme $\frac{p^*(V_{\rho(i)} \cap U)}{p^*(V_+ \cap U)} \leq 1$, il est clair que $\sum_{i=2}^{l} \frac{p^*(V_{\rho(i)} \cap U)}{p^*(V_+^l \cap U)} \log(1 + \frac{1}{i-1}) \leq \sum_{i=2}^{l} \log(1 + \frac{1}{i-1})$ et donc $G(U, \mathcal{V}) \leq \log(l)$. En conséquence, la propriété *ii* est satisfaite : $G(U, \mathcal{V})$ est maximum quand $F(U, V_1) = ... = F(U, V_l)$

À partir de la fonction G ainsi définie, on peut définir la fonction H suivante. Soit $\mathcal{U}$ une partition de $\mathcal{E}$ en m sous-ensembles flous U_j, on définit $H : \mathbb{P}[\mathbb{P}[\mathcal{E}]] \times \mathbb{P}[\mathbb{P}[\mathcal{E}]] \longrightarrow \mathbb{R}^+$ par

$$H(\mathcal{U}, \mathcal{V}) = \sum_{j=1}^{m} p(U_j) G(U_j, \mathcal{V}).$$

On montre facilement que H est une $\mathcal{H}$-fonction. En effet, pour tout j, on a $G(U_j, \mathcal{U}) = 0$ (minimum) car G est une $\mathcal{G}$-fonction (propriété *iv*). On a donc bien un minimum pour H. Par conséquent, la propriété *vi* est satisfaite. De la même façon, on montre que la propriété *vii* est aussi valide.

Ainsi, on peut donc mettre en évidence une décomposition de la mesure d'ambiguïté en trois niveaux de fonctions qui respectent les définitions des niveaux $\mathcal{F}$, $\mathcal{G}$ et $\mathcal{H}$ du modèle hiérarchique.

Application à l'index flou de Gini

La $\mathcal{F}$-fonction associée à cet index est identique à celle vue pour les deux précédentes mesures. Cette fonction $F : \mathbb{P}[\mathcal{E}] \times \mathbb{P}[\mathcal{E}] \longrightarrow \mathbb{R}^+$ est définie par $F(U, V) = -\log\ p^*(V|U)$.

À partir de cette fonction F, on définit la fonction G suivante. Pour $\mathcal{V}$, une partition en l sous-ensembles flous V_k, la fonction $G : \mathbb{P}[\mathcal{E}] \times \mathbb{P}[\mathbb{P}[\mathcal{E}]] \longrightarrow \mathbb{R}^+$ est définie par

$$G(U, \mathcal{V}) = \sum_{k=1}^{n} (e^{-F(U,V_k)})^2$$

Il est facile de vérifier qu'avec la fonction F donnée, on a bien $G(U, \mathcal{V}) = \sum_{k=1}^{l} p(V_k|U)^2$ qui permet donc de retrouver G_G.

On vérifie dans ce qui suit que la fonction G ainsi définie est bien une $\mathcal{G}$-fonction.

- On vérifie que $G(U, \mathcal{V})$ est minimum quand il existe un unique V_j $(1 \leq j \leq l)$ tel que $F(U, V_j) = 0$ et que pour tout V_k, $(1 \leq k \leq l,\ k \neq j)$, $F(U, V_k)$ est maximum. La propriété *iv* est donc satisfaite.
- Si on suppose que $F(U, V_1) = ... = F(U, V_l)$, et que $p^*(V_k|U) = K_V$ pour tout k. on a alors $G(U, \mathcal{V}) = l\ K_V^2$. On peut donc vérifier que l'on a :
  - avec la t-norme produit comme opérateur d'intersection, $K_V = \frac{1}{l}$ et $G(U, \mathcal{V}) = 1$ qui est un maximum pour G.
  - avec la t-norme de Zadeh, $K_V = 1$ et $G(U, \mathcal{V}) = l$ qui est un maximum pour G.

  Ainsi, on vient de vérifier que $G(U, \mathcal{V})$ est maximum quand $F(U, V_1) = ... = F(U, V_l)$ pour les deux t-normes considérées (produit et Zadeh), ce qui valide donc la propriété *v*.

On peut donc en conclure que G est bien une $\mathcal{G}$-fonction.

À partir de cette fonction G, nous pouvons définir la fonction H suivante. Si $\mathcal{U}$ est une partition en m sous-ensembles flous U_j, on définit $H : \mathbb{P}[\mathcal{E}] \times \mathbb{P}[\mathcal{E}] \longrightarrow \mathbb{R}^+$ par

$$H(\mathcal{U}, \mathcal{V}) = \sum_{j=1}^{m} p(U_j) G(U_j, \mathcal{V})$$

Pour tout j, si $G(U_j, \mathcal{V}) = 0$ (minimum) alors $H(\mathcal{U}, \mathcal{V}) = 0$ qui est un minimum pour H. Par conséquent, H vérifie la propriété *vi*. De même, on montre que H satisfait aussi la propriété *vii* et l'on vérifie donc bien que H est une $\mathcal{H}$-fonction.

En résumé, dans cette section, nous avons pu mettre en évidence une décomposition de l'index flou de Gini dans les trois niveaux de fonctions du modèle hiérarchique. Cet index est donc une $\mathcal{H}$-fonction, ce qui en fait une mesure de discrimination adéquate. De plus, nous avons pu montré cela quelle que soit la t-norme utilisée (produit ou Zadeh) pour définir l'intersection d'ensembles flous.

1.4 Comparer des mesures de discrimination

Dans la Section 1.3, nous avons vu comment choisir une bonne mesure de discrimination, c'est-à-dire une mesure correcte pour mesurer le pouvoir de discrimination d'attributs. Par contre, savoir qu'une mesure est correcte ne permet pas de savoir comment comparer cette mesure avec une autre mesure de discrimination. En particulier, il reste difficile de dire si une mesure donnée est plus adéquate qu'une autre et quelles différences il résultera de leur utilisation.

Dans cette partie, nous allons donc maintenant étudier plus finement les différences existantes entre des mesures de discrimination. Nous allons alors montrer comment la décomposition d'une mesure de discrimination en niveaux par le modèle hiérarchique, permet de mener ensuite une étude comparative des mesures. Notre étude portera sur les trois mesures que nous avons vues précédemment : l'entropie d'événements flous, l'index flou de Gini et la mesure d'ambiguïté. Nous nous restreindrons à ces mesures, car seuls les mécanismes de la comparaison nous intéressent, mais d'autres mesures pourraient être étudiées selon le même protocole

1.4.1 Correspondances entre les mesures

Pour simplifier, pour la mesure d'ambiguïté, considérons que les classes soient ordonnées de telle sorte que l'on ait $\Pi = \Pi^*$. Cette hypothèse peut être faite sans vraiment faire perdre en généralité étant donné qu'elle n'a pas d'influence sur les autres mesures : G_E et G_G sont des mesures symétriques complètement indépendantes de l'ordre des probabilités des valeurs c_1,..., c_K.

Sous cette hypothèse, V_1 est donc la valeur V_+^i telle que $V_+^i = \text{argmax}_{V_k} S(U_i, V_k))$, etc.

Nos trois mesures étudiées s'expriment donc dans une forme qui facilite l'examen de leurs correspondances :

$$G_E(\mathcal{V}|U_i) = -\sum_{k=1}^{l} \frac{p^*(V_k \cap U_i)}{p^*(U_i)} \log \frac{p^*(V_k \cap U_i)}{p^*(U_i)}, \tag{1.6}$$

$$G_G(\mathcal{V}|U_i) = 1 - \sum_{k=1}^{l} (\frac{p^*(V_k \cap U_i)}{p^*(U_i)})^2, \tag{1.7}$$

et

$$G_Y(\mathcal{V}|U_i) = \sum_{k=2}^{l} \frac{p^*(V_k \cap U_i)}{p^*(V_1 \cap U_i)} \log(1 + \frac{1}{k-1})$$

Cette dernière peut aussi s'écrire :

$$G_Y(\mathcal{V}|U_i) = \frac{p^*(U_i)}{p^*(V_1 \cap U_i)} \sum_{k=2}^{l} \frac{p^*(V_k \cap U_i)}{p^*(U_i)} \log(1 + \frac{1}{k-1}) \quad (1.8)$$

afin de bien faire ressortir la correspondance entre ces trois mesures liée à l'utilisation de $\frac{p^*(V_k \cap U_i)}{p^*(U_i)}$.

Maintenant, dans ce qui suit, nous allons étudier les variations de ces trois fonctions, G_E, G_G et G_Y, dans le but de mettre en évidence leur comportement vis-à-vis d'une légère modification dans la distribution des valeurs dans $\mathcal{U}$ et dans $\mathcal{V}$. L'idée est d'en déduire leur influence sur l'ordonnancement des attributs. On souhaite alors pouvoir répondre à la question : *"comment caractériser des attributs qui sont ordonnés différemment par ces mesures ?"*. Ceci revient donc à se demander quelles sont les composantes associées à un attribut qui jouent un rôle dans l'inversion de son ordonnancement par ces mesures.

Nous allons réaliser notre étude en deux étapes. Tout d'abord, nous allons étudier le cas où seules deux valeurs sont possibles dans $\mathcal{V}$ (en apprentissage, cela correspond alors à un cas de classes binaires). Ensuite, nous étudierons le cas plus général où $\mathcal{V}$ peut avoir plus de trois valeurs différentes.

1.4.2 En présence de classes binaires

On note V_+^i la classe majoritaire et V_-^i la classe minoritaire pour la valeur U_i de l'attribut étudié (on rappelle que l'on a $0 < p^*(V_+^i \cap U_i) \leq 1$ et $0 \leq p^*(V_-^i \cap U_i) < 1$). Pour alléger nos notations dans cette partie, on notera $p_i^* = p^*(U_i)$, $p_{i+}^* = p^*(V_+^i \cap U_i)$, et $p_{i-}^* = p^*(V_-^i \cap U_i)$.

Étant donné que l'on se trouve dans un cas à seulement deux classes, on a alors $p_{i-}^* = p_i^* - p_{i+}^*$ ce qui amène :

$$\begin{aligned} G_E(\mathcal{V}|U_i) &= -\frac{p_{i+}^*}{p_i^*} \log \frac{p_{i+}^*}{p_i^*} - \frac{p_i^* - p_{i+}^*}{p_i^*} \log \frac{p_i^* - p_{i+}^*}{p_i^*} \\ G_Y(\mathcal{V}|U_i) &= \frac{p_i^* - p_{i+}^*}{p_{i+}^*} \log 2 \end{aligned}$$

Nous considérerons maintenant que $\mathcal{U}$ contient exactement deux valeurs U_1 et U_2. On note plus simplement $p_1^* = p^*(U_1)$ et $p_2^* = p^*(U_2)$. On a alors :

$$\begin{aligned} H_E(\mathcal{V}|\mathcal{U}) &= p_1^* \cdot G_E(\mathcal{V}|, U_1) + p_2^* \cdot G_E(\mathcal{V}|U_2) \\ H_Y(\mathcal{V}|\mathcal{U}) &= p_1^* \cdot G_Y(\mathcal{V}|U_1) + p_2^* \cdot G_Y(\mathcal{V}|U_2) \end{aligned}$$

Soit, étant donné que $p_2^* = 1 - p_1^*$:

$$\begin{aligned} H_E(\mathcal{V}|\mathcal{U}) &= -p_1^* \cdot \big[\frac{p_{1+}^*}{p_1^*} \log \frac{p_{1+}^*}{p_1^*} \frac{p_1^* - p_{1+}^*}{p_1^*} \log \frac{p_1^* - p_{1+}^*}{p_1^*}\big] \\ &\quad -(1 - p_1^*) \cdot \big[\frac{p_{2+}^*}{1 - p_1^*} \log \frac{p_{2+}^*}{1 - p_1^*} + \frac{1 - p_1^* - p_{2+}^*}{1 - p_1^*} \log \frac{1 - p_1^* - p_{2+}^*}{1 - p_1^*}\big] \end{aligned}$$

et

$$H_Y(\mathcal{V}|\mathcal{U}) = p_1^* \cdot \big[\frac{p_1^* - p_{1+}^*}{p_{1+}^*} \log 2\big] + (1 - p_1^*) \cdot \big[\frac{1 - p_1^* - p_{2+}^*}{p_{2+}^*} \log 2\big]$$

Sous nos hypothèses, on voit donc clairement que $H_E(\mathcal{V}|\mathcal{U})$ et $H_Y(\mathcal{V}|\mathcal{U})$ ne dépendent que des trois variables p_1^*, la probabilité de la valeur majoritaire de $\mathcal{U}$, p_{1+}^* la probabilité de la valeur majoritaire de $\mathcal{V}$ relativement à U_1, et p_{2+}^* la probabilité de la valeur majoritaire de $\mathcal{V}$ relativement à U_2.

Il est alors possible de comparer les valeurs de $H_E(\mathcal{V}|\mathcal{U})$ et $H_Y(\mathcal{V}|\mathcal{U})$ en tant que fonctions de 3 variables :

$$\begin{aligned} H_E(\mathcal{V}|\mathcal{U}) &= f_E(p_1^*, p_{1+}^*, p_{2+}^*) \\ H_Y(\mathcal{V}|\mathcal{U}) &= f_Y(p_1^*, p_{1+}^*, p_{2+}^*) \end{aligned}$$

avec donc

$$\begin{aligned} f_E(x,y,z) &= -x \cdot \big[\frac{y}{x}\log\frac{y}{x} + \frac{x-y}{x}\log\frac{x-y}{x}\big] - (1-x)\cdot\big[\frac{z}{1-x}\log\frac{z}{1-x} \\ &\quad + \frac{1-x-z}{1-x}\log\frac{1-x-z}{1-x}\big] \\ &= x\log x + (1-x)\log(1-x) - y\log y - (x-y)\log(x-y) \\ &\quad -z\log z - (1-x-z)\log(1-x-z) \end{aligned}$$

et

$$f_Y(x,y,z) = \big[x(\frac{x}{y}-1) + (1-x)(\frac{1-x}{z}-1)\big]\log 2$$

où $x \in [0.5, 1]$, $\frac{y}{x} \in [0.5, 1]$ et $\frac{z}{1-x} \in [0.5, 1]$ (si $x \neq 1$, $z = 0$ sinon).

Étude des variations de f_E et de f_Y

Soit $\mathcal{V}$, un ensemble à deux valeurs. Considérons deux ensembles $\mathcal{U}_1$ et $\mathcal{U}_2$ qui respectent nos hypothèses. $\mathcal{U}_1$ est associé aux trois valeurs x_1, y_1, et z_1, et $\mathcal{U}_2$ est associé aux valeurs x_2, y_2, et z_2.

Ces deux ensembles sont ordonnés par H_E (resp. H_Y) selon leurs valeurs respectives $H_E(\mathcal{V}|\mathcal{U}_1)$ et $H_E(\mathcal{V}|\mathcal{U}_2)$ (resp. $H_Y(\mathcal{V}|\mathcal{U}_1)$ et $H_Y(\mathcal{V}|\mathcal{U}_2)$). On notera que "$\mathcal{U}_1$ est préféré à $\mathcal{U}_2$" selon H_E (resp. H_Y) par $\mathcal{U}_1 \succ_E \mathcal{U}_2$ (resp. $\mathcal{U}_1 \succ_Y \mathcal{U}_2$).

Nous allons caractériser le cas où l'on a conjointement $\mathcal{U}_1 \succ_E \mathcal{U}_2$ et $\mathcal{U}_2 \succ_Y \mathcal{U}_1$. Dans ce cas, $\mathcal{U}_1 \succ_E \mathcal{U}_2$ implique $H_E(\mathcal{V}|\mathcal{U}_1) < H_E(\mathcal{V}|\mathcal{U}_2)$ qui est équivalent à $H_E(\mathcal{V}|\mathcal{U}_1) - H_E(\mathcal{V}|\mathcal{U}_2) < 0$, et $\mathcal{U}_2 \succ_Y \mathcal{U}_1$ implique $H_Y(\mathcal{V}|\mathcal{U}_1) > H_Y(\mathcal{V}|\mathcal{U}_2)$ qui est équivalent à $H_Y(\mathcal{V}|\mathcal{U}_1) - H_Y(\mathcal{V}|\mathcal{U}_2) > 0$. Cela correspond donc à :

$$\frac{H_E(\mathcal{V}|\mathcal{U}_1) - H_E(\mathcal{V}|\mathcal{U}_2)}{H_Y(\mathcal{V}|\mathcal{U}_1) - H_Y(\mathcal{V}|\mathcal{U}_2)} < 0.$$

qui peut s'exprimer aussi :

$$\frac{f_E(x_1,y_1,z_1) - f_E(x_2,y_2,z_2)}{f_Y(x_1,y_1,z_1) - f_Y(x_2,y_2,z_2)} < 0. \tag{1.9}$$

Si l'on fixe les valeurs de deux des trois variables, on se retrouve dans un cas classique d'étude de fonctions.

Par exemple, soit y_1, z_1, y_2, et z_2 fixés. On considère que y_1 et y_2 (resp. z_1 et z_2) sont égaux et on note k_1 (resp. k_2) cette valeur ; Les deux fonctions se simplifient donc en :

$$\begin{aligned} f_{Ex} &= f_E(x, k_1, k_2), \\ f_{Yx} &= f_Y(x, k_1, k_2). \end{aligned}$$

Si on pose que $x_2 > x_1$, à partir de l'équation (1.9), par le théorème des accroissements finis, on peut dire qu'il existe donc une valeur $c \in [x_1, x_2]$ telle que

$$\frac{f_{Ex}(x_1) - f_{Ex}(x_2)}{f_{Yx}(x_1) - f_{Yx}(x_2)} = \frac{f'_{Ex}(c)}{f'_{Yx}(c)},$$

avec $f'_{Ex}(c) = \frac{df_{Ex}(x)}{dx}$ et $f'_{Yx}(c) = \frac{df_{Yx}(x)}{dx}$.

On doit donc étudier maintenant les dérivées partielles suivantes : $\frac{\frac{\partial f_E(x,y,z)}{\partial x}}{\frac{\partial f_Y(x,y,z)}{\partial x}}$, $\frac{\frac{\partial f_E(x,y,z)}{\partial y}}{\frac{\partial f_Y(x,y,z)}{\partial y}}$, et $\frac{\frac{\partial f_E(x,y,z)}{\partial z}}{\frac{\partial f_Y(x,y,z)}{\partial z}}$. Dans ce qui suit, on note ces rapports R_x, R_y, et R_z respectivement. On a : $\frac{\partial f_E(x,y,z)}{\partial x} = \log \frac{x(1-x-z)}{(x-y)(1-x)}$, $\frac{\partial f_E(x,y,z)}{\partial y} = \log(\frac{x}{y} - 1)$, $\frac{\partial f_E(x,y,z)}{\partial z} = \log(\frac{1-x}{z} - 1)$ et $\frac{\partial f_Y(x,y,z)}{\partial x} = 2(\frac{x}{y} - \frac{1-x}{z}) \log 2$, $\frac{\partial f_Y(x,y,z)}{\partial y} = -\frac{x^2}{y^2} \log 2$, $\frac{\partial f_Y(x,y,z)}{\partial z} = -\frac{(1-x)^2}{z^2} \log 2$.

$$\begin{aligned} \frac{\partial f_Y(x,y,z)}{\partial x} &= 2(\frac{x}{y} - \frac{1-x}{z}) \log 2 \\ \frac{\partial f_Y(x,y,z)}{\partial y} &= -\frac{x^2}{y^2} \log 2 \\ \frac{\partial f_Y(x,y,z)}{\partial z} &= -\frac{(1-x)^2}{z^2} \log 2 \end{aligned}$$

On trouve donc :

$$\begin{aligned} R_x &= \frac{1}{2 \log 2} \cdot \frac{\log(\frac{1-\frac{z}{1-x}}{1-\frac{y}{x}})}{\frac{x}{y} - \frac{1-x}{z}} \\ R_y &= -\frac{1}{\log 2} \cdot \frac{y^2}{x^2} \log(\frac{x}{y} - 1) \\ R_z &= -\frac{1}{\log 2} \cdot \frac{z^2}{(1-x)^2} \log(\frac{1-x}{z} - 1) \end{aligned}$$

On obtient alors $R_y \geq 0$ et $R_z \geq 0$, ainsi que $R_x \leq 0$.

On peut donc maintenant observer les différences de comportement de ces fonctions en étudiant ces formes différentielles. Dans un premier temps, on remarque que $\frac{\partial f_E}{\partial x}$ dépend de $\log x$ alors que $\frac{\partial f_Y}{\partial x}$ dépend de x. Ensuite, on voit que $\frac{\partial f_E}{\partial y}$ varie avec $\log x$ alors que $\frac{\partial f_Y}{\partial y}$ varie avec x^2. On voit donc que les variations en x ou en y auront donc plus d'impact sur f_Y que sur f_E.

De plus, on sait que x correspond à la distribution des exemples selon les valeurs de $\mathcal{U}$. Si on considère que $\mathcal{U}_1$ et $\mathcal{U}_2$ partitionnent chacun la population $\mathcal{E}$ en deux sous-ensembles : $\mathcal{E}_{11}$ et $\mathcal{E}_{12}$ pour U_1, $\mathcal{E}_{21}$ et $\mathcal{E}_{22}$ pour $\mathcal{U}_2$ (voir Figure 1.1), une valeur donnée x indique un nombre égal d'exemples de $\mathcal{E}_{11}$ et dans $\mathcal{E}_{21}$, et par conséquent un nombre égal d'exemples dans $\mathcal{E}_{12}$ et dans $\mathcal{E}_{22}$. On peut voir que $\frac{\partial f_E}{\partial y}$ et $\frac{\partial f_E}{\partial z}$ sont toujours négatifs, tout comme $\frac{\partial f_Y}{\partial y}$ et $\frac{\partial f_Y}{\partial z}$.

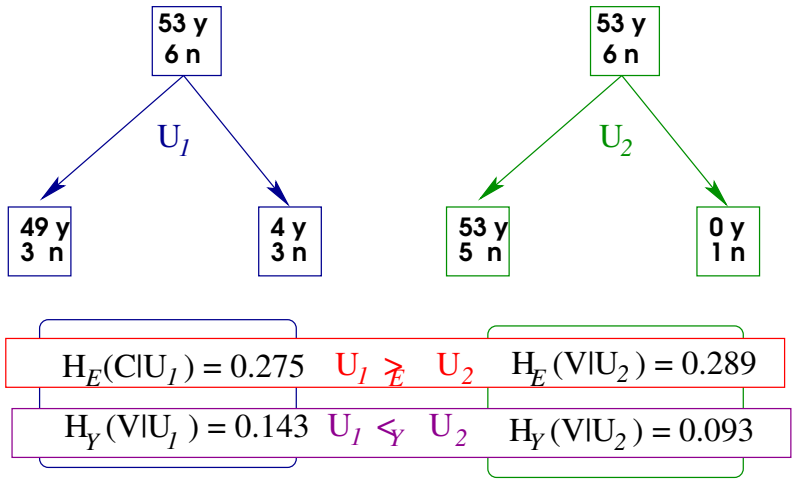

Figure 1.1 – Partition de la base d'apprentissage.

Cela signifie que f_E et f_Y varient de la même façon pour une valeur fixée x, et différentes valeurs pour y ce qui correspond à différentes fréquences d'exemples dans $\mathcal{E}_{11}$ et dans la classe majoritaire, comparés à ceux qui sont dans $\mathcal{E}_{21}$ et dans la classe majoritaire. Même chose du point de vue de z.

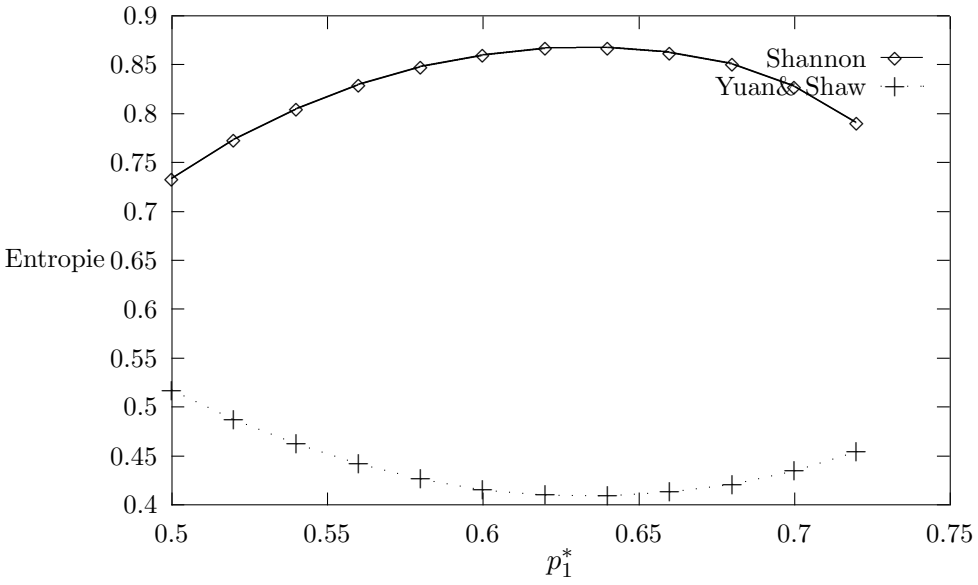

Figure 1.2 – Variations des entropies.

A l'opposé, si on fixe y et z, f_E et f_Y varient de façon contraire quand x varie (voir Figure 1.2). Plus précisément, si $y \geq z$, on observe que f_E atteint son maximum alors que f_Y atteint son minimum quand $x = \frac{y}{y+z}$. De même, si $y < z$, f_E est non croissante et f_Y est non décroissante.

Par exemple, les différences de variation entre H_E et H_Y sont données dans la Figure 1.2. Dans cette figure, on considère une base d'apprentissage avec $n = 100$ exemples, et y et z sont

fixés à 0.45 et 0.26 respectivement.

Une conséquence est que la seule façon d'obtenir des variations opposées pour f_E et f_Y repose sur x, ce qui met en évidence le fait que l'alternance $\mathcal{U}_1 \succ_E \mathcal{U}_2$ et $\mathcal{U}_2 \succ_Y \mathcal{U}_1$ est liée à cette variable. Quand $x = \frac{y}{y+z}$, f_E atteint son maximum, égal à

$$f_E^{\max} = -(y+z)\log(y+z) - (1-y-z)\log(1-y-z)$$

et f_Y atteint, lui, son minimum, égal à

$$f_Y^{\min} = \frac{1-y-z}{y+z}\log 2.$$

Interprétation

On se focalise ici sur le cas particulier où la valeur majoritaire de $\mathcal{V}$ est la même pour les deux valeurs de $\mathcal{U}$, $V_+^1 = V_+^2 = V_+$. Alors $x = \frac{y}{y+z}$ correspond à $p^*(V_+) = p^*(V_+|U_1)$ ce qui signifie que la valeur U_1 n'a aucune influence sur la reconnaissance de la valeur de $\mathcal{V}$. Les valeurs extrêmes de f_E et f_Y correspondent respectivement à la valeur d'entropie

$$f_E^{\max} = -p(V_+)\log p(V_+) - p(V_-)\log p(V_-)$$

et à la valeur de la mesure de non spécificité

$$f_Y^{\min} = \frac{p(V_-)}{p(V_+)}\log 2.$$

La différence entre ces deux mesures pour l'ordonnancement des deux attributs est donc seulement liée à la différence de probabilités des valeurs de chaque attribut.

On peut remarquer que si x (*ie.* $p^*(U_1)$) croît quand y (*ie.* $p^*(V_+ \cap U_1)$) est constant, alors $p^*(V_+|U_1) = \frac{p^*(V_+ \cap U_1)}{p^*(U_1)}$ décroît, et quand z (*ie.* $p^*(V_1 + \cap U_2)$) est constant, alors $p^*(V_+|U_2) = \frac{p^*(V_+ \cap U_2)}{p^*(U_2)}$ croît. Une conséquence est que H_E croît quand x croît jusqu'à ce que $p^*(V_+|U_1)$ et $p^*(V_+|U_2)$ deviennent égaux (maximum d'entropie), et elle décroît ensuite.

Ainsi, H_E favorise les attributs avec une grande différence de valeurs pour G_E. La "stratégie" pour cette mesure est alors de trouver des feuilles "pures" en premier lieu, même si cela entraîne la construction d'un arbre déséquilibré.

A l'opposé, H_Y favorise les attributs avec une petite différence de valeurs pour G_Y. Ici, la "stratégie" est de construire si possible un arbre équilibré. Dans ce cas, les arbres obtenus seront souvent plus large que les précédents et le nombre de questions pour trouver une feuille pure sera plus élevé.

La conséquence directe est qu'il apparaît donc que H_E favorise les attributs avec une grande différence dans les probabilités de ses valeurs, alors que H_Y favorise les attributs avec une relative égalité dans la répartition des probabilités de ses valeurs.

1.4.3 En présence d'au moins 3 classes

Maintenant, on se place dans le cas où il existe strictement plus de deux classes. Comme précédemment, on considére les classes ordonnées sur la fréquence des exemples qui la possèdent, et c_1 est la classe majoritaire. On considére de plus que la probabilité $p*(c_1)$ est fixée et on étudie les effets de variations sur les probabilités des autres classes.

Dans ce cas, $p^*(U_l)$, et $p^*(V_1 \cap U_l)$ sont constants, et seules les probabilités $\frac{p^*(V_k \cap U_l)}{p^*(U_l)}$ varient. On note $x_k = \frac{p^*(V_k \cap U_l)}{p^*(U_l)}$ et $x_1 = \frac{p^*(V_1 \cap U_l)}{p^*(U_l)}$ est constant. Avec ces notations, les équations (1.6), (1.7) et (1.8) peuvent se réécrire :

$$G_E(\mathcal{V}|U_l) = -x_1 \log x_1 - \sum_{k=2}^{K} x_k \log x_k \tag{1.10}$$

$$G_G(\mathcal{V}|U_l) = 1 - \sum_{k=1}^{K} x_k^2 \tag{1.11}$$

et

$$G_Y(\mathcal{V}|U_l) = \sum_{k=2}^{K} \frac{x_k}{x_1} \log(1 + \frac{1}{k-1}) \tag{1.12}$$

On rappelle qu'avec nos hypothèses, on a $1 > x_1 \geq x_2 \geq x_3 \geq ... \geq x_K > 0$.

Soit deux valeurs x_i et x_j, $x_i \geq x_j$, telles que x_i décroisse et x_j croisse avec une variation $\varepsilon \in [0, \frac{x_i - x_j}{2}]$, de telle sorte que l'on ait toujours $x_i - \varepsilon \geq x_j + \varepsilon$. On note $\bar{U}_l$ cette modification de U_l.

Variations de G_E

De l'équation (1.10), on obtient :

$$\begin{aligned} G_E(\mathcal{V}|\bar{U}_l) = & -x_1 \log x_1 - \sum_{k=2, k\neq i, k\neq j}^{K} x_k \log x_k \\ & -(x_i - \varepsilon)\log(x_i - \varepsilon) - (x_j + \varepsilon)\log(x_j + \varepsilon) \end{aligned}$$

Quand $\varepsilon > 0$, la variation $\Delta_E(\varepsilon)$ pour passer de $G_E(\mathcal{V}|U_l)$ à $G_E(\mathcal{V}|\bar{U}_l)$ est donnée par :

$$\Delta_E(\varepsilon) = -x_i \log(1 - \frac{\varepsilon}{x_i}) - x_j \log(1 + \frac{\varepsilon}{x_j}) + \varepsilon \log \frac{x_i - \varepsilon}{x_j + \varepsilon}$$

Si on étudie alors le comportement de la fonction Δ_E quand ε varie on a :

$$\frac{d\Delta_E(\varepsilon)}{d\varepsilon} = \log \frac{x_i - \varepsilon}{x_j + \varepsilon} \tag{1.13}$$

qui est toujours positif pour $\varepsilon \in [0, \frac{x_i - x_j}{2}]$.

De l'équation (1.13), on remarque que, quand ε augmente, la valeur de $G_E(\mathcal{V}|U_l)$ croît jusqu'à son maximum qui est $\varepsilon = \frac{x_i - x_j}{2}$. D'un côté, si on a $x_j = x_{i+1}$, on observe que $G_E(\mathcal{V}|U_l)$ est maximale quand $x_i + \varepsilon = x_{i+1} - \varepsilon$ *i.e.* quand les valeurs sont égales. D'un autre côté, $G_E(\mathcal{V}|U_l)$ est minimale quand les valeurs de x_i et de x_{i+1} sont très éloignées ($x_i >> x_{i+1}$) l'une de l'autre.

La propriété qui en ressort ici pour la mesure $G_E(\mathcal{V}|U_l)$ est qu'elle favorise donc les attributs qui partitionnent la base d'apprentissage en sous-ensembles d'exemples avec une classe fortement majoritaire et une grande différence, en proportion, avec les autres classes.

Variations de G_G

À partir de l'équation (1.11), on a :

$$G_G(\mathcal{V}|\bar{U}_l) = 1 - \sum_{k=1,k\neq i,k\neq j}^{K} x_k^2 - (x_i-\varepsilon)^2 - (x_j+\varepsilon)^2$$

Quand $\varepsilon > 0$, la variation de $G_G(\mathcal{V}|U_l)$ à $G_G(\mathcal{V}|\bar{U}_l)$ est donnée par :

$$\Delta_G(\varepsilon) = 2\varepsilon(x_i - x_j - \varepsilon)$$

ce qui donne donc :

$$\frac{d\Delta_G(\varepsilon)}{d\varepsilon} = 2(x_i - x_j - 2\varepsilon) \tag{1.14}$$

qui est une valeur toujours positive pour $\varepsilon \in [0, \frac{x_i - x_j}{2}]$.

On vient donc de mettre en évidence que G_G et G_E ont le même comportement.

Variations de G_Y

À partir de l'équation (1.12), on arrive à des conclusions différentes :

$$\begin{aligned} G_Y(\mathcal{V}|\bar{U}_l) = \frac{1}{x_1}\Big[& \sum_{k=2,k\neq i,k\neq j}^{K} x_k \log(1+\frac{1}{k-1}) \\ & +(x_i-\varepsilon)\log(1+\frac{1}{i-1}) \\ & +(x_j+\varepsilon)\log(1+\frac{1}{j-1})\Big] \end{aligned}$$

et donc :

$$\begin{aligned} G_Y(\mathcal{V}|\bar{U}_l) = \frac{1}{x_1}\Big[& \sum_{k=2,k\neq i,k\neq j}^{K} x_k \log(1+\frac{1}{k-1}) \\ & +x_i\log(1+\frac{1}{i-1}) + x_j\log(1+\frac{1}{j-1}) \\ & -\varepsilon\log(1+\frac{1}{i-1}) + \varepsilon\log(1+\frac{1}{j-1})\Big] \end{aligned}$$

Quand $\varepsilon > 0$, la variation de $G_Y(\mathcal{V}|U_l)$ à $G_Y(\mathcal{V}|\bar{U}_l)$ est donnée par :

$$\Delta_Y(\varepsilon) = \frac{\varepsilon}{x_1}(\log(1+\frac{1}{j-1}) - \log(1+\frac{1}{i-1}))$$

Ce qui donne donc :

$$\frac{d\Delta_Y(\varepsilon)}{d\varepsilon} = \frac{1}{x_1}(\log(1+\frac{1}{j-1}) - \log(1+\frac{1}{i-1}))$$

Ainsi, à partir de $x_i \geq x_j$ on a $i < j$ et donc $\frac{d\Delta_Y(\varepsilon)}{d\varepsilon} < 0$ pour tout ε.

Il est donc aisé de voir que $G_Y(\mathcal{V}|U_l)$ décroît quand ε croît, et donc, $G_Y(\mathcal{V}|U_l)$ est minimum quand $x_i - \varepsilon = x_j + \varepsilon$. Si on a $x_j = x_{i+1}$, cela conduit à, d'une part, $G_Y(\mathcal{V}|U_l)$ minimum pour $x_i + \varepsilon = x_{i+1} - \varepsilon$ *i.e.* quand les valeurs sont égales, d'autre part, $G_Y(\mathcal{V}|U_l)$ maximum quand les valeurs x_i et x_{i+1} sont le plus éloignées ($x_i >> x_{i+1}$) l'une de l'autre.

Une importante propriété est mise en évidence ici pour la mesure $G_Y(\mathcal{V}|U_l)$, elle montre que cette mesure favorise les attributs qui permettent de partitionner la base d'apprentissage en sous-ensembles dont un possède une classe qui est majoritaire, et les autres sous-ensembles possèdent une proportion proche d'exemples des autres classes.

1.4.4 Alors quelle mesure prendre ?

Notre étude met en évidence deux propriétés importantes qui différencient les trois mesures que nous avons décidé d'étudier dans le cas où il y a plus de deux classes.

Pour les mesures G_E et G_G, on remarque qu'elles favorisent les attributs qui partitionnent la base d'apprentissage en sous-ensembles d'exemples avec une classe fortement majoritaire et une grande différence, en proportion, avec les autres classes

Par contre, la mesure G_Y favorise, elle, les attributs qui permettent de partitionner la base d'apprentissage en sous-ensembles dont un possède une classe qui est majoritaire, et les autres sous-ensembles possèdent une proportion proche d'exemples des autres classes.

Ainsi, dans ce cas où il y a plus de deux classes, on remarque que la proportion des classes dans les sous-ensembles induits par l'attribut est importante pour chaque mesure, mais pas de la même façon. Même si la probabilité pour v_l est la même pour les deux attributs, ils seront ordonnés différemment par ces trois mesures s'ils conduisent à des proportions d'exemples différentes pour les classes.

Cela donne une information sur le comportement de ces mesures pour la construction d'un arbre de décision flou. Si on utilise l'entropie d'événements flous, l'arbre de décision flou possédera des chemins qui permettront de caractériser très rapidement une classe particulière. Lors de la descente dans les branches de l'arbre, à chaque nœud, on trouvera une question très discriminante pour une des classes (si elle existe), la discrimination vis-à-vis des autres classes pouvant se faire de façon moins forte. Ce genre d'arbres sera très utile s'il est primordial d'avoir une classification très rapide d'exemples "critiques". Par exemple, dans une application médicale, si on doit décider de la possibilité d'une attaque cardiaque chez un patient, il est primordial de pouvoir faire la détection le plus rapidement possible (et donc en minimisant si possible le nombre de questions à poser pour classer le patient).

Si on utilise la mesure d'ambiguïté pour construire un arbre de décision flou, on obtiendra un arbre beaucoup plus équilibré. Pour obtenir la classe d'un exemple lors de la phase de classification, il sera nécessaire de poser un nombre de questions similaire (ou très similaire) quel que soit le chemin pris dans l'arbre. Ce type d'arbre sera très utile dans des applications où l'on recherchera une base de règles équilibrées, les règles possédant un nombre équivalent de prémisses.

D'autre part, on a aussi remarqué que, dans le cas où il n'existe que deux classes, la proportion des classes après le partitionnement par un attribut n'est pas importante pour l'ordonnancement des attributs à l'aide de mesures étudiées. Ce qui fait la différence dans les ordonnancements dépend seulement de la probabilité de v_l.

Une remarque finale peut ici être faite sur la conséquence de ces propriétés que nous venons de mettre en évidence. Si ces propriétés ne donnent aucune information supplémentaire sur le taux de bonnes classifications que l'on pourrait s'attendre à avoir avec un arbre de décision flou

construit selon la mesure choisie, elles nous renseignent de façon précise sur la forme de l'arbre construit.

1.5 Événements conditionnels possibilistes et indépendance

Jusqu'à présent, nous avons essayé de définir les qualités nécessaires à une mesure afin qu'elle puisse être utilisée correctement comme mesure de discrimination. Nous sommes partis des propriétés mathématiques requises dans un processus de discrimination afin de caractériser pleinement les mesures adéquates pour la mise en œuvre de ce processus.

Une autre façon de procéder est de raisonner non pas directement sur la mesure mais sur la notion d'événement même, l'idée étant de dégager précisément ce que l'on entend par événement dans un cadre de discrimination et de l'appliquer ainsi au processus de discrimination pour la construction d'un arbre de décision par exemple.

Ainsi, la théorie des probabilités manipule des événements et leurs probabilités d'occurrence (ou de réalisation). Un événement peut être le résultat d'une observation ou d'une expérience, on peut considérer qu'il correspond à une *proposition* E qui peut être soit vraie, soit fausse. Mais en fait, on ne peut souvent pas connaître a priori la véracité de cette proposition, ce qui entraîne alors une incertitude sur E. Dans ce cas, la probabilité (d'occurrence) de E peut être considérée comme le degré avec lequel on estime probable sa réalisation. Mais, plus délicat, et plus problématique, est la question de savoir comment définir la mesure de la probabilité d'occurrence d'un événement conditionnée par l'occurrence d'un autre événement.

En théorie des probabilités, la *probabilité conditionnelle* d'un événement E_i sachant qu'un autre événement E_j s'est produit est définie à partir de la probabilité conjointe d'occurrence de E_i et E_j, et de la probabilité de E_j.

Mais une autre façon de procéder est de définir directement un *événement conditionnel* $E_i|E_j$, $E_j \neq \emptyset$, par exemple, à partir de sa valeur de vérité $T(E_i|E_j)$ qui est égale à :

i) $T(E_i|E_j) = 1$ si E_i et E_j sont vrais

ii) $T(E_i|E_j) = 0$ si E_i^c et E_j sont vrais

iii) $T(E_i|E_j) = t(E_i|E_j)$ si E_j est faux, avec t une fonction à valeurs dans $[0, 1]$.

La fonction t peut être définie comme une fonction de $(E_i \wedge E_j, E_i^c \wedge E_j, E_j^{c)}$

A partir de cette définition d'événement conditionnel proposée par Giuliannella Coletti, nous avons mené des travaux conjoints avec Giuliannella Coletti et Bernadette Bouchon-Meunier sur le conditionnement possibiliste et la notion d'indépendance dans ce cadre-là. Mais l'application de ces travaux aux mesures de discrimination est encore à réaliser et fait partie des perspectives de mon travail sur les mesures. Plus de détails sur les résultats de nos travaux peuvent être trouvés dans [7, 16, 51, 53].

1.6 Conclusion

Dans ce chapitre, j'ai présenté les travaux que j'ai menés sur les mesures de discrimination et leur utilisation pour l'ordonnancement des attributs lors de la construction d'un arbre de décision flou.

Cela m'a amené à proposer deux études. Dans une première étude, j'ai proposé un nouveau modèle hiérarchique de fonctions, qui étend celui que j'avais proposé durant ma thèse, et qui

permet de vérifier la validité d'une mesure pour son utilisation comme mesure de discrimination. Ce modèle m'a permis de vérifier que les mesures les plus couramment utilisées lors de la construction d'arbres de décision flous sont pertinentes pour un tel usage. Dans une seconde étude, j'ai proposé une étude des différences entre deux mesures afin de mettre en évidence les effets que leur utilisation produit sur l'arbre de décision flou construit.

1.7 Références

Les travaux de ce chapitre poursuivent et étendent mes travaux de thèse. Je les ai réalisés en travaillant avec Bernadette Bouchon-Meunier.

D'autre part, en parallèle, des travaux sur ce thème ont fait l'objet de la thèse de Thanh Ha Dang que j'ai co-encadré (de 2003 à 2007) avec Bernadette Bouchon-Meunier.

Quelques références :

- travaux avec Bernadette Bouchon-Meunier : [30, 39, 40, 43] ;
- travaux avec Thanh Ha Dang : [32], [38] ;
- co-encadrement de la thèse de Thanh Ha Dang : [36], soutenue en juillet 2007 ;
- travaux avec Bernadette Bouchon-Meunier et Giuliannella Coletti [7, 16, 51, 53].

Chapitre 2

Combinaisons de modèles flous

2.1 Introduction

Dans ce chapitre, nous présentons différentes approches pour combiner des modèles afin d'en augmenter leurs performances.

En plus des méthodes existantes bien connues, qui ne sont ici rappelées que dans les grandes lignes, nous présentons plus en détail les forêts d'arbres de décision flous. Les forêts mettent en œuvre une combinaison d'arbres de décision flous afin d'améliorer leurs performances en classification. De plus, elles offrent un bon moyen pour leur permettre de mieux prendre en compte des imperfections de la base d'apprentissage, comme le déséquilibre des distributions des classes, ou le contexte multi-classes (qui est plus délicat à gérer avec des modèles d'apprentissage comme les constructeurs d'arbres de décision) en le transformant en plusieurs contextes bi-classes.

2.2 État de l'art

L'idée de combiner des classifieurs n'est pas nouvelle, de nombreux travaux sur ce thème ont été menés depuis (au moins) les années 1990. Dans le domaine statistique, cela fait longtemps que sont utilisées des méthodes d'estimation de paramètres de distribution de probabilités. Ainsi, les méthodes de type *leave one out*, la *cross-validation*, et le boostrap sont des outils usuels dans ce domaine. Leur passage dans le domaine de l'apprentissage automatique s'est fait naturellement à partir des années 1990.

En fait, le domaine de l'inférence statistique a introduit beaucoup de méthodes qui ont par la suite été reprises en apprentissage artificiel. Cela n'est pas surprenant quand on sait qu'en inférence statistique, la problématique de base est de prédire une distribution de probabilité (inconnue a priori) à partir d'un échantillon censé suivre cette distribution. Pour reprendre Efron qui écrit dans [46] (p. 20) : *"Statistical inference concerns learning from experience : we observe a random sample* $x = (x_1, x_2, ..., x_n)$ *and wish to infer properties of the complete population* $\mathcal{X} = (X_1, X_2, ..., X_N)$ *that yielded the sample. Probability theory goes in the opposite direction : from the composition of a population* $\mathcal{X}$ *we deduce the properties of a random sample* x*, and of statistics calculated from* x*. Statistical inference as a mathematical science has been developed almost exclusively in terms of probability theory".*

Des approches comme le bagging, les random forests, et arbres aléatoires sont des adaptations spécifiques pour construire des "ensembles" de classifieurs et dérivent des travaux sur le boostrap.

Il existe d'autres méthodes, comme le boosting, qui s'attachent à améliorer les performances d'un classifieur en le multipliant.

Ce qui différencie ces algorithmes est très bien résumé par [9] : *"Voting algorithms can be divided into two types : those that adaptively change the distribution of the training set based on the performance of previous classifiers (as in boosting methods) and those that do not (as in Bagging)."*.

Dans [60], une approche originale de construction d'ensembles de classifieurs est proposée. L'idée est de combiner des classifieurs qui fournissent un ordonnancement (*ranking*) lors de la classification d'un ensemble d'exemples. Un seuil dans cet ordonnancement est alors utilisé pour sélectionner les exemples les mieux classés. A partir du résultat de plusieurs classifieurs, une agrégation permet d'optimiser l'ordonnancement final de l'ensemble des exemples.

2.2.1 Le Bootstrap originel

Le bootstrap n'est pas à l'origine une méthode d'apprentissage, mais il a été introduit par la suite afin d'étudier le comportement d'algorithmes d'apprentissage. Il a donné lieu à une méthode d'apprentissage basée sur l'utilisation de ses spécificités.

Le bootstrap a été introduit par Efron dans le cadre de l'estimation biais - variance d'une distribution de probabilité [45, 46]. Dans cette approche, l'idée était de pouvoir estimer des paramètres d'une distribution de probabilité F en ne connaissant qu'un échantillon quelconque de taille n suivant F.

Étant donnée une population $X = (x_1, x_2, ..., x_n)$ qui est une réalisation d'une distribution de probabilité F (que l'on souhaite étudier à l'aide de X), le bootstrap se met en œuvre comme suit :

1. on considère que chaque x_i a une probabilité d'occurrence de $\frac{1}{n}$,
2. on extrait aléatoirement de X un échantillon de taille n que l'on appelle *l'échantillon bootstrap*, le tirage est effectué avec replacement,
3. on utilise l'échantillon bootstrap pour approcher la distribution F et pour en estimer ainsi les paramètres.

Par la suite, le bootstrap a été appliqué en apprentissage automatique afin d'estimer le taux d'erreur d'un algorithme d'apprentissage d'une manière différente que ne pouvait le faire une cross-validation [46, 47].

2.2.2 Le Bootstrap en apprentissage

Dans l'utilisation en apprentissage du bootstrap, on considère une base d'apprentissage $X = (x_1, x_2, ..., x_n)$, dont les observations $x_i = (t_i, y_i)$ sont constituées d'un *prédicteur* t_i (ou *vecteur de caractéristiques*), et d'une réponse y_i. Grâce à X, les statisticiens sont capables de construire un *système de prédiction* $r_X(t)$, qui associe à tout t une réponse y correspondante. Une fois r_X construit, il est alors souvent nécessaire d'estimer le taux d'erreur qui interviendra lors de la prédiction de la réponse obtenue pour de nouveaux vecteurs de caractéristiques.

Pour réaliser cette estimation, une méthode courante était de réaliser une cross-validation. Pourtant, Efron et Tibshirani ont mis en évidence le fait que si une cross-validation permettait une estimation peu biaisée du taux d'erreur cherché, elle avait l'inconvénient d'être soumise à une grande variabilité [48].

Pour pallier ceci, ils proposèrent d'utiliser le bootstrap afin de réduire cette variabilité de l'estimation du taux d'erreur recherché. Ils ont alors proposé de tirer indépendemment un grand

nombre d'échantillons boostrap de X afin d'obtenir ainsi un grand nombre d'échantillons de la population et de réduire ainsi la variance de l'estimation qui en sera faite.

2.2.3 Bagging

Cette méthode d'apprentissage, dont le nom est un acronyme pour *Bootstrap aggregating* (*Bagging*), a été introduite par Breiman [27]. Elle est très proche du boostrap, dont elle diffère par la finalité. Le bagging est une méthode d'apprentissage à part entière qui a pour but de créer un ensemble de prédicteurs afin d'en augmenter le pouvoir de prédiction.

La procédure est du même type que le boostrap : à partir d'une base d'apprentissage $\mathcal{E}$, on sélectionne n exemples de $\mathcal{E}$ pour construire $\mathcal{E}_k$, un sous-ensemble d'exemples de la base d'apprentissage. Le tirage est effectué avec remplacement, ainsi, chaque exemple est tiré aléatoirement dans $\mathcal{E}$ et peut apparaître plusieurs fois dans $\mathcal{E}_k$. La distribution des exemples dans $\mathcal{E}_k$ doit approximer la distribution des exemples dans $\mathcal{E}$. À partir de $\mathcal{E}_k$, on construit un classifieur Φ_k

On voit donc qu'ici, le but est d'utiliser les échantillons non plus pour estimer le taux d'erreur d'un classifieur mais pour constituer un ensemble de classifieurs.

2.2.4 Boosting

Introduit en 1990 [103],[9], [52], le boosting est une approche qui propose de combiner un ensemble d'apprenants dits "faibles" (*weak learner*) afin d'obtenir un système plus efficace.

La "force" d'un apprenant a été introduit par Valiant et Kearns (source : [103]). Un *strong learner* est un algorithme, applicable en temps polynomial, qui atteint un faible taux d'erreur (avec une grande confiance) pour tous les concepts de la classe. Un *weak learner* est un algorithme permettant de générer un classifier (généralement pour un problème bi-classes) dont la performance est garantie comme étant meilleure qu'une prédiction aléatoire.

Dans ses travaux, [103] a par la suite montré que l'on pouvait améliorer les performances d'un *weak learner* en l'utilisant dans une combinaison d'un ensemble de *weak learner*. Il introduisit alors le *boosting* d'algorithme d'apprentissage afin d'en optimiser les performances : "Boosting *refers to a general and provably effective method of producing a very accurate prediction rule by combining rough and moderately inacurate rules of thumb ...*" [54]. Le boosting a été mis en œuvre dans AdaBoost qu'ont proposé [53].

Le principe du boosting est de faire se focaliser l'algorithme d'apprentissage sur les parties de l'espace d'apprentissage qui sont les plus difficiles à apprendre. Pour réaliser cela, chaque exemple de la base est associé à un poids (initialement uniforme pour tous les exemples) qui doit être pris en compte par l'algorithme d'apprentissage pour la construction de son modèle (ou *hypothèse*). Une fois construit, l'hypothèse est utilisée pour classer les exemples : les exemples bien classés voient leur poids se réduire, et les exemples mal classés voient leur poids augmenter. Une fois tous les exemples classés, on peut alors calculer un coût (ou marge d'erreur) de l'hypothèse en sommant les variations de poids qu'elle a entraînées. Selon le nombre d'étapes que l'on s'est fixé, et selon l'importance du coût, on recommence alors éventuellement le processus et on reconstruit une nouvelle hypothèse à partir des nouveaux poids obtenus.

Au final, on obtient donc une séquence d'hypothèses, chacune associée à son coût, qui sera utilisée pour classer tout nouvel exemple. Une aggrégation pondérée classique des résultats de classification de chaque hypothèse pour l'exemple permettra d'obtenir la classification de l'exemple par l'ensemble des hypothèses.

Comme il est montré dans [103], l'intérêt du boosting est sa capacité d'augmenter les performances de n'importe quel weak learner.

2.2.5 Forêts aléatoires

Les algorithmes à base de forêts aléatoires existent, sous différentes formes, depuis le milieu des années 1990. Parmi les premières approches, on peut citer celle de [61, 62]. Afin d'optimiser les taux de reconnaissance des arbres de décision, l'auteur proposa de multiplier les arbres construits à partir de la base d'apprentissage. Le moyen proposé a alors été de construire des arbres sur des sous-espaces aléatoires (*random subspaces*)) de l'espace des caractéristiques. Ainsi, dans cette approche, c'est la description des exemples (*ie.* la liste des attributs) qui est modifiée afin de différencier les arbres construits. Un tirage aléatoire des attributs à conserver pour décrire les exemples de la base d'apprentissage est réalisé, et un arbre de décision est alors construit avec cette base. La forêt est constituée par un ensemble d'arbres construits en faisant à chaque fois un tirage aléatoire pour conserver les attributs qui décrivent les exemples. On peut aussi noter que l'algorithme de construction d'arbres choisi par l'auteur est d'abord un algorithme d'induction d'arbres obliques [61], puis, plus tard, l'algorithme C4.5 [62].

Pour classer de nouveaux exemples à l'aide de cette forêt, une agrégation des décisions de chaque arbre est réalisée par un vote (avec la possibilité, pour un arbre, de répartir son vote sur plusieurs classes si l'exemple est arrivé dans une classe "impure" de cet arbre).

Dans son article, [43] mène une comparaison de 3 méthodes pour construire des ensembles d'arbres de décision. Outre le bagging et le boosting, il introduit une méthode aléatoire pour construire les arbres de décision d'une forêt. Dans cette approche, ce sont les tests dans les nœuds internes de l'arbre qui sont choisis de façon aléatoire. En fait, pour construire un nœud d'un arbre, le test est choisi aléatoirement parmi les 20 meilleures coupures qui entraînent un gain d'information. Pour les attributs continus, le choix du seuil de discrétisation est lui aussi effectué de façon aléatoire et rentre lui aussi dans le classement des meilleures coupures. Les résultats de ces expérimentations placent le boosting comme la méthode donnant les meilleurs résultats, alors que le bagging et la méthode aléatoire apportent des résultats à peu près similaires, l'auteur remarquant même que la méthode aléatoire était un peu meilleure que le bagging en présence de données faiblement bruitées.

Un des algorithmes de forêts aléatoires les plus connus en apprentissage automatique, l'algorithme des *Random Forests*, a été introduit par Breiman [28] à la suite de ses travaux sur le bagging. Une forêt aléatoire est un ensemble d'arbres de décision.

Chaque arbre de décision est construit en prenant un échantillon de la base d'apprentissage X fournie. Cet échantillon est construit en prenant aléatoirement des exemples dans X, le tirage étant réalisé avec remise, comme pour le bagging. Comme le dit [28] : *"A random forest is a classifier consisting of a collection of tree-structured classifiers* $\{h(x, \Theta_k),\ k = 1, ...\}$ *where the* $\{\Theta_k\}$ *are* independent identically distributed random *vectors and each tree casts a unit vote for the most popular class at input* x*"*.

2.2.6 Ensembles d'arbres complètement aléatoires

L'algorithme des "*arbres extrêmement aléatoires*" repose sur une construction entièrement aléatoire des arbres de décision. Le choix des attributs constituants les nœuds, comme le choix des seuils de discrétisation pour les attributs numériques, est effectué par un tirage complètement aléatoire parmi l'ensemble des choix possibles [55]. À l'aide de cet algorithme, [55] construisent un ensemble d'arbres aléatoires (une forêt d'arbres aléatoires).

À la différence des méthodes de construction d'ensembles classiques, la forêt n'est pas obtenue ici en prenant des échantillons (type échantillons bootstrap) aléatoires de la base des exemples fournie comme base d'apprentissage pour construire les arbres de décision, mais en prenant

toujours toute la base des exemples comme base d'apprentissage, et en construisant les arbres aléatoirement.

Dans cette approche, l'aléatoire se place donc au niveau de la construction des arbres de décision, et non plus au niveau d'une étape de présélection des exemples d'apprentissage.

2.3 Les forêts d'arbres de décision flous

2.3.1 Forêts primitives

Une première approche, primitive, pour la construction de forêts d'arbres flous avait déjà été proposée durant ma thèse [82, 79]. En présence de problèmes à plus de 2 classes, dans l'application Tanit, un ensemble de Salammbô était lancé, chaque Salammbô ayant pour tâche de construire un arbre de décision flou séparant une classe contre toutes les autres réunies. Lors de sa classification à l'aide de cette forêt d'arbres de décision flous, un nouveau cas était classé, dans un premier temps, par chacun des arbres de la forêt, puis, ensuite, les résultats des classifications de tous les arbres de la forêt étaient agrégés (différentes méthodes avaient été étudiées alors : agrégation par votes (normalisés ou non), agrégation possibiliste).

Malgré l'aspect "sommaire" de l'approche, les résultats en classification étaient prometteurs car ils surclassaient la plupart du temps, et souvent de façon drastique, les résultats obtenus par un arbre unique.

Par contre, son inconvénient premier était de construire des arbres à partir de distributions déséquilibrées : aucun ré-échantillonnage n'était effectué alors sur la population des exemples afin de rétablir une équiprobabilité dans la représentativité des classes.

2.3.2 Forêts pour les classes déséquilibrées

C'est un travers bien connu de la méthode : lorsque le déséquilibre entre les distributions des classes dans la base d'apprentissage est trop important, la construction d'un arbre de décision (flou ou non) devient problématique. Selon la mesure de discrimination qui sera utilisée, il peut être alors peu pertinent d'utiliser la base d'apprentissage telle quelle pour construire un arbre de décision avec nos algorithmes.

La raison principale en est que la classe majoritaire (ou les classes majoritaires s'il y en a plusieurs) aura tendance à être fortement favorisée lors des choix d'attributs, ainsi que lors de la mesure du critère d'arrêt (si on se fixe un seuil non nul autorisant une impureté des feuilles). Une classe trop fortement minoritaire sera masquée par la classe majoritaire lors des calculs de discrimination.

Par exemple, dans l'application sur la prévention des maladies cardio-vasculaires, deux classes étaient à reconnaître (il fallait détecter si une personne était à risque ou non) et une des classes représentaient seulement 5% des exemples de la base d'apprentissage (cette application sera décrite dans le Chapitre 4).

Dans ce genre de problèmes, il est donc naturel de chercher un rééquilibrage des classes avant de construire un arbre de décision. L'approche la plus triviale est d'extraire aléatoirement un échantillon de la base de départ en forçant une équiprobabilité des classes.

2.3.3 Forêts d'arbres de décision flous

Construction d'une forêt.

Algorithme 1 Construction d'une forêt d'arbres de décision flous

Input :
Base d'apprentissage $\mathcal{E}$,
$n > 0$, le nombre d'arbres souhaité,
$p \in]0, 1]$, la proportion d'exemples de la classe minoritaire à conserver
$0 < M < |\mathcal{E}|$, le nombre maximum d'exemples d'une même classe à garder.
L_p, liste des paramètres pour le constructeur d'arbres de décision flou
Output : $\mathcal{F}$, une forêt de n arbres de décision flous
Begin
$\mathcal{F} \leftarrow \{\}$
$c \leftarrow$ nombre d'exemples de la classe minoritaire dans $\mathcal{E}$
$m \leftarrow \min(p \cdot c,\ M)$ /* nombre d'exemples d'une même classe à prendre */
$i \leftarrow 1$
while $i \leq n$ **do**
$\mathcal{E}_i \leftarrow$ tirer aléatoirement m exemples de chaque classe dans $\mathcal{E}$
$\mathcal{A}_i \leftarrow$ construire un arbre de décision flou à partir de $\mathcal{E}_i$ avec les paramètres L_p
$\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathcal{A}_i\}$
$i \leftarrow i + 1$
endwhile
End.

Dans cet algorithme 1, on peut noter que l'on préjuge d'un élément important : les arbres construits le sont en utilisant pour tous le même paramétrage de l'algorithme de construction d'arbres (même mesure de discrimination, même critère d'arrêt, etc.). En fait, il peut être plus pertinent d'utiliser un paramétrage différent afin d'obtenir des arbres encore plus différents les uns les autres. De plus, on peut aussi choisir d'utiliser une forêt plus hétérogène, constituée de classifieurs différents les uns les autres, mais construite sur le même principe.

En ce qui concerne le tirage aléatoire, celui-ci peut se faire avec remise (pour réaliser donc un bagging d'arbres de décision flous) ou sans remise.

Utilisation d'une forêt

Dans l'algorithme 2, deux fonctions particulières doivent être précisées :

- `classerAvecArbre(`$\mathcal{A}$, e`)` : c'est la fonction usuelle d'utilisation d'un arbre de décision pour le classement d'un nouvel exemple. Ici, c'est une classification qui peut être floue (si l'arbre l'est) et qui peut aussi être de différentes formes. En effet, d'une part, la classification d'un exemple par un arbre flou peut rendre :
  - une classe unique, ainsi qu'un degré d'appartenance (qui peut être implicitement 1),
  - une liste de classes, chacune associée à un degré d'appartenance,

  et, d'autre part, les degrés d'appartenance ainsi rendus peuvent être :
  - soit normalisés, leur somme valant 1 ;
  - soit non normalisés, chaque degré étant alors le rendu "brut" de l'arbre flou.
- `fusionnerLesRésultats(`L_A, L_W`)` : cette fonction a pour mission de combiner les résultats individuels de la classification par les arbres de la forêt et de fournir en retour, comme dans le point précédent, soit une classe unique, soit une liste de classes, associées à des degrés d'appartenance (normalisés ou non) ou associées à des quantificateurs flous. Cette agrégation peut, éventuellement, utiliser une pondération (éventuellement équivalente)

Algorithme 2 Classification d'un exemple avec une forêt d'arbres de décision flous

Input :
$\mathcal{F}$, une forêt d'arbres de décision flous ;
e, un exemple à classer ;
L_W, une liste de poids pour chaque arbre de la forêt.
Output : L_F, liste qui donne pour chaque classe c_k le degré d'appartenance $\mu_k(e) \in [0,1]$
Begin
$\mathcal{F}_s \leftarrow \{\}$ /* Liste des arbres qui ont déjà classé e */
$L_A \leftarrow \{\}$ /* Liste des résultats individuels de classification de e */
while $\mathcal{F}_s \neq \mathcal{F}$ **do**
$\mathcal{A} \leftarrow$ choisir un arbre de $\mathcal{F}$ qui n'est pas dans $\mathcal{F}_s$
$R \leftarrow$ `classerAvecArbre(`$\mathcal{A}$, e`)`
$L_A \leftarrow L_A \cup \{R\}$
$\mathcal{F}_s \leftarrow \mathcal{F}_s \cup \{A\}$
endwhile
$L_F \leftarrow$ `fusionnerLesRésultats(`L_A, L_W`)`
End.

pour chaque arbre (reflétant ainsi une certaine mesure de la confiance qui peut être accordée à chacun des arbres). Elle peut se faire de différentes façons :
- par un vote simple,
- par un vote pondéré,
- par une combinaison plus sophistiquée (*cf.* par exemple, l'utilisation des OWA opérateurs [111])
- par apprentissage, en utilisant un algorithme d'apprentissage entraîné sur la même base d'apprentissage que celle utilisée pour construire les arbres, ou sur une partie distincte de la base d'apprentissage qui aura été réservée pour cette tâche.

Nous développons un peu cette fonction (cruciale) dans la partie qui suit.

2.3.4 L'agrégation dans une forêt d'arbres de décision flous

Dans cette section, on considère que l'on a donc :
- un exemple e dont on cherche la classe (ou les classes selon le cas ou l'application) parmi un ensemble $\mathcal{C}$ de n_K classes,
- une liste L_F de résultats de classification par chacun des arbres d'une forêt : chaque arbre $\mathcal{A}_i$ de la forêt (on considère que la forêt comporte un nombre fini, n, d'arbres) a fourni, pour chaque classe c_k de l'ensemble des classes, un degré d'appartenance $\mu_{ik}(e) \in [0,1]$ avec lequel il estime que l'exemple possède la classe,
- une liste L_W de pondérations qui associe à chaque arbre $\mathcal{A}_i$ un poids $w_i \in [0,1]$ qui rend compte de la "confiance" que l'on a dans l'arbre.

Éventuellement, une normalisation peut être demandée à la fois pour les degrés d'appartenance (mais alors seulement pour un même arbre), mais aussi pour les poids de L_W. Elle s'exprime ainsi :
- normalisation des degrés d'appartenance aux classes : pour tout i, $\sum_{k=1}^{n_K} \mu_{ik}(e) = 1$,
- normalisation des pondérations : $\sum_{i=1}^{n} w_i = 1$. Dans ce cas-là, il faut noter que l'on a alors tendance à considérer que la confiance dans un arbre dépend aussi de la confiance que l'on a dans les autres arbres de la forêt, ce qui peut être un implicite assez gênant et peu

réaliste si l'on ne sait pas évaluer les arbres en fonction des uns et des autres.

Comme on l'a vu dans la partie précédente, lors de la classification d'un exemple par une forêt, on en arrive au problème de devoir agréger un ensemble de résultats de classification, donnés par chacun des arbres de la forêt. Cette agrégation peut se faire de différentes façons, en fonction du problème, de la taille de la forêt, ou de différentes autres contingences.

Dans ce qui suit, nous présentons quelques types d'agrégation usuels.

Le vote simple et strict

Chaque arbre de la forêt a droit à un vote, de même poids que celui des autres, sur la classe envisagée de l'exemple. De plus, on considère qu'un arbre ne peut voter que pour une classe unique, et on associe le degré 1 à son vote. Si tel n'est pas le cas, on ne retient alors, pour chaque arbre, que la classe pour l'exemple e a obtenu le degré d'appartenance le plus élevé.

La classe de l'exemple pour la forêt est donc celle qui obtient le plus grand nombre de votes parmi les arbres qui la composent.

Le vote simple et flou

De la même façon que précédemment, chaque arbre a droit à un vote, mais on lui garde la possibilité de répartir son vote sur plusieurs classes en leur associant un degré d'appartenance (cette répartition pouvant être normalisée ou non selon le cas).

Le vote pondéré

Ici, on associe chaque vote $\mu_{ik}(e)$ d'un arbre avec le poids w_i qui lui est associé. Cela amène une nouvelle question : comment agréger un vote avec un poids ? Le poids devant atténuer le degré d'appartenance, on peut suggérer de réaliser le produit de ces deux valeurs ($w_i \cdot \mu_{ik}(e)$) mais on peut aussi préférer utiliser le minimum ($\min(w_i, \mu_{ik}(e))$).

L'agrégation évoluée

Il existe un grand nombre de fonctions d'agrégation qui sont utilisables pour ce genre de problèmes. Par exemple, parmi toutes les méthodes existantes, on peut citer les *Ordered Weighted Averaging* (OWA) opérateurs d'agrégation introduits par [111]. Mais d'autres méthodes pourraient ici être introduites comme, par exemple, l'utilisation de quantificateurs.

L'apprentissage

Un choix possible est aussi d'utiliser un algorithme d'apprentissage. Ainsi, un algorithme d'apprentissage peut être entraîné à agréger les résultats de classification des arbres de la forêt. Cela permettrait d'obtenir une agrégation prenant en compte une certaine spécificité dans les arbres de la forêt.

Un moyen de réaliser cela serait d'utiliser le résultat de classifications sur une base d'exemples, différents de ceux utilisés pour construire les arbres, ou sur une partie distincte de la base d'apprentissage qui aurait été réservée pour cette tâche.

Par exemple, dans le cas où l'on souhaite que l'agrégation fournisse une classe et une seule pour e, on peut se placer dans le cadre d'un apprentissage supervisé : les classes de $\mathcal{C}$ restent les classes à reconnaître bien sûr, et les descripteurs utilisés pour la prédiction de cette classe ont pour valeur les degrés obtenus par chacun des arbres. Une partie de la base d'apprentissage

de départ est alors réservée afin de permettre la construction d'un classifieur de ce type. Les exemples de cette sous-base sont classés par les arbres de la forêt, et les résultats obtenus sont utilisés pour la constitution d'une base d'apprentissage pour construire un arbre de décision flou (ou un autre modèle) qui servira d'agrégateur.

2.3.5 Les autres forêts d'arbres de décision flous

De la même manière que les méthodes de combinaison de classifieurs, et d'arbres de décision en particulier, sont très à la mode dans la communauté d'apprentissage automatique, depuis quelques années, quelques méthodes de construction de forêts d'arbres de décision flous sont apparues dans la communauté d'apprentissage floue. Par contre, on peut remarquer que l'appellation "forêt d'arbres de décision flous" ne respecte pas forcément les mêmes critères selon les auteurs qui l'utilisent. On ne retrouve pas forcément non plus les mêmes critères uniques de définition que ce qui peut être vu en apprentissage automatique classique.

En particulier, on peut dénombrer trois grands types d'approches :

- les méthodes qui introduisent des réétiquettages des données de la base d'apprentissage afin de construire des arbres différents. C'est le type d'approche que nous avons développé durant ma thèse [82]. Dans cette approche, l'algorithme de construction d'arbres flous est l'algorithme standard.
- les méthodes qui utilisent un rééchantillonnage de la base d'apprentissage pour construire plusieurs arbres qui seront donc autant de visions différentes d'une partie des données disponibles. Ces travaux sont très récents, on peut citer l'article de [11] qui présente une approche encore a l'état préliminaire mais qui s'appuie complètement sur les random forests de Breiman. Dans cette approche aussi, l'algorithme de construction d'arbres flous est l'algorithme standard.
- les méthodes qui travaillent sur la même base d'apprentissage, sans la ré-échantillonner, et qui introduisent seulement des variantes dans l'algorithme de construction d'arbres de décision flous en altérant le processus de sélection d'attribut, soit en éliminant des attributs de la liste des attributs [33], soit en retenant un groupe d'attributs au lieu d'un seul [65].

Forêt d'arbres reconnaissant 1 classe contre les autres

Dans ces travaux [82], plusieurs réétiquettages des données de la base d'apprentissage sont réalisés. Un arbre de la forêt sera associé à chacune des classes existantes. Pour le construire, le réétiquettage consistera à définir une classe unique, commune, pour tous les exemples des autres classes que celle que l'arbre doit reconnaître. Ainsi, l'arbre construit aura pour tâche de discriminer une classe contre toutes les autres.

Lors de sa classification à l'aide de cette forêt d'arbres de décision flous, un nouveau cas était classé, dans un premier temps, par chacun des arbres de la forêt, puis, ensuite, les résultats des classifications de tous les arbres de la forêt étaient agrégés (différentes méthodes avaient été étudiées alors : agrégation par votes (normalisés ou non), agrégation possibiliste).

Forêt de type random forest

Un article récent traite de l'utilisation d'arbres de décision flous dans la combinaison de classifieurs [11]. Dans cet article qui décrit, comme ils le précisent par ailleurs, des travaux préliminaires, les auteurs proposent à leur tour de construire des forêts d'arbres de décision flous sur le modèle des *Random Forest* de Breiman. Malheureusement, les auteurs ne soulèvent

quasiment pas de questions sous-jacentes à l'utilisation des forêts : taille critique de la forêt (en nombre d'arbres), apport des arbres flous par rapport aux arbres classiques, etc.

Forêt construite en éliminant des attributs.

Des forêts d'arbres de décision flous, d'au moins deux arbres, sont utilisées par [33]. Dans ces travaux, la construction de la forêt se fait en deux étapes :

1. une séquence de n arbres de décision (non flous) est créée. Le premier arbre de la séquence est construit en considérant tous les attributs disponibles. Le second arbre est construit en enlevant de la liste des attributs disponibles, l'attribut racine du premier arbre. Et ainsi de suite, on retire l'attribut racine pour construire l'arbre suivant etc., jusqu'à ce qu'on l'on ait construit n arbres. D'après [33], une petite valeur de n est souvent suffisante (entre 3 et 10).
2. ensuite, tous les arbres de décision sont transformés en règles floues. Tous les seuils de décision *ie.* des fonctions d'appartenance sont associées au test dans l'arbre de décision) sont fuzzifiés et une optimisation des paramètres de ces seuils par algorithmes génétiques est réalisée.

Ces travaux sont intéressants et utilisent une méthode spécifique de construction d'arbres de décision (le retrait d'attribut avant la construction), mais ils ne se basent pas sur l'utilisation de véritables arbres de décision flous car la fuzzification n'est réalisée qu'à l'issue de la construction de l'arbre.

Forêt construite en gardant plusieurs attributs.

Le terme de forêt d'arbres de décision flous a aussi été utilisé dans [65, 66]. Dans ce travail, une forêt est un ensemble d'arbres issus du même processus de construction. Lors de la construction de l'arbre, le choix de l'attribut pour partitionner la base d'apprentissage se faisant avec une mesure, il se peut que plusieurs attributs soit également sélectionnables. Dans ce cas, Janikow décide de ne pas sélectionner un attributs parmi ceux-ci, et de les prendre tous pour construire donc plusieurs noeuds, qui donneront dont plusieurs variantes du même arbre (la distinction intervenant ici sur le partionnement induit par le choix de l'attribut qui amène donc des sous-arbres qui lui sont propres).

2.4 Expérimentations

Un ensemble d'expérimentations ont été réalisées pour étudier l'impact de la taille de la forêt en nombre d'arbres de décision flous sur le taux d'erreur en classification. Pour plus de détails, je renvoie à l'article [22] où certaines de ces expérimentations sont décrites.

D'autre part, la méthode de construction et d'utilisation de forêts d'arbres de décision flous décrite dans la Section 2.3.3 a été appliquée dans le cadre de la participation au challenge TrecVid (*cf.* Section 4.4).

2.5 Conclusion

Dans cette partie, nous avons présenté une étude sur les forêts d'arbres de décision flous. La combinaison de plusieurs arbres de décision flous permet de construire un classifieur augmentant

grandement les capacités de classification d'un arbre de décision flou unique. Par contre, une telle amélioration ne se fait pas sans dommage pour le pouvoir explicatif du modèle basé sur les arbres de décision flous. Le nombre de règles augmentant d'autant que le nombre d'arbres dans la forêt. Une perspective de ce travail sera donc d'étudier les possibilités de réduction de la taille de cette forêt en ne retenant qu'un ensemble d'arbres suffisant pour lui permettre d'obtenir de bons résultats en généralisation tout en lui conservant un bon pouvoir explicatif.

2.6 Références

Les travaux de ce chapitre ont donné lieu à plusieurs publications, dont beaucoup sont liées aux applications qui m'ont amené à développer ces forêts :

- premiers travaux sur les forêts d'arbres flous : [82].
- applications des forêts en video mining (voir la Section 4.4) : [10, 79, 90, 92].
- sur la construction et l'utilisation des forêts : [22]

Chapitre 3

Modèles d'apprentissage et de raisonnement flous

3.1 Introduction

Dans ce chapitre, j'ai décidé de regrouper les travaux que j'ai menés, en collaboration avec plusieurs de mes collègues de l'équipe LoFTI, sur les modèles d'apprentissage et de raisonnement flou.

L'idée sous-jacente de ces travaux est que ces deux modèles sont fortement reliés entre eux dans nos domaines de recherche : l'un est concerné par l'utilisation des bases de règles floues (qui peuvent être incomplètes) et l'autre se focalise plutôt sur les façons de construire des règles floues. Le lien est donc évident, les utilisations de l'un et de l'autre sont fondamentales et demandent à être étudiées plus en détails, et conjointement.

Dans un premier temps, je présente les travaux communs que j'ai menés avec Bernadette Bouchon-Meunier et Anne Laurent et dans lesquels nous avons réalisé une étude mettant en parallèle deux modèles d'apprentissage de règles floues : la méthode de construction de résumés flous (à base de règles graduelles floues), et la méthode de construction d'arbres de décision flous. Cette étude nous a permis de mettre en évidence différentes possibilités pour améliorer certains aspects de chacun de ces deux modèles en lui faisant profiter des avantages de l'autre.

Ensuite, je présente les travaux que nous avons menés avec Bernadette Bouchon-Meunier, Maria Rifqi, et d'autres collègues, sur le raisonnement interpolatif (qui met en œuvre l'utilisation de bases de règles incomplètes). Nous avons choisi le raisonnement interpolatif par opposition au raisonnement flou usuel qui est souvent mieux étudié et pour lesquels des méthodes de raisonnement "éprouvées" existent déjà. Le raisonnement interpolatif est généralement moins bien défini et il méritait donc de servir de cadre à de nouvelles recherches. Je développe plus particulièrement le modèle de raisonnement interpolatif pour des règles multi-prémisses que nous avons proposé.

Dans cette partie, je présente aussi une approche possible pour l'apprentissage de règles incomplètes afin de les mettre en œuvre avec un raisonnement interpolatif. Cette approche repose sur l'utilisation d'une méthode à base d'arbres de décision flous qui reste encore un outil idéal dans un tel but.

Ces deux travaux de recherche, sur les modèles d'apprentissage de règles floues et sur l'interpolation, sont reliés car la connaissance d'un modèle de raisonnement par interpolation fiable et efficace offre la possibilité, par la suite, de gérer des bases de règles incomplètes (c'est-à-dire

qui ne couvrent pas tout l'espace des valeurs de description des données du problème étudié) qui peuvent avoir été générées par n'importe quelle méthode d'apprentissage de règles.

Finalement, je présente, dans la dernière partie, un bilan des collaborations qui m'ont permis de réaliser les travaux de ce chapitre et je rappelle les références des publications qui en ont été issues.

3.2 Modèles d'apprentissage de règles floues

Dans cette partie, je présente les travaux que j'ai menés avec Anne Laurent et Bernadette Bouchon-Meunier et qui ont pour cadre l'étude de deux méthodes d'apprentissage de règles de décision floues.

Il est un fait connu que l'on peut considérer un arbre de décision (flou) comme une base de règles (floues) [79, 8]. Chaque chemin dans l'arbre, de la racine vers une feuille, est une règle dont la prémisse est composée par les valeurs d'attributs rencontrées sur chaque arc sortant des nœuds qui ont été pris. La décision associée par cette règle est donnée par la (ou les) valeur(s) de la classe trouvée(s) dans la feuille à laquelle on arrive par ce chemin. Un algorithme de construction d'arbres de décision flous à partir d'une base d'apprentissage est donc un moyen d'apprendre à construire une base de règles floues par induction.

D'un autre côté, Anne Laurent a travaillé sur les résumés flous, qui sont aussi des règles floues, et les règles d'association floues, et sur leur génération à partir d'une base de données (et aussi de bases de données multidimensionnelles). De même, Bernadette Bouchon-Meunier a beaucoup travaillé sur les quantificateurs et les règles graduelles que l'on retrouve souvent dans l'énoncé de règles floues.

Ainsi, avec Bernadette et Anne, nous avons donc décidé d'étudier plus avant les points communs et les différences qui pouvait exister entre les deux méthodes de génération de règles que sont la construction d'arbres flous et la construction de résumés flous, ainsi que leurs liens avec les quantificateurs linguistiques. L'idée première était de pouvoir faire profiter chacune de ces méthodes des avantages des autres approches. On peut donc espérer pouvoir améliorer d'une part les algorithmes de construction impliqués dans chacune de ces approches, mais aussi, augmenter grandement l'interprétabilité des modèles de règles obtenus en y insérant des quantificateurs.

3.2.1 Les arbres de décision flous

Je vais revenir ici sur le processus de construction d'arbres de décision flous. La présentation sera succincte car plus de détails ont déjà été donnés dans [79].

La plupart des algorithmes (du moins, pour les algorithmes non incrémentaux) procèdent de la même manière pour construire un arbre de décision. Cette manière, appelée approche *Top Down Induction of Decision Tree* (TDIDT), construit l'arbre de façon récursive, de sa racine à ses feuilles, en réalisant des partitions successives de la base d'apprentissage. Chaque partition est réalisée par un test sur les valeurs d'un attribut (ce qui engendre une partition des exemples de la base selon la valeur qu'ils possèdent pour cet attribut-là). Ce test sur les valeurs de l'attribut donne alors naissance à un nœud de l'arbre. L'algorithme TDIDT de construction d'arbres est rappelé plus formellement dans l'Algorithme 3.

Le meilleur attribut pour réaliser cette partition est choisi à l'aide d'une *mesure de discrimination* H. Une telle mesure permet d'ordonner les attributs en fonction de leur *pouvoir de discrimination* relativement aux valeurs de la classe. L'attribut avec le plus haut pouvoir de discrimination est sélectionné pour constituer un nœud de l'arbre de décision. Grâce à cet

attribut, et à ses valeurs, un partitionnement de la base d'apprentissage est réalisé selon une stratégie de partitionnement P. Un critère d'arrêt T permet d'arrêter ce processus de partition et de construire une feuille de l'arbre.

Dans l'Algorithme 3, les fonctions suivantes sont nécessaires :

- `ensembleSuivant(`$\mathcal{B}$`)` : rend un ensemble d'apprentissage de $\mathcal{B}$.
- `vérifieCritereArret(`T, E, C`)` : regarde si le critère d'arrêt T, éventuellement dépendant de C, est vérifié pour l'ensemble d'exemples E.
- `rajouteUneFeuille(`$\mathcal{T}$, E`)` : rend l'arbre de décision flou obtenu en rajoutant, au bon endroit, une feuille constituée par les exemples de E.
- `rajouteUnNoeud(`$\mathcal{T}$, A`)` : rend l'arbre de décision flou obtenu en rajoutant, au bon endroit, un nœud constituée à l'aide des valeurs de l'attribut A.
- `meilleurAttribut(`H, $\mathcal{A}$, E`)` : rend l'attribut de $\mathcal{A}$ qui est optimal (*ie.* minimise) selon la mesure H pour la base E.
- `partitionneBase(`P, E, A`)` : découpe en sous-ensembles l'ensemble E selon les valeurs de A en respectant la stratégie de partitionnement P.

Algorithme 3 Construction d'un arbre de décision flou

Input :
Base d'apprentissage $\mathcal{E}$,
Ensemble d'attributs $\mathcal{A}$,
$C \notin \mathcal{A}$ un attribut particulier appelé *classe*,
un critère d'arrêt T
une stratégie de partitionnement P
une mesure de discrimination H
Output : $\mathcal{T}$ un arbre de décision flou
Begin
  $\mathcal{T} \leftarrow \{\}$
  $\mathcal{B} \leftarrow \{\mathcal{E}\}$
  while $\mathcal{B} \neq \emptyset$ **do**
    $E \leftarrow$ `ensembleSuivant(`$\mathcal{B}$`)`
    $\mathcal{B} \leftarrow \mathcal{B}/E$
    if `vérifieCritereArret(`T, E, C`)` **then**
      $\mathcal{T} \leftarrow$ `rajouteUneFeuille(`$\mathcal{T}$, E`)`
    else
      $A \leftarrow$ `meilleurAttribut(`H, $\mathcal{A}$, E`)`
      $\mathcal{T} \leftarrow$ `rajouteUnNoeud(`$\mathcal{T}$, A`)`
      $\{\mathcal{E}_1, ..., \mathcal{E}_p\} \leftarrow$ `partitionneBase(`P, E, A`)`
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{\mathcal{E}_1, ..., \mathcal{E}_p\}$
    end if
  endwhile
End.

Comme il a été montré précédemment, beaucoup de méthodes TDIDT de construction d'arbres de décision, ne diffèrent en fait que dans le choix de la mesure H qui est utilisée pour mesurer le pouvoir discriminant des attributs [79].

On sait que les algorithmes classiques, comme l'algorithme ID3 [99], fonctionnent très bien en présence d'attributs dont les valeurs sont symboliques. Par contre, ils rencontrent quelques

difficultés pour la prise en compte des attributs numériques, imprécis, ou à valeurs floues.

Pour prendre en compte les valeurs floues, les algorithmes de construction d'arbres classiques ont été adaptés grâce à la théorie des sous-ensembles flous. Un panel de telles méthodes à été présenté dans [79]. Peu de méthodes nouvelles pour construire des arbres de décision flous sont apparues depuis.

Dans le cadre des arbres de décision flous, on peut distinguer deux familles principales de méthodes de construction. La première est composée par les méthodes qui sont basées sur l'utilisation d'une généralisation de l'entropie de Shannon (l'entropie d'événements flous) comme mesure de discrimination [24, 64, 100, 110, 112]. La seconde est composée par les méthodes qui utilisent une mesure non "entropique" [26, 31, 109, 97]. Dans cette famille-ci, on peut mettre en évidence une mesure de discrimination très intéressante, introduite par [112]. La mesure qu'il utilise pour construire un arbre de décision flou est une mesure d'ambiguïté de classification qui est définie à la fois comme une mesure d'inclusion floue, et comme une mesure de non-spécificité. Parmi les publications très récentes, on peut citer, essentiellement par souci d'exhaustivité, [30] qui reprend l'utilisation d'une version floue de l'index de Gini pour construire des arbres flous. Mais cet article n'est pas vraiment convainquant car les auteurs n'étudient pas les propriétés de cette mesure afin d'en justifier son utilisation pour choisir des attributs. D'autre part, dans cet article, les auteurs se contentent à proposer une fuzzification de l'index de Gini, sans vraiment expliquer *pourquoi* une nouvelle méthode est nécessaire et en quel sens les méthodes floues existantes sont outrepassées et nécessitent une nouvelle approche. Ils ne proposent d'ailleurs ensuite aucune comparaison avec les méthodes floues existantes. Plus intéressants sont, par contre, les travaux de [1, 2].

Nous avons présenté, dans le Chapitre 1, une étude sur le comportement comparé de ces deux principales mesures de choix d'attributs et je renvoie donc à ce chapitre-là pour plus de détails sur ces mesures.

3.2.2 Les résumés flous

Une présentation très rapide et succincte des résumés flous est faite dans cette partie. Mon but ici est de donner les éléments qui permettent de situer la comparaison que l'on a menée avec Anne Laurent entre les arbres flous et les résumés flous. Pour plus de détails et pour des références plus complètes, voir [71, 72, 73, 12, 63].

Rappels

Des résumés flous sont généralement construits à partir d'une base de données $D = \{y_1, ..., y_n\}$, dans laquelle chaque objet y_i $(i = 1, ..., n)$ est décrit par un ensemble d'attributs. Un résumé flou est une règle de la forme [12] :

$$Q\ A\ y_i\ sont\ C : \tau$$

avec Q un quantificateur flou, y_i un objet issu d'une base de donnée, A et C sont des critères flous, et τ est un degré de vérité. Les quantificateurs flous sont des sous-ensembles flous de l'univers des fréquences $[0, 1]$ et les critères flous sont comme des sous-ensembles flous définis sur les univers des objets y_i. En général, Q, A et C sont représentés par leur fonctions d'appartenance respective μ_Q, μ_A and μ_C. Des exemples de quantificateurs, *peu*, *la moitié*, et *la plupart*, sont donnés dans la Figure 3.1.

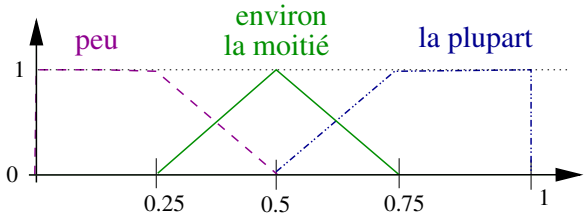

Figure 3.1 – Exemples de quantificateurs (extrait de [12])

Le degré de vérité τ peut être estimé de différentes façons. En particulier, dans une approche inductive fondée sur la connaissance d'un ensemble de n objets, Anne Laurent [12] donne la forme suivante pour calculer τ :

$$\tau = \mu_Q \left(\frac{1}{n} \sum_{i=1}^{n} \top(\mu_A(y_i), \mu_C(y_i)) \right) \tag{3.1}$$

Construction de résumés

Dans ses travaux de thèse, Anne Laurent a développé une approche pour la construction de résumés flous basée sur les méthodes de génération de règles d'association classiques. Les règles d'association qu'elle considère sont de la forme `Antécédent` $\rightarrow$ `Conséquent`. L'antécédent et le conséquent sont formés par une combinaisons de valeurs des attributs disponibles dans le problème considéré.

La génération de règles d'association est généralement liée à l'utilisation de mesures permettant d'estimer la qualité d'une association. Anne [12] en recense plusieurs principales :

- la *couverture* : qui est la proportion des exemples de la base qui vérifient l'antécédent de l'association,
- le *support* : qui est la proportion des exemples qui vérifient à la fois l'antécédent et le conséquent,
- la *confiance* : qui est la proportion des exemples vérifiant l'antécédent qui vérifient en plus le conséquent. Cette valeur peut être considérée comme la probabilité conditionnelle pour un exemple de vérifier le conséquent sachant qu'il vérifie l'antécédent.

Il existe plusieurs algorithmes pour découvrir des règles d'association dans une base de données. Comme dans ces approches, Anne a proposé une méthode de génération de résumés flous à partir d'une base de données multidimensionnelle[1] qui s'effectue en deux étapes.

Dans un premier temps, on recense toutes les combinaisons de valeurs d'attributs qui apparaissent de façon fréquente dans la base. La notion "fréquente" est évaluée en comparant la proportion des exemples qui vérifient les valeurs correspondant à un seuil fixé par l'utilisateur (on mesure généralement le support de ces combinaisons). Cette étape est réalisée par paliers : d'abord les valeurs seules sont considérées, puis les combinaisons de deux valeurs, etc.

Dans un deuxième temps, on génère les règles à partir des combinaisons trouvées à l'étape précédente. Pour chaque combinaison, toutes les règles possibles sont étudiées, sauf celles qui amèneraient un conséquent vide. Chacune de ces règles est évaluée par la mesure de confiance.

[1] Dans sa thèse, Anne a introduit la théorie des sous-ensembles flous pour étendre le modèle OLAP (On Line Analytical Processing) de gestion de bases de données multidimensionnelles [72].

Ensuite, on utilise un ensemble de quantificateurs fournis par l'utilisateur : pour chaque quantificateur, on évalue le degré d'appartenance du résumé au quantificateur à l'aide de sa fonction caractéristique. Le quantificateur pour lequel le degré d'appartenance est le plus élevé sera celui qui sera utilisé pour quantifier le résumé.

Anne a proposé une description plus formelle de la construction de résumés flous (voir [72] ou [12] par exemple). Il en ressort que la mesure principale, le support d'un résumé, s'évalue par la probabilité conditionnelle d'avoir le conséquent sachant que l'antécédent est vrai :

$$conf(A \rightarrow C) = \frac{support(A\ and\ C)}{support(A)} = P(C|A),$$

où $support(E)$ est le nombre d'exemples (cardinal) qui vérifient la propriété E.

Pour chaque quantificateur Q souhaité, le degré de vérité τ_Q du résumé flou Q *objets qui sont A sont C* est donné par $\tau_Q = \mu_Q(conf(L \rightarrow R))$.

3.2.3 Similarités entre ces approches

Je vais maintenant présenter ici les similarités et les différences de ces deux approches de construction de règles floues que nous avons fait ressortir avec Anne Laurent et Bernadette Bouchon-Meunier dans nos travaux communs [12]. Pour mettre en évidence les points de comparaison de nos approches, nous avons mis en parallèle non seulement les méthodes, mais aussi leurs buts. Je vais commencer par reprendre les points saillants de nos deux approches, puis je détaillerai la comparaison que l'on en a fait.

Arbres de décision flous

Ce qui est, avant tout, demandé à un tel algorithme de construction de règles, c'est de générer des bases de règles ayant le plus haut pouvoir de prédiction possible.

Dans ce but, il semble alors très intéressant de doter un système automatique de la faculté de sélectionner les exemples d'apprentissage à prendre en compte pour construire un arbre de décision. Tout en gardant en tête bien sûr, le fait que le modèle construit au final devra toujours couvrir l'ensemble des exemples d'apprentissage afin de garantir son plein pouvoir de généralisation.

On se place ici dans le cadre de l'apprentissage dit *supervisé* car chaque exemple de la base d'apprentissage est associé à une classe que l'on souhaite prédire (ou caractériser) à partir d'un ensemble d'attributs donnés.

Les algorithmes de construction d'arbres de décision flous sont évalués et comparés les uns par rapport aux autres selon leur performance pour la classification d'exemples d'une base de test. On évalue alors leur taux d'erreur lors de cette reconnaissance.

Résumés flous

Le but des résumés flous est de décrire les données, soit pour valider une hypothèse formulée par l'utilisateur, soit pour mettre en évidence des connaissances pertinentes à partir des données. De par leur nature, les résumés ne sont pas censés couvrir tout l'espace des données et toutes les possibilités qui en découlent. Ainsi, contrairement aux arbres de décision flous, un ensemble de résumés flous n'est pas forcément performant pour évaluer un nouvel exemple, en particulier, s'ils ont été construits à partir d'une base de données qui n'est pas suffisamment exhaustive.

D'autre part, la méthode de construction de façon automatique des résumés flous est une méthode non supervisée car généralement aucune classe à reconnaître en particulier n'a été fixée. La qualité des règles ainsi générées est évaluée à l'aide de leur support et de leur mesure de confiance. Des seuils dans ces valeurs, fixés par l'utilisateur, permettent de réduire le nombre de règles.

Comparaison des deux méthodes

Au premier abord, les méthodes de construction d'arbres de décision flous et des résumés flous sont difficilement comparables voire complètement différentes. En effet, la première est une méthode supervisée, et l'autre est une approche plutôt non supervisée.

Ensuite, on peut aussi remarquer que ces deux approches utilisent un comptage des exemples, à travers l'utilisation de probabilités conditionnelles.

La Table 3.1, issue de notre article [12], met en parallèle ces deux approches et fait ressortir les points de comparaison possibles qui peuvent être mis en évidence.

Ainsi, on peut donc mieux voir ressortir les éléments en adéquation dans ces deux méthodes. De façon plus intéressante aussi, on peut voir apparaître les grandes différences entre les deux approches et les différences

3.2.4 Améliorer l'interprétabilité

Après avoir mis en évidence des points de comparaison possibles entre ces deux approches, nous avons étudié les moyens d'utiliser les points forts d'une des approches pour améliorer l'autre.

En particulier, nous avons essayé de faire ressortir les moyens d'augmenter l'interprétabilité des bases de règles construites. En fait, il existe différentes façons d'exporter des propriétés d'une approche vers l'autre, ce qui permet d'enrichir chacune de ces méthodes.

Apports des résumés flous pour les arbres de décision flous

Les résumés flous sont très intéressants parce qu'ils expriment une connaissance linguistique en langage naturel. En plus, l'utilisation de quantificateurs flous afin d'exprimer la confiance dans une règle est une des caractéristiques importantes qu'il serait très intéressant de généraliser à d'autres approches afin de pouvoir en améliorer l'interprétabilité.

Quantification de la classification. L'utilisation des quantificateurs, sur le modèle de leur utilisation dans les résumés flous que nous avons présentés dans la partie précédente, est tout à fait applicable dans le cadre des arbres de décision flous.

Par exemple, on peut les utiliser pour l'expression des proportions des classes obtenues dans les feuilles de l'arbre construit. Le degré d'impureté de la feuille, estimé par les fréquences de chaque classe présente dans la feuille, est utilisable pour choisir, comme on l'a vu dans la Section 3.2.2, le quantificateur le plus adéquat qualifiant la proportion des exemples possédant cette classe.

Ainsi, dans un arbre de décision flou, une feuille est étiquetée par un ensemble de classes $\{c_1, ..., c_K\}$. À chacune de ces classes c_j est associé un poids $P^*(c_j|(v_{l_1}, ..., v_{l_p}))$. Ce poids est calculé à l'aide de la probabilité d'appartenir à la classe c_j pour un exemple de la base d'apprentissage possédant les valeurs $(v_{l_1}, ..., v_{l_p})$, pondérée par son degré d'appartenance à la feuille.

	Arbres de décision flous	**Résumés flous**
Domaine	machine learning	base de données
Type	supervisé	non supervisé (en général)
Prise en compte des données	base d'apprentissage vs base de test	tout l'ensemble
Critères de construction	entropie	support
Objectif principal	prédiction couvrir tous les exemples généralisation	description fournir des règles intéressantes pertinence, utilisabilité
Degrés de qualité		
Support	éventuellement comme critère d'arrêt	critère de qualité et de construction
Couverture	-	mesure de qualité
Confiance	pureté des feuilles	mesure de qualité
Appropriateness	-	mesure de qualité
Imprécision	-	mesure de qualité
Longueur	optimisée par construction	mesure de qualité
Précision/taux d'erreur	à optimiser	-
Rappel	à optimiser	-
Complexité/taille	à optimiser / optimisé par construction	réduit à l'aide du support
Interprétabilité	bonne	bonne
Quantificateurs	-	utilisés dans l'expression de la confiance
Agrégation	utilisée pour la fusion des règles	- -
Type de règles	conséquent unique (classe)	tout type

Table 3.1 – Comparatif des deux approches (extrait de [12])

On peut remarquer ici qu'un tel poids n'a pas vraiment de sens dans le cadre des arbres de décision classiques où les valeurs d'apprentissage ne sont pas floues et où les exemples appartiennent à une et une seule feuille de l'arbre. Dans ce cas-là, on ne peut avoir que $P^*(c_j|(v_{l_1},...,v_{l_p}))$ égal à 1 pour chaque c_j.

Ainsi, on sait que, dans un arbre de décision flou, une branche est équivalente à une règle de la forme [8] : si $A_{l_1} = v_{l_1}$ et ... et $A_{l_p} = v_{l_p}$ alors $C = c_1$ avec le degré $P^*(c_1|(v_{l_1},...,v_{l_p}))$ et ...$C = c_K$ avec le degré $P^*(c_K|(v_{l_1},...,v_{l_p}))$.

Le poids $P^*(c_j|(v_{l_1},...v_{l_p}))$ peut être considéré comme la *force* de l'association de l'exemple à la classe c_j selon cette règle. Plus ce poids est proche de 1 et plus la confiance que l'exemple soit associé à la classe c_j par cette règle est forte. De plus, ce poids appartient donc à l'intervalle $[0, 1]$ et, par conséquent, comme ce qui est fait dans les résumés flous, on peut alors utiliser un quantificateur flou (comme ceux donnés dans la Figure 3.1) pour quantifier l'appartenance d'un exemple à la classe par ce chemin dans l'arbre. Une règle quantifiée peut donc être déduite à partir d'une branche de l'arbre :

$$Q\ (A_{l_1} = v_{l_1} \text{ et } ...\text{et } A_{l_p} = v_{l_p})\ sont\ C = c_j : \tau.$$

en prenant :

$$\tau = \mu_Q \left(P^*(c_j|(v_{l_1}, ...v_{l_p})) \right) \tag{3.2}$$

Le support comme critère d'arrêt. Une autre possibilité d'introduire une propriété de la construction des résumés flous pour la construction des arbres de décision est de tenir compte du nombre d'exemples comme un critère d'arrêt, à l'issue de chaque partition de la base d'apprentissage. On peut alors garantir qu'un nombre suffisant d'exemples de la base d'apprentissage est conservé pour le développement d'une branche de l'arbre, ce qui garantit donc un support suffisant à la règle déduite de cette branche pour en assurer une bonne légitimité.

Un tel critère d'arrêt a déjà été introduit dans la construction des arbres de décision flous [79], mais on obtient ici, en plus, un lien avec la notion de support qui n'existe pas dans le cadre des arbres mais qui est fortement utilisée dans les règles d'association. Cela offre un excellent moyen de légitimer un tel critère d'arrêt pour la construction des arbres (au moins des arbres flous).

Algorithme 4 Construction d'un arbre de décision flou complexe

Input :
Base d'apprentissage $\mathcal{E}$,
Ensemble d'attributs $\mathcal{A}$= $\{A_1, ..., A_{N+1}\}$, avec $N > 0$
k un entier tel que $1 \leq k \leq N$,
T un critère d'arrêt,
P une stratégie de partitionnement,
H une mesure de discrimination,
Output : $\mathcal{T}$ un arbre de décision flou
Begin
$\mathcal{C} \leftarrow A_{\rho_1} \times ... \times A_{\rho_k}$, sélection de k attributs $A_{\rho_1}, ..., A_{\rho_k}$ de $\mathcal{A}$,
$\mathcal{T} \leftarrow$ construction par l'Algorithme 3 avec $\mathcal{E}$, $\mathcal{A}/\mathcal{C}$, $\mathcal{C}$, T, H, et P comme paramètres.
End.

Du non-supervisé au supervisé. Une troisième possibilité d'apport des résumés flous dans la méthodologie des arbres de décision flous est de s'en inspirer pour la construction d'arbres de décision flous lorsqu'aucune classe n'est définie dans la base d'apprentissage. Ainsi, à partir d'un problème d'apprentissage non supervisé, la méthode appliquée aux résumés flous permet de construire aussi des arbres de décision. L'idée est de sélectionner, au choix ou aléatoirement, une classe parmi les attributs disponibles avant la construction de l'arbre. À partir de cela, plusieurs arbres de décision peuvent être construits, et comparés par des mesures d'évaluation afin de pouvoir sélectionner ceux qui sont les plus pertinents à utiliser.

A fortiori, on peut même alors dans ce cas-là décider de retenir, non pas un seul attribut, mais un sous-ensemble d'attributs dont les valeurs, obtenues par un produit cartésien des ensembles de définition de ces attributs, peuvent alors être utilisées comme classes dans la construction de l'arbre de décision (voir Algorithme 4).

Algorithme 5 Construction d'un arbre de décision flou complexe (version 2)

Input :
Base d'apprentissage $\mathcal{E}$,
Ensemble d'attributs $\mathcal{A}$= $\{A_1, ..., A_{N+1}\}$, avec $N > 0$
k un entier tel que $1 \leq k \leq N$,
T un critère d'arrêt,
P une stratégie de partitionnement,
H une mesure de discrimination,
Output : $\mathcal{T}$ un arbre de décision flou
Begin
$\mathcal{C}\leftarrow$ choix d'un attribut parmi $A_{\rho_1}, ..., A_{\rho_k}$ de $\mathcal{A}$, tels que $A_{\rho_1}, ..., A_{\rho_k}$ soient les attributs qui apparaissent dans les résumés flous ayant la confiance la plus élevée, construits à partir de $\mathcal{E}$,
$\mathcal{T} \leftarrow$ construction par Algorithme 3 avec $\mathcal{E}$, $\mathcal{A}/\mathcal{C}$, $\mathcal{C}$, T, H, et P comme paramètres.
End.

Filtrage d'attributs. Finalement, les résumés flous peuvent être utilisés dans une étape préliminaire pour la construction d'un arbre de décision, afin de présélectionner des attributs pouvant être utilisés comme classe. Ainsi, la classe peut être choisie parmi les valeurs des attributs qui offrent les résumés ayant la confiance la plus élevée (voir Algorithme 5).

Quantifier les arbres. Les résumés flous peuvent servir à améliorer l'interprétation des arbres de décision flous.

Tout d'abord, on peut remarquer qu'il est aisé de convertir un arbre flou en un ensemble de résumés flous.

D'autre part, la mesure de discrimination (comme, par exemple, l'entropie d'événements flous) peut être représentée à l'aide de quantificateurs flous, de la même façon que ce qui a été présenté dans la Section 3.2.3. Ainsi, par exemple, la valeur de $H(C|A_j)$, une fois normalisée dans l'intervalle $[0, 1]$, peut être exprimée à l'aide de quantificateurs du type de ceux qui sont donnés dans la Figure 3.1.

On peut donc alors obtenir une expression floue pour caractériser le pouvoir de discrimination d'un attribut ou la présence d'une classe dans une feuille en fonction de la valeur d'entropie qui lui est associée. De plus, cette quantification est alors utilisable durant le processus de construction afin d'offrir des informations interprétables à l'utilisateur (qui a donc alors la possibilité de choisir lui-même l'attribut à retenir lors de la construction d'un nœud de l'arbre).

Apports des arbres de décision flous pour la génération de résumés

Résumés multi-valeurs. Le point clé de l'algorithme de construction d'un arbre de décision réside dans l'utilisation d'une mesure de discrimination pour ordonner les attributs en fonction de leur pouvoir discriminant vis-à-vis de la classe. Cet ordonnancement a la particularité de prendre en compte simultanément la distribution de toutes les valeurs d'un attribut. Il n'y a pas d'équivalent dans la construction de résumés flous, les attributs ne sont pas examinés globalement sur leur ensemble de valeurs, mais leurs valeurs sont étudiées de façon indépendante les unes des autres.

Un premier apport intéressant peut donc être d'utiliser une mesure de discrimination afin de sélectionner un ensemble d'attributs pertinents dont les valeurs pourront éventuellement

servir à générer des résumés. Par contre, il est alors nécessaire, pour la mesure d'un pouvoir discriminant, de connaître une classe à reconnaître. Si elle n'est pas fournie au préalable, une telle classe peut être constituée, dans ce cadre, par un des attributs existants, ou bien une combinaison d'attributs.

Ainsi, la mesure du pouvoir de discrimination permet d'ordonner les attributs et de pouvoir sélectionner ceux pour lesquels l'algorithme de résumés flous est susceptible de donner les meilleurs résultats (ou du moins, les résultats les plus ou les mieux exploitables).

De la même manière, le gain d'information d'un attribut A_j, $I(A_j, C) = H(C) - H(C|A_j)$, qui est utilisé pour sélectionner les attributs dans la construction d'un arbre de décision, peut aussi être très utile pour pré-sélectionner des attributs pertinents pour la génération de résumés flous.

Ordonnancement d'attributs. Dans le cas où aucune classe n'est fournie ou n'est pas susceptible d'être trouvée parmi les attributs, alors les attributs peuvent quand même être ordonnés en utilisant une mesure d'entropie simple comme $H_S(A_j)$. Cette mesure est minimale si l'attribut n'a qu'une seule valeur et, au contraire, sera très élevée si toutes ses valeurs sont uniformément distribuées. Ainsi, un attribut est d'autant plus pertinent que sa mesure $H_S(A_j)$ est proche de 0 (mais surtout pas nulle) car dans ce cas, cela voudra dire qu'une seule de ses valeurs pourra amener des résumés flous notoirement connus (même avec un support très élevé).

Fusion de règles. La taille d'une base de règles peut être réduite en utilisant des méthodes de fusion et de simplification des règles. On peut élaborer deux types de méthodes, les méthodes "*intra-règle*" ou bien des méthodes "*inter-règle*" selon que :

- le même attribut apparaît plusieurs fois dans la prémisse d'une règle, mais pour des valeurs ou plages de valeurs différentes,
- plusieurs règles sont très "proches".

Comme on l'a décrit dans [45], les règles d'une base de règles floues peuvent être fusionnées quand le même attribut apparaît plusieurs fois dans la prémisse avec des valeurs floues différentes ou proches. Cela se produit fréquemment quand la base de règles est obtenue par l'intermédiaire de certains algorithmes de construction automatique (tel celui des arbres de décision flous).

Le second cas se produit, lui, quand deux règles d'une même base sont suffisamment "proches" pour pouvoir être fusionnées en une seule. Plusieurs façons de fusionner pour réaliser cela ont été étudiées dans [10].

Ces deux types de méthodes sont facilement adaptables et utilisables dans le cadre des résumés flous afin de permettre une réduction du nombre de résumés flous générés.

3.2.5 *So what*

Cette étude que nous avons menée avec Anne et Bernadette a permis de faire ressortir les points communs existants entre ces trois approches (arbres de décision flous, résumés flous, et modificateurs linguistiques). À partir de cette mise en parallèle, les similitudes conceptuelles des algorithmes de construction d'arbres de décision flous et de ceux de construction de résumés flous ont pu être mises en évidence. De plus, des ajustements ont pu être proposés afin de faire bénéficier chacune des méthodes des avantages des autres méthodes (interprétabilité, traitement conjoint des attributs, traitement des classes,...).

Ces travaux restent encore préliminaires, et demanderaient à être développés plus avant afin d'en obtenir une unification plus globale de ces approches.

3.3 Raisonnement interpolatif

En tant que mode de raisonnement approximatif, le raisonnement interpolatif (ou, plus simplement, interpolation) a été beaucoup étudié ces dernières années [6, 7, 44, 67, 68, 69, 98].

En collaboration avec Bernadette Bouchon-Meunier, Maria Rifqi, et d'autres chercheurs de notre équipe, nous avons étudié le raisonnement interpolatif applicable sur une base de règles floues "incomplètes" (*ie.* dont les prémisses ne couvrent pas tout l'espace des valeurs d'entrées). Cela nous a amenés à proposer une nouvelle méthode de raisonnement interpolatif.

Dans cette partie, je vais présenter brièvement l'approche de raisonnement interpolatif que nous avons proposée, et je détaillerai un peu plus le traitement des règles multi-prémisses [49].

3.3.1 Gradualité et analogie

On considère donnée une règle de production de la forme :

$$\texttt{(R)} : \text{si } U \text{ est } A \text{ alors } V \text{ est } B,$$

pour laquelle U et V sont des variables linguistiques qui peuvent prendre des valeurs numériques ou floues.

A partir de ce type de règles, le Modus Ponens classique demande l'observation *exacte* de la valeur A pour U afin d'en déduire la valeur B pour V. Par contre, en raisonnement approximatif, on accepte de ne pas observer *exactement* A pour déduire une conclusion avec cette règle `(R)`. Ainsi, si on observe une valeur A^* qui est une approximation de A, on souhaite pouvoir utiliser `(R)` afin de pouvoir en déduire une valeur B^* qui puisse être aussi, éventuellement, une approximation de B. On en arrive alors à souhaiter disposer d'un Modus Ponens *généralisé* autorisant ce mode de raisonnement. On parle de généralisation du Modus Ponens car, bien entendu, on souhaite conserver le raisonnement par Modus Ponens usuel si on n'observe pas d'approximation mais les prémisses exactes.

Un mode de raisonnement approximatif particulier est le raisonnement interpolatif (on dira aussi "*interpolation*"). Dans ce modèle, on considère données deux règles de production sur les mêmes variables linguistique U et V :

$$\texttt{(R}_1\texttt{)} : \text{si } U \text{ est } A_1 \text{ alors } V \text{ est } B_1$$

$$\texttt{(R}_2\texttt{)} : \text{si } U \text{ est } A_2 \text{ alors } V \text{ est } B_2$$

De plus, on considère que les variables linguistiques U et V sont définies sur un univers de valeurs ordonné. Et, d'autre part, les valeurs A_1 et A_2 de U peuvent être séparées par un ensemble de valeurs n'entrant pas en jeu dans les 2 règles données.

En raisonnement interpolatif, on souhaite alors, étant donnée une valeur observée A pour la variable U, déterminer la meilleure valeur B correspondante pour V. On admet alors que la valeur A puisse être :

- soit la valeur A_1, dans ce cas la valeur correspondante pour V serait B_1,
- soit la valeur A_2, dans ce cas la valeur correspondante pour V serait B_2,
- mais aussi, n'importe quelle valeur (précise ou floue) A *comprise*[2] entre A_1 et A_2, et, dans ce cas, la solution B recherchée pour V pourra donc être aussi une valeur comprise entre B_1 et B_2.

[2] En théorie des sous-ensembles flous, cette propriété n'est pas si évidente à caractériser.

Notre approche pour la constitution d'un tel mécanisme de raisonnement a été développée en se basant sur deux hypothèses : la gradualité et l'analogie [59, 20, 52].

La gradualité suppose que les variables de ce type de règles ont un comportement graduel, qu'elles se trouvent en prémisse ou en conclusion de la règle. Ainsi, par exemple, on considère que la valeur de V augmente quand la valeur de U augmente (ou décroît selon le type de gradualité requise).

De plus, nous considérons que la connaissance dont on dispose sur V est analogue à celle que l'on possède sur U, du point de vue de la fiabilité et de l'incertitude. Cela sous-entend que l'on s'attend donc à ce que toute précision d'information sur U amène une précision sur l'information sur V.

Pour mettre en œuvre cela, il y a plusieurs possibilités. On peut considérer :

- que la précision sur B^* ne doit dépendre que de la précision que l'on a sur A^*. Ainsi, si A^* est une valeur précise, alors B^* devra aussi être une valeur précise, et, si A^* est une valeur très imprécise, alors B^* sera aussi très imprécise.
- ou alors que la précision sur B^* ne dépend pas seulement de la précision que l'on a sur A^* mais aussi sur la précision même des valeurs A et B qui se trouvent dans la règle.

Dans notre modèle, nous avons choisi la seconde possibilité, mais il faut garder en mémoire que ce n'est qu'un choix possible et que d'autres seraient tout aussi pertinents. On peut aussi noter qu'un point important, qui a aussi conditionné nos travaux, est le souhait de rester compatible avec l'interpolation linéaire classique.

Dans notre approche, nous avons proposé de déterminer le sous-ensemble flou B^* à l'aide de ses deux composantes essentielles : sa *localisation* dans l'univers de définition de V, et sa *forme* (valeur précise, sous-ensemble flou triangulaire, trapézoïdal, etc.). Dans ce cadre, la détermination de la position de B^* peut s'effectuer d'une manière similaire à l'interpolation linéaire classique. Par contre, la forme de B^* est plus difficile à déterminer et nous avons donc proposé d'utiliser des formes simples de fonctions d'appartenance définies par un minimum de paramètres. Nous avons alors introduit une méthode qui, à partir de la comparaison de la forme de A^* avec les formes des prémisses A_1 et A_2, et des formes de B_1 et B_2, permet de déduire une forme pour B^*.

Dans ce qui suit, je vais commencer par rappeler brièvement la méthode d'interpolation "simple", quand il n'y a qu'un seul attribut en prémisse, puis je détaillerais notre approche prenant en compte des règles multi-prémisses.

3.3.2 L'interpolation avec un seul attribut en prémisse

On considère ici le cas où l'on possède une base de m règles et une observation :

$(\mathtt{R}_1)$: si U est A_1 alors V est B_1

...

$(\mathtt{R}_m)$: si U est A_m alors V est B_m

(Observation) : U est A.

Je rappelle que U et V sont des variables linguistiques. Les valeurs (éventuellement floues) A_1, ..., A_m prises par U sont définies sur un univers ordonné (typiquement, sur $\mathbb{R}$). Il en est de même pour les valeurs (éventuellement floues aussi) B_1, ..., B_m prises par V. Dans ce qui suit, on notera $\mathbf{F}$ l'ensemble de tous les sous-ensembles flous possibles de $\mathbb{R}$.

Position d'un sous-ensemble flou

Les modèles que nous avons proposés [59, 20] utilisent tous une *localisation* $l(f)$ de sous-ensembles flous f (de $\mathbb{R}$). Cette localisation représente le positionnement de f dans $\mathbb{R}$. Il y a évidemment plusieurs façons de définir une localisation de sous-ensembles flous. Dans notre approche, nous nous sommes focalisés sur des sous-ensembles flous triangulaires ou trapézoïdaux, normalisés, et nous avons considéré la définition suivante :

Définition 6 (localisation d'un sous-ensemble flous) *La localisation $l(f)$ du sous-ensemble flou f de $\mathbf{F}$ est l'abscisse $x_M \in \mathbb{R}$ qui correspond au milieu du noyau de f.*

D'autres choix sont possibles ; en particulier, on pourrait utiliser le centre de gravité ou le milieu du support de f pour le localiser [49].

Cette mesure de position de sous-ensembles flous est très importante car elle permet alors de définir une relation d'ordre entre éléments de $\mathbf{F}$ (l'ordre entre éléments de $\mathbf{F}$ correspondant à un ordre sur les localisations, éléments de $\mathbb{R}$).

Définition 7 (Ordre sur F) *Soit f_1 et f_2 deux sous-ensembles flous de $\mathbf{F}$, on a $f_1 \preceq f_2$ si et seulement si $l(f_1) \leq l(f_2)$.*

L'interpolation comme un processus analogique

Dans notre approche, nous avons choisi de fonder le choix de B sur une *approche analogique* [59, 19, 20, 52]. Ainsi, pour tout A, si on peut trouver A_i et A_{i+1} tels que $A_i \preceq A \preceq A_{i+1}$, il faut alors étudier les relations existantes entre A et les deux valeurs A_i et A_{i+1}, afin d'en déduire des relations similaires entre B et les deux valeurs B_i et B_{i+1}.

Ensuite, l'idée de base d'un tel raisonnement par analogie est de considérer que "B est relié à (B_i, B_{i+1}) de la même façon que A est relié à (A_i, A_{i+1})". Ainsi, le résultat de la comparaison de B avec (B_i, B_{i+1}) doit donc être identique au résultat de la comparaison de A avec (A_i, A_{i+1}).

Dans notre cadre, cette comparaison doit donc être liée à :

- la localisation : la localisation de B entre B_i et B_{i+1} doit être équivalente à celle de A entre A_i et A_{i+1},
- la forme : les différences entre la forme de B et les formes de B_i et B_{i+1} doivent être équivalentes aux différences existantes entre la forme de A et celles de A_i et A_{i+1}.

Selon cette hypothèse de comportement, on procède comme suit pour déterminer la conclusion B caractérisant le mieux V étant donnée l'observation A caractérisant U [20, 52].

Étape 1 : Déterminer l'index i, $1 \leq i \leq m$, tel que $A_i \preceq A \preceq A_{i+1}$.

Étape 2 : Comparer A avec les éléments du couple (A_i, A_{i+1}).

Étape 3 : Construire les éléments de $\mathbf{F}$ qui ressemblent aux éléments du couple $(B_i,\ B_{i+1})$ de la même façon que A ressemble aux éléments de (A_i, A_{i+1}).

Étape 4 : Choisir un de ces éléments de $\mathbf{F}$ comme valeur pour B.

Dans ce qui suit, je vais décrire plus finement ces différentes étapes de la détermination de B. Pour de plus amples détails sur ces points, je renvoie encore à nos publications sur le sujet.

Détermination de A_i et de A_{i+1}

Étant donné A, la détermination du couple $(A_i,\ A_{i+1})$ est faite grâce à l'ordre sur les A_j induit par leurs localisations.

Comparaison de A avec (A_i, A_{i+1})

La comparaison de A avec (A_i, A_{i+1}) est faite sous deux aspects :
- la localisation de A entre A_i et A_{i+1},
- une mesure de la distinguabilité qui existe entre la forme de A et les formes de A_i et A_{i+1}.

Construction de la solution B

Dans notre approche, nous avons décidé de considérer les transformations appliquées sur les sous-ensembles flous A_i, et A_{i+1} afin d'obtenir l'observation A. Ces transformations sont comparées pour une même localisation des sous-ensembles flous (la localisation ne doit pas interférer avec cette étude basée essentiellement sur la forme des sous-ensembles flous).

Partant de là, nous reportons ces transformations sur B_i et sur B_{i+1} et nous agrégeons les deux formes obtenues afin d'en déduire la forme de la solution B.

Après ce rappel sur notre approche de raisonnement interpolatif en présence de prémisses simples, je vais présenter plus en détail l'extension de ce modèle au cas multi-prémisses.

3.3.3 L'interpolation avec des règles multi-prémisses

De façon plus générale, les règles à traiter sont composées par des prémisses multiples, et non pas simples. Plusieurs attributs sont présents dans ces prémisses, et plus seulement un seul attribut.

Dans la littérature, très peu de travaux s'attaquent au problème du traitement des prémisses multiples en raisonnement interpolatif [8, 34, 67, 108].

Nous avons donc été amenés à étendre notre propre approche de raisonnement interpolatif décrite dans la partie précédente, en la généralisant d'un espace à une dimension, à un espace à m dimensions ($m > 1$).

On considère toujours ici une base de m règles et une observation :.

(R$_1$) : si U est A_1 alors V est B_1

...

(R$_m$) : si U est A_m alors V est B_m

(Observation) : U est A.

Mais, dans cette partie, on va maintenant considérer que la variable U est composée (par produit cartésien) d'un ensemble de n variables (numériques) $U_1, U_2, ..., U_n$ avec chaque U_j définie sur un sous-ensemble X_j de $\mathbb{R}$, pour tout $j = 1, ..., n$. Dans la suite, on notera X le produit cartésien des X_j, $j = 1, ..., n$. On suppose de plus que chaque description A_i, $i = 1, ..., m$, est composée d'un ensemble de sous-ensembles flous $a_{i1}, a_{i2}, ..., a_{in}$, où a_{ij} est un sous-ensemble flou sur X_j. De même, la variable (numérique) V sera définie sur Y, sous-ensemble de $\mathbb{R}$, et les B_i, $i = 1, ..., m$, sont des sous-ensembles flous de Y.

Ainsi, étant donnée une description A, composée d'un ensemble de sous-ensemble flous de $\mathbb{R}$: $a_1, a_2, ..., a_n$, il faut trouver un sous-ensemble flou B de $\mathbb{R}$ qui puisse être associé à A selon la base de règles.

Dans ce qui suit, on appellera *description*, un ensemble de n sous-ensembles flous et on notera encore $\mathbf{F}$ l'ensemble de tous les sous-ensembles flous de $\mathbb{R}$.

Remarque préliminaire

En fait, dans un tel espace à m dimensions, il existe différentes possibilités de correspondance[3] entre les attributs de l'observation et les attributs présents en prémisse d'une règle. En effet, l'observation peut correspondre complètement, partiellement, ou de façon incomplète à la prémisse de la règle. Mais aussi, seule une partie de la description de l'observation peut correspondre à la prémisse, et aucune correspondance peut n'être possible.

J'illustre ces possibilités dans la Table 3.2 pour le cas simple où les descriptions sont dans un espace à 3 dimensions.

Type de correspondance	**Attributs**	
	prémisse	**observation**
Complète	a_1, a_2, a_3	a_1, a_2, a_3
Partielle	a_1, a_2	a_1, a_2, a_3
Intersection	a_1, a_2	a_2, a_3
Incomplète	a_1, a_2	a_1
Vide	a_1	a_3

Table 3.2 – Possibilités de correspondances règle/observation

Le type de correspondance "Vide" est assez trivial : la règle ne s'applique pas pour cette observation, de même, pour le type "Incomplète" qui exclut donc l'utilisation de cette règle pour l'observation donnée. On peut considérer qu'il en est ainsi pour le type "Intersection" qui se ramène donc au cas où la règle ne peut pas être utilisée pour cette observation. Le type "Partielle" est lui aussi assez simple à traiter car il se ramène au type "Complète". Le cas "Complète" est celui qui nous intéresse donc ici.

Dans le cadre du raisonnement interpolatif, il faut considérer non plus une unique règle, mais plusieurs règles susceptibles de s'appliquer pour une même observation. La correspondance de l'observation à toutes ces règles se doit donc d'être soit de type "Complète", soit de type "Partielle".

Le cas d'une correspondance "Complète" est trivial à régler, c'est celui qui est le plus simple et l'application de la règle sera automatique. Par contre, si certaines règles ne correspondent que partiellement à l'observation, cela peut entraîner quelques difficultés pour mettre en œuvre un raisonnement.

On considérera dans la suite que, dans notre modèle, les prémisses de la base de règles que l'on utilise ont été *augmentées* de telle sorte que la correspondance de l'observation aux règles soit toujours complète. Si nécessaire, une règle partiellement couverte sera remplacée par un ensemble de règles complètement couvertes en utilisant des valeurs couvrant l'univers total des attributs non utilisés.

Position d'une description

Comme dans la partie mono-prémisse, on a toujours besoin de pouvoir localiser une description parmi l'ensemble des descriptions qui sont présentes dans les règles. Par contre, ici, cette localisation s'effectue dans un espace défini par l'ensemble de tous les univers des sous-ensembles qui sont présents dans toutes les règles de la base.

[3] Dans ce cadre, on utilise le terme "matching".

Définition 8 (localisation d'une description) *La localisation $L(F)$ d'une description composée par la conjonction de n sous-ensembles flous f_j, $j = 1, ..., n$ de $\mathbb{R}$ est définie par : $L(F) = (l(f_1), l(f_2), ..., l(f_n))$. On a donc $L(F) \in \mathbb{R}^n$.*

Ordonnancement des descriptions

À partir de la localisation, on peut donc définir un ordre $\preceq$ sur les descriptions, mais pour cela, il est nécessaire de choisir une métrique sur l'espace $\mathbb{R}^n$ où sont définies ces descriptions. La distance classique sera utilisée dans ce sens.

On considère que $\mathbb{R}^n$ est muni d'un repère centré sur $O = (0, ..., 0)$. Dans ce qui suit, on notera : $d : \mathbb{R}^n \longrightarrow \mathbb{R}$ telle que, pour tout $M = (x_1, ..., x_n)$, $d(M) = \|\overrightarrow{OM}\| = \sqrt{\sum_{k=1}^{n} x_k^2}$.

Définition 9 (Ordre sur les descriptions) *Soient A_j et A_k deux descriptions. On dira que $A_j \preceq A_k$ si et seulement si $d(L(A_j)) \leq d(L(A_k))$.*

Changement de référentiel

Soit W une variable définie sur un univers de valeurs numériques X_W et soit Z une autre variable définie sur un univers de valeurs numériques X_Z.

Dans ce qui suit, apparaîtra la nécessité de devoir définir comment "basculer" une valeur de l'univers X_W vers l'univers X_Z. On aura alors besoin de définir comment "traduire" une mesure sur un objet de X_W en une mesure sur un objet de X_Z. Ce changement sera lié à l'échelle des valeurs de X_W et à l'échelle des valeurs de X_Z. On définira la *plage des valeurs* de l'univers de valeurs numériques (supposé connexe) X_W par $r(X_W) = a^+ - a^-$, où $[a^-, a^+]$ est l'intervalle des valeurs prises par la variable W. On définit de même X_Z par $r(X_Z) = b^+ - b^-$, avec $[b^-, b^+]$ l'intervalle des valeurs prises par la variable Z.

Définition 10 (Changement de référentiel) *Soient $X_W \subseteq \mathbb{R}$ et $X_Z \subseteq \mathbb{R}$ deux univers de valeurs numériques. La* fonction de changement de référentiel *$R_{X_W X_Z} : X_W \longrightarrow X_Z$ est définie à partir des plages de valeurs des deux univers, pour tout $x \in X_W$, par :*

$$R_{X_W X_Z}(x) = \frac{r(X_Z)}{r(X_W)}\, x.$$

On peut noter que, dans le cas particulier où $X_W = [0, 1]$, on a donc pour tout $x \in X_W$, $R_{[0,1]X_Z}(x) = x\ r(X_Z)$. De même, si on a $X_Z = [0, 1]$, alors, pour tout $x \in X_W$, $R_{X_W[0,1]}(x) = \frac{x}{r(X_W)}$. On notera alors $R_{X_W[0,1]}$ simplement R_{X_W}.

Par souci de simplicité, dans la suite, on notera simplement $(A_1,\ A_2)$ le couple $(A_i,\ A_{i+1})$, et on notera $(B_1,\ B_2)$ le couple des conclusions associées.

Localisation de la description B

La localisation de B dans le contexte de B_1 et B_2 est déterminée par la propre localisation de A dans le contexte de A_1 et A_2.

Ainsi, si on note, pour $i = 1, 2$, $A_i = (a_{i1}, a_{i2}, ..., a_{in})$, on a la localisation de A_i telle qu'on l'a définie précédemment : $L(A_i) = (l(a_{i1}), l(a_{i2}), ..., l(a_{in}))$. Ici, pour alléger un peu les notations, on écrira simplement $L_i = L(A_i)$ pour $i = 1, 2$. On rappelle que L_i est un point de $\mathbb{R}^n$, et par

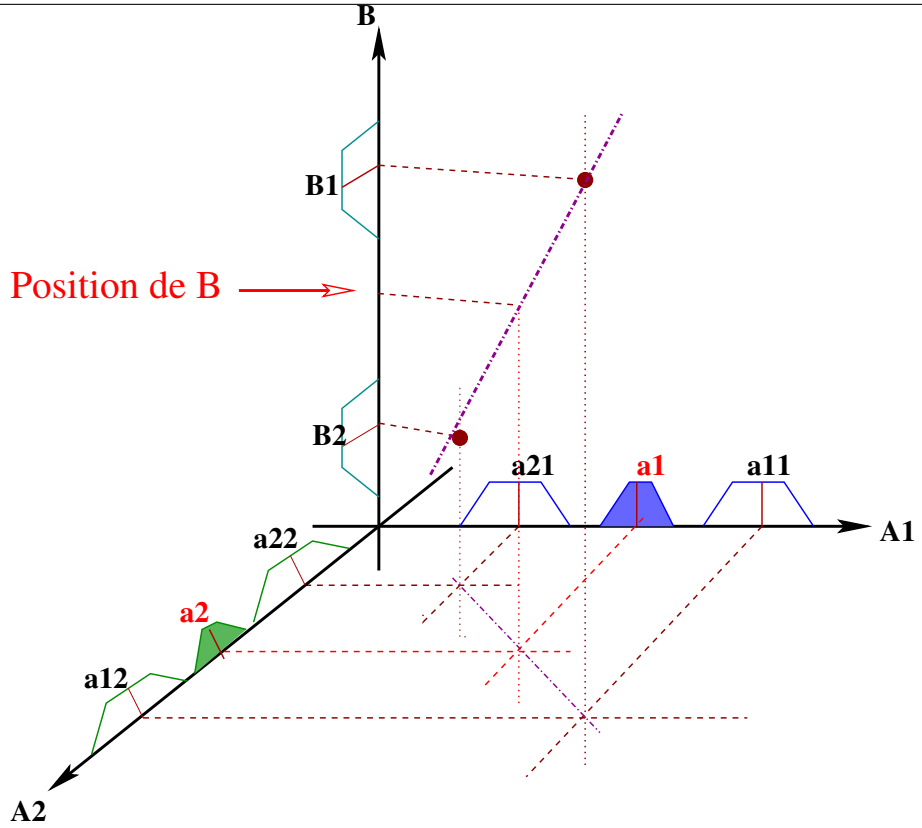

Figure 3.2 – Positionnement de B.

conséquent la localisation de A selon A_i, $i = 1, 2$ peut se définir directement à partir de L_A et de L_i par :

$$L_iL_A = \|\overrightarrow{L_iL_A}\|$$

Ainsi, la localisation de A dans le contexte de A_1 et de A_2 est définie par les trois valeurs : L_1L_A, L_2L_A, et L_1L_2.

Selon les hypothèses de notre modèle de raisonnement interpolatif, la localisation de B dans le contexte de B_1 et de B_2 se doit d'être équivalente à celle de A dans le contexte de A_1 et de A_2.

Ainsi, on a donc :

- $L_{B_1}L_B = R_{XY}(\frac{L_1L_A}{L_1L_A + L_AL_2}\, L_1L_2)$,
- $L_{B_2}L_B = R_{XY}(\frac{L_2L_A}{L_1L_A + L_AL_2}\, L_1L_2)$,

Il est aisé de voir que ces valeurs définissent une position unique pour B entre B_1 et B_2, sur la droite (de $\mathbb{R}$) (B_1, B_2) (par exemple, voir la Figure 3.2).

Une fois la localisation de B déterminée, on peut regarder dans la prochaine étape comment construire la forme de B.

Comparaison de la forme de A avec les formes des A_i

De la même façon que dans la méthode avec une seule prémisse, nous nous intéressons aux différences de formes. Afin de comparer les formes des descriptions, nous allons donc les projeter sur la même localisation. Cela veut dire que, pour comparer deux descriptions, chacune des composantes d'une description sera translatée pour correspondre avec la composante de l'autre.

Soit A'_i, $i = 1, 2$, le translaté de A_i tel que $L(A'_i) = L(A)$, c'est-à-dire tel que tout $j = 1, ..., n$, $l(a'_{ij}) = l(a_j)$. A'_i est obtenu de A_i en translatant a_{ij} sur la localisation de a_j.

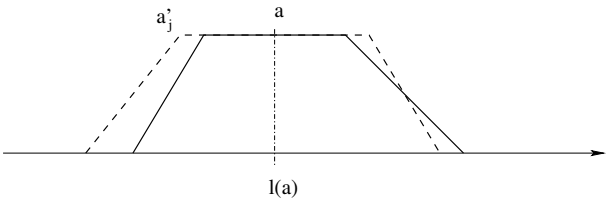

Figure 3.3 – Translaté de A_i sur A

Pour étudier la différence de forme, on considère que A doit être obtenu à partir de A'_i à l'aide d'une transformation mathématique qui s'applique sur la forme de A'_i. Soit $T^i_{sh_X} : \mathbf{F}^n \longrightarrow \mathbf{F}$ une telle transformation de forme qui permette de transformer la forme de A'_i en celle de A. On a donc $A = T^i_{sh_X}(A'_i)$.

Si on considère qu'une description de X est un vecteur de n composantes, la transformation de forme de A'_i à A peut se décomposer en n transformations de formes, chacune s'appliquant sur une des composantes a'_{ik} de A'_i vers la composante correspondante a_k de A. Ainsi, $T^i_{sh_X}$ se décompose en n transformations $T^{ik}_{sh_{X_k}} : \mathbf{F} \longrightarrow \mathbf{F}$ telles que $a_k = T^{ik}_{sh_{X_k}}(a'_{ik})$, $i = 1, ..., n$.

Comme nous l'avons traité dans notre approche mono-prémisse (voir section précédente), dans le cas particulier où v et w sont des sous-ensembles flous de fonction d'appartenance triangulaire ou trapézoïdale, qui peuvent donc se caractériser par quatre points d'inflexion : $[v_1, v_2, v_3, v_4]$ et $[w_1, w_2, w_3, w_4]$, la transformation $T^{ik}_{sh_{X_k}}(v) = w$ peut se caractériser par la seule différence entre les composants de v et de w.

On peut donc ainsi considérer que la transformation $T^{ik}_{sh_{X_k}}$ est un vecteur de transformations locales : $(\delta^k_1, \delta^k_2, \delta^k_3, \delta^k_4)$ telles que $T^{ik}_{sh_{X_k}}(v) = [v_1 + \delta^k_1, v_2 + \delta^k_2, v_3 + \delta^k_3, v_4 + \delta^k_4]$. Et, par conséquent, $T^{ik}_{sh_{X_k}}(v) = w$ permet de définir $\delta^k_j = w_j - v_j$, pour $j = 1, ..., 4$.

Dans le contexte du raisonnement interpolatif, on considère que le changement de forme peut être utilisé pour déduire la forme de B. On doit donc adapter la transformation précédente qui s'applique dans X pour qu'elle puisse s'appliquer sur des éléments de Y.

Cette adaptation s'effectue en considérant que la transformation globale de A'_i en A est définie par :

$$T^i_{sh_X}(A'_i) = \bigoplus_{k=1}^{n} T^{ik}_{sh_{X_k}}(a'_{ik})$$

où $\bigoplus$ est un opérateur d'agrégation que l'on se fixe. Par exemple, on peut choisir comme opérateur d'agrégation la moyenne.

Afin d'être agrégées convenablement, les transformations $T^{ik}_{sh_{X_k}}$, $k = 1, ..., n$, doivent appartenir à la même échelle de valeurs (*ie* normalisée). On adapte donc les transformations par un

processus de normalisation classique :

$$T^i_{sh_X}(A'_i) = \frac{1}{n}\sum_{k=1}^{n} R_{X_k}(T^{ik}_{sh_{X_k}}(a'_{ik}))$$

et $T^i_{sh_X}$ peut s'exprimer comme le vecteur de transformations suivant :

$$(\frac{1}{n}\sum_{k=1}^{n} R_{X_k}(\delta^k_1), \frac{1}{n}\sum_{k=1}^{n} R_{X_k}(\delta^k_2), \frac{1}{n}\sum_{k=1}^{n} R_{X_k}(\delta^k_3), \frac{1}{n}\sum_{k=1}^{n} R_{X_k}(\delta^k_4)).$$

Ainsi donc, à partir de A'_1 et de A'_2 pour atteindre A, nous obtenons deux transformations (normalisées) : $T^1_{sh_X}$ and $T^2_{sh_X}$. A partir de chacune de ces fonctions, il est alors possible de construire une image pour A selon la règle correspondante.

Détermination des formes de B' et de B''

Les deux transformations $T^1_{sh_X}$ et $T^2_{sh_X}$, trouvées dans la partie précédente, vont permettre chacune de construire une conclusion partielle pour A. La conclusion B' pour A doit être similaire à B_1 comme A est similaire à A_1. La conclusion B'' pour A doit être similaire à B_2 comme A est similaire à A_2. Pour rendre compte de cette similarité, on applique la transformation $T^1_{sh_X}$ à B_1 pour obtenir B'. Avant cela, on doit adapter cette opération pour qu'elle puisse s'appliquer sur l'univers des Y. On a alors : $T^1_{sh_Y} = R_{[0,1]Y}(T^1_{sh_X})$ et $B' = T^1_{sh_Y}(B_1)$.

De la même façon, on construit B'' qui est similaire à B_2 comme A est similaire à A_2. On a : $T^2_{sh_Y} = R_{[0,1]Y}(T^2_{sh_X})$ et $B'' = T^2_{sh_Y}(B_2)$.

Construction de B

À partir des conclusions partielles B' et B'', la conclusion B pour A obtenue selon les deux règles étudiées, s'obtient en agrégeant les deux solutions partielles B' et B'' : $B = B' \oplus B''$ avec $\oplus$ un opérateur d'agrégation choisi. De plus, comme on considère que l'importance de chaque solution partielle dans la construction de B doit être dépendante de la distance qui sépare A des prémisses correspondantes, l'importance de B' (resp. B'') pour construire B dépend de L_1L_A (resp. L_2L_A), la distance qui sépare A et A_1 (resp. A_2). Dans l'agrégation à réaliser, il est donc naturel de pondérer le poids de chaque solution partielle par la distance qui lui correspond (plus c'est loin, moins le poids de la transformation doit être important). On a alors :

$$B = \frac{R_{XY}(L_1L_A)}{R_{XY}(L_1L_A + L_2L_A)}B' + \frac{R_{XY}(L_2L_A)}{R_{XY}(L_1L_A + L_2L_A)}B''.$$

et, ainsi, plus simplement :

$$B = \frac{L_1L_A}{L_1L_A + L_2L_A}B' + \frac{L_2L_A}{L_1L_A + L_2L_A}B''.$$

Exemple d'application

Considérons une base de règles assez simple, composée par les deux règles suivantes :

Règle 1 si la demande est faible et que la production est basse alors le bénéfice sera bas.

Règle 2 si la demande est importante et que la production est grande alors le bénéfice sera haut

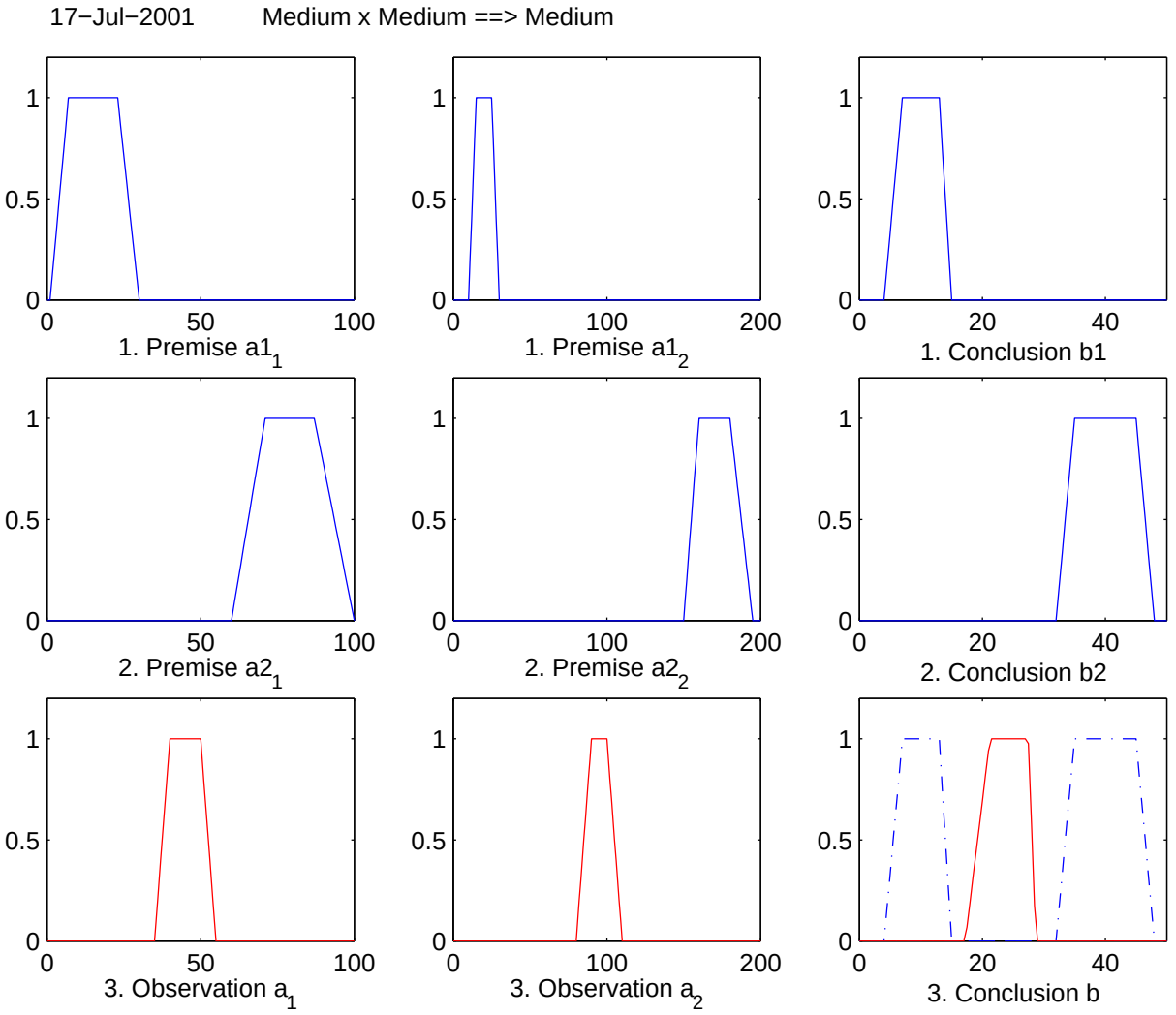

Figure 3.4 – Exemple d'application.

Observation La demande est normale et la production est moyenne.

Dans cet exemple simple, chaque règle permet de déterminer le bénéfice estimé pour une entreprise, en fonction de sa production et de la demande commerciale d'un produit. Dans cette application, aucune base complète ne peut être construite, et on ne peut donc posséder qu'un ensemble de règles décrivant les cas extrêmes.

Quand une observation est fournie ("la demande est normale et la production est moyenne" par exemple), le raisonnement par interpolation nous permet quand même d'en déduire une conclusion adéquate.

Par exemple, dans la Figure 3.4, on peut voir le résultat obtenu par notre approche de raisonnement interpolatif. Dans cette figure, la première ligne correspond à la première règle (on note $a1_1$ pour *faible*, $a1_2$ pour *basse*, et b_1 *bas*), la rangée correspond à la seconde règle (on note $a2_1$ pour *importante*, $a2_2$ pour *grande*, et b_1 pour *haute*). La troisième rangée de courbes représente l'observation (on note a_1 pour *normale*, et a_2 pour *moyenne*) et la conclusion b construite par notre approche. On peut noter que b peut ici s'interpréter comme une valeur

intermédiaire de bénéfice.

On peut aussi vérifier que dans le cas où les valeurs de l'observation sont incluses dans les valeurs des prémisses d'une des règles, la conclusion qui en est déduite est, elle aussi, incluse dans la conclusion de la règle correspondante (Figure 3.5).

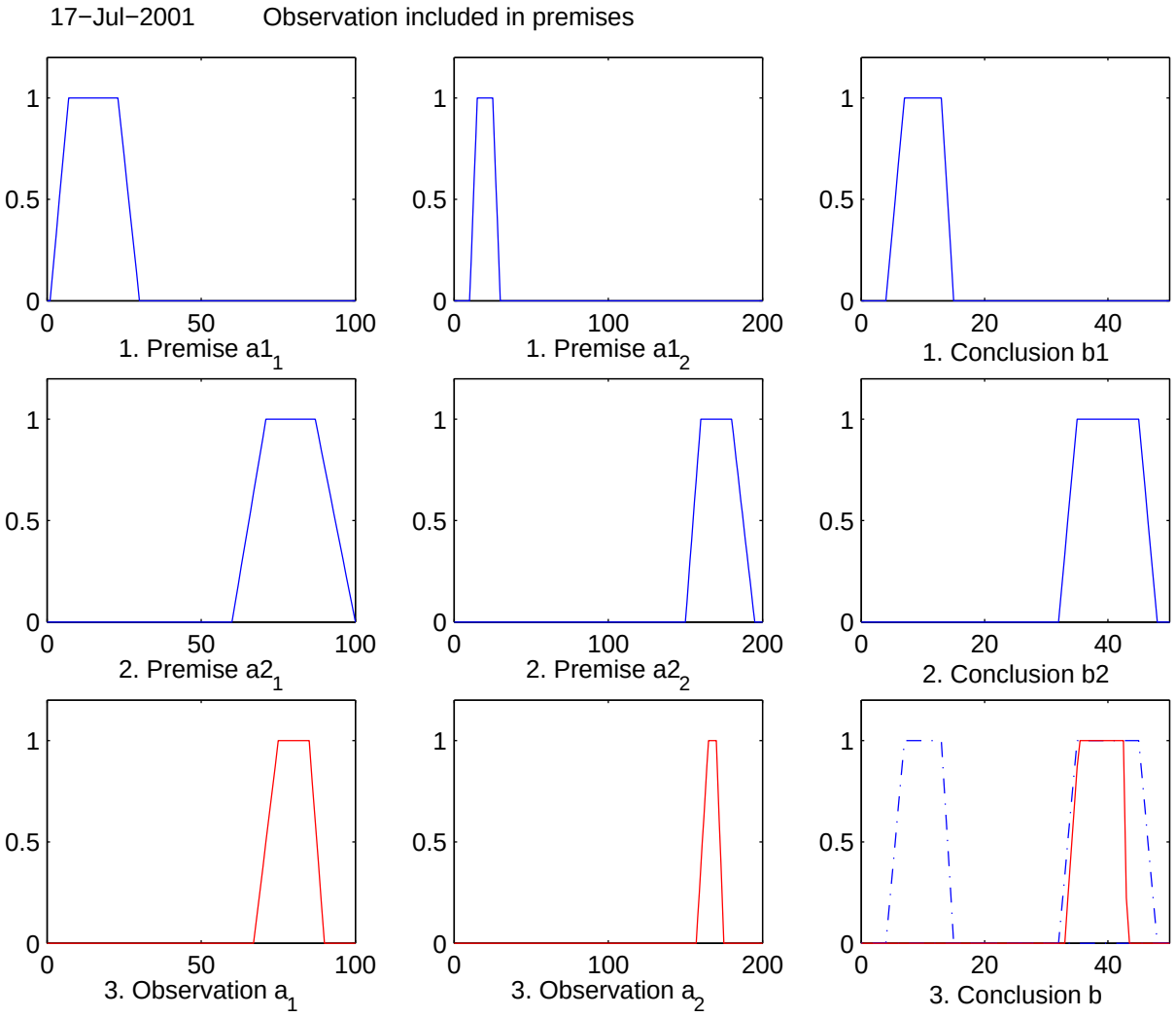

Figure 3.5 – Exemple d'application (2).

3.4 Apprentissage et interpolation

Il est possible de construire par apprentissage une base de règles comportant des règles incomplètes. Cela peut se produire dans le cadre d'applications où les données de la base d'apprentissage ne couvrent pas totalement l'espace des données possibles et pour lesquelles il ne serait alors pas judicieux de vouloir construire un modèle englobant tout l'espace à partir de ces données car cela n'aurait alors aucun sens réel.

Comme on vient de le voir dans ce chapitre, la possession finale d'une base incomplète n'est donc pas rédhibitoire si on peut mettre en œuvre un raisonnement interpolatif à partir de ses

règles.

Dans la première partie de ce chapitre, nous avons vu principalement deux méthodes de construction de bases de règles floues. La méthode de construction de résumés flous est une approche qui produit généralement un ensemble de règles incomplètes.

La méthode de construction d'arbres de décision flous, elle, par contre, n'est généralement pas conçue pour générer des règles couvrant incomplètement l'espace des données. En effet, son but est plutôt de construire un modèle recouvrant totalement l'espace des valeurs d'apprentissage. Ainsi, le recouvrement de cet espace sera même construit s'il n'est pas fourni intrinsèquement par l'ensemble des exemples de la base d'apprentissage. Cela se voit nettement par la mise en œuvre, en présence de données numériques, d'une discrétisation des univers de ces valeurs afin d'en déduire un recouvrement total.

Dans le cadre de la méthode de construction d'arbres de décision flous que nous avons vue dans la section précédente, les sous-ensembles flous construits le sont aussi dans le but d'obtenir un recouvrement maximum de l'univers des valeurs numériques. Pourtant, il est possible d'utiliser une méthode de construction de sous-ensembles flous qui pourrait générer une discrétisation floue incomplète de l'espace.

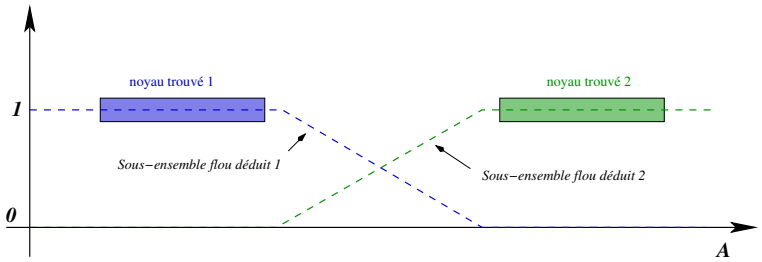

Figure 3.6 – Discrétisation continue de l'espace des valeurs réelles d'un attribut.

Par exemple, dans la méthode utilisée dans Salammbô pour déterminer des sous-ensembles flous sur l'ensemble des valeurs des attributs numériques [79], dans un premier temps, les noyaux possibles des sous-ensembles flous sont détectés dans l'ensemble des valeurs. Dans un second temps, les pentes de ces sous-ensembles flous sont déduites en rejoignant les bornes des noyaux trouvés, afin de fournir un degré d'appartenance à tout point de l'espace (voir Figure 3.6).

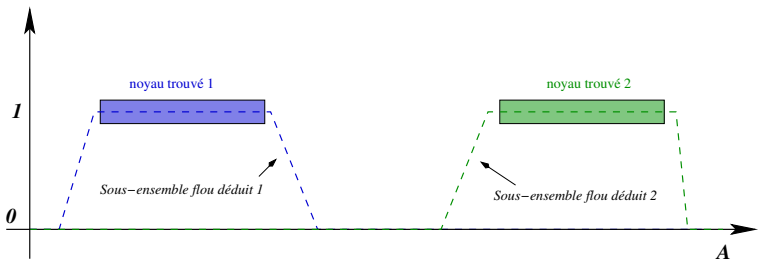

Figure 3.7 – Discrétisation incomplète de l'espace des valeurs réelles d'un attribut.

Pourtant, si les noyaux trouvés sont très éloignés, il est souvent utile de vouloir étendre leur support aussi largement dans l'espace des valeurs numériques. Il peut alors être préférable de restreindre leurs supports à une partie plus réduite, quitte à ne pas obtenir un recouvrement maximal de l'espace (voir Figure 3.7).

Ce type de modification n'engendre pas d'adaptation particulière de l'algorithme de construction d'arbres de décision flous pour en tenir compte. Par contre, à l'issue de la construction, on obtient un ensemble de règles qui peuvent donc couvrir incomplètement l'espace des données.

Ainsi, avec ces deux méthodes, par résumés flous ou arbres de décision flous, de construction de bases de règles incomplètes, il est alors possible d'obtenir un ensemble de règles par apprentissage. Ces règles peuvent alors être exploitées à l'aide de notre approche de raisonnement par interpolation.

Mais il existe aussi d'autres moyens pour modifier l'algorithme de construction d'arbres flous pour lui permettre de générer des bases de règles incomplètes. Cela mérite donc des recherches plus approfondies afin d'en déduire la façon de procéder la plus efficace.

3.5 Conclusion

Dans ce chapitre, j'ai présenté les travaux que j'ai menés conjointement avec Anne, Bernadette, Maria, et d'autres chercheurs de l'équipe LoFTI, sur les modèles d'apprentissage et de raisonnement flou.

Le but dans cet axe de mes recherches était double. D'une part, il était de mettre en parallèle le modèle de construction des arbres de décision flous avec un modèle différent de construction de règles floues par apprentissage, comme celui qui permet la génération des résumés flous. Cette comparaison m'a paru essentielle afin d'appréhender les différents paramètres de l'algorithme de construction d'arbres, et leur rôle, et de comprendre encore mieux leur impact sur l'arbre généré et sa relation avec l'espace des données. Cette mise en évidence des attributs communs à ces deux méthodes (résumés, arbres) et leurs correspondances ont permis de générer de nouvelles pistes d'amélioration pour chacun de ces algorithmes.

D'autre part, le but était de mieux comprendre les modes de raisonnement qu'il est possible de mettre en œuvre lorsque l'on possède une base de règles incomplètes. C'est ce qui a amené ainsi l'étude du raisonnement interpolatif et la proposition de nouveaux modes d'utilisation de ce type de règles. L'intérêt de telles connaissances incomplètes apparaît clairement quand on le relie à l'apprentissage inductif car souvent, il reste préférable de limiter la généralisation que l'on peut tirer d'une base d'apprentissage (qui, en fait, ne rend que très rarement une vue parfaitement complète de l'espace de recherche) et de se restreindre à la génération de règles qui ne couvrent que les données d'apprentissage et laissent inexpliquées les zones sur lesquelles aucune donnée n'est fournie.

Ces deux aspects sont donc fortement liés et j'en ai tiré quelques pistes de nouvelles recherches, comme je l'ai décrit dans la section sur l'apprentissage et l'interpolation.

Mais les travaux ici ne sont que les prémisses de travaux qu'il reste à mener pour mieux perfectionner la prise en compte de ce type de connaissances incomplètes et de leur apprentissage.

Des travaux doivent encore être poursuivis pour l'adaptation de l'algorithme de construction d'arbres de décision flous, ainsi que celui de génération de résumés flous (ce qu'a réalisé Anne Laurent dans ses propres travaux). De même, les mécanismes du raisonnement interpolatif demandent encore à être étudiés et approfondis afin d'en offrir un qui soit le plus proche du modèle humain d'interpolation.

3.6 Références

Ces travaux de ce chapitre sont issus de recherches conjointes que j'ai menées avec d'autres chercheurs. Ils ont donné lieu à des publications :

- travaux sur l'apprentissage de règles floues avec Anne Laurent et Bernadette Bouchon-Meunier : [12] ;
- travaux sur le raisonnement interpolatif avec Bernadette Bouchon-Meunier et Maria Rifqi principalement, mais aussi d'autres chercheurs : [20, 48, 49, 52, 59, 77].

Chapitre 4

Méthodes floues pour le Data Mining

4.1 Introduction

Dans ce chapitre, je présente les travaux communs que j'ai réalisés avec différents collègues chercheurs, sur quelques applications axées sur l'extraction de connaissances dans le but de construire des classifieurs et de mettre en œuvre une chaîne complète de fouille de données.

En extraction de connaissances à partir de données (KDD), le but principal est d'extraire des connaissances explicites à partir d'un grand ensemble de données, généralement stockées dans une base de données ou un entrepôt de données. Le KDD a ainsi pu s'appliquer à différents domaines dans lesquels une telle recherche de connaissances, qui représente alors une synthèse des données présentes dans la base, permet de mieux comprendre les données recueillies.

J'ai décidé de décomposer ces applications en deux types :

- les applications de data mining typiques, dans lesquelles on cherche à construire un classifieur qui sera utilisé par la suite pour classer des exemples. Cette construction se fait en travaillant sur un corpus comportant un grand nombre d'exemples d'apprentissage et pour lequel il est obligatoire de mettre en œuvre des techniques typiques d'extraction de connaissances afin de pouvoir obtenir un résultat dans un temps raisonnable.
- les applications de caractérisation, dans lesquelles on recherche surtout à obtenir un ensemble des caractérisations liant les exemples apprentissage. Dans ce type d'applications, ce qui intéresse surtout, c'est de mieux comprendre les relations existantes entre les valeurs des attributs qui décrivent un exemple, et la valeur de la classe qui lui correspond.

Comme applications de data mining, je vais présenter plusieurs applications sur lesquelles j'ai travaillé après ma thèse et jusqu'à ce jour. Tout d'abord, je décris l'application d'extraction de connaissances à partir d'une base de données spatiales orientée-objet géographique que nous avons réalisée conjointement avec Nara Martini Bigolin durant la fin de sa thèse. L'idée directrice dans ces travaux était d'introduire l'algorithme d'apprentissage au niveau du système de gestion de la base de données (SGBD) et de proposer un langage de requête étendu qui autorisait, dans le formalisme de ses clauses, à la fois un traitement des données par la théorie des sous-ensembles flous, et aussi la construction d'un modèle de connaissances par arbres de décision flous.

Ensuite, une présentation de travaux que j'ai menés avec Anne Laurent durant sa thèse pour la mise en œuvre de la construction de l'algorithme de construction d'arbres de décision flous à partir de données stockées dans une base de données multidimensionnelles. Ces travaux-ci sont le pendant des travaux précédents dans le sens où l'idée directrice qui les guide est de séparer l'algorithme d'apprentissage du SGBD et d'autoriser une communication entre lui et un SGBD évolué (multidimensionnel).

Comme applications plus particulièrement axées sur la caractérisation d'exemples, je présente deux applications dans le domaine médical. Une application sur la caractérisation de l'observance, menée en collaboration avec le Dr. Alain Lurie, dans laquelle les caractéristiques du suivi de leur traitement par des patients asthmatiques sont mises en évidence. Une application sur la caractérisation de maladies cardio-vasculaires, menée dans le cadre du projet national INDANA en collaboration avec Florence d'Alché-Buc pour le LIP6, et d'autres chercheurs d'équipes extérieures.

Finalement, pour clore ce chapitre, je présente une application de video mining pour l'extraction de connaissances de haut niveau à partir de séquences vidéos sur laquelle j'ai travaillé avec Marcin Detyniecki, et qui nous a amené à utiliser les forêts d'arbres de décision flous.

La description de ces travaux sera faite ici, dans le contexte de l'époque où ils furent effectués. Cela veut donc dire que je ne me suis pas attaché à mettre à jour les références pour les actualiser et que je vais présenter leur état de l'art au moment où ils furent menés (il y a 10 ans par exemple, pour les travaux sur le data mining sur les données géographiques). D'autre part, je ne présenterai ici qu'une synthèse rapide de chacune de ces recherches, plus de détails pourront être trouvés dans les publications que nous avons faites à l'époque et dont je récapitulerai les références à la fin de cette partie.

4.2 Lier un algorithme d'apprentissage et un SGBD

En extraction de connaissances à partir de données, les approches classiques ont souvent des difficultés à travailler sur des données structurées et manipulées par de véritables systèmes de gestion de bases de données (SGBD relationnel, objet, spatial). De plus, si les données sont massives et proviennent de sources multiples, comme c'est le cas pour les entrepôts de données (*data warehouses*), le problème de l'adaptation des algorithmes d'apprentissage est un point crucial dans la conception de logiciels de data mining efficaces.

Une des façons de faire est d'intégrer l'algorithme d'apprentissage directement dans le système de gestion de la base de données et d'y ajouter également tout ce qui concerne le prétraitement des données de la base à partir desquelles les connaissances doivent être extraites. C'est la première approche sur laquelle j'ai travaillé pour adapter les algorithmes de data mining et de les rendre aptes à explorer des bases de données existantes. Pour cela, l'idée est d'étendre un langage de requêtes pour extraire, dans un premier temps, des données pertinentes et les représenter alors sous la forme d'une simple base d'apprentissage. Cette base est alors prise en compte par un algorithme d'apprentissage, intégré dans le SGBD, qui l'utilise pour en induire des connaissances.

Une autre façon de faire est de laisser l'algorithme d'apprentissage complètement indépendant mais de déléguer également au SGBD tout ce qui concerne l'accès aux données. Ainsi, on peut utiliser les spécificités et les avantages d'un SGBD pour aider l'algorithme d'apprentissage pendant la construction de son modèle de connaissances. De cette façon, il est donc possible de conserver les spécificités et points forts de chaque système (SGBD et logiciel d'apprentissage) et de les amener à collaborer dans le processus même du data mining. Pour cela, il est nécessaire de choisir un SGBD qui soit adapté à la collaboration avec un logiciel d'apprentissage. C'est ainsi le cas avec SGBD pour les bases de données multidimensionnelles [32] et la technologie OLAP (*On Line Analytical Processing*) qui fournissent des solutions efficaces de manipulation et de synthèse des données d'une base. Ce modèle autorise l'exécution de requêtes complexes, travaillant sur un important volume de données, les données étant alors traitées par groupe et non à un niveau

individuel. Cette capacité des bases de données multidimensionnelles à calculer efficacement des agrégats complexes les rend donc très intéressantes dans le cadre de l'apprentissage. Cet intérêt de coupler les technologies de l'OLAP et du data mining a donné naissance à l'*OLAP Mining* ([59]), un processus de data mining intégrant une composante OLAP.

Dans ce qui suit, je présente des applications de ces deux types d'approches, sur lesquelles j'ai pu travailler.

4.2.1 Intégrer l'algorithme dans le SGBD

Les travaux de cette partie ont été réalisés conjointement avec Nara Martini Bigolin. Ils portent sur l'intégration d'un algorithme d'apprentissage flou dans le processus d'extraction de connaissances [50]. L'idée sous-jacente est d'insérer l'algorithme de data mining dans le langage de requêtes du système de gestion de la base de données. Ainsi, le but est donc d'enrichir ce langage d'une nouvelle commande de requête pour réaliser de la fouille de données et d'extraire une connaissance sur des données de la base plutôt qu'un ensemble de valeurs comme ceux fournis par les requêtes classiques.

Contexte

Avec Nara Martini Bigolin, nous avons travaillé sur l'étude de l'extraction de connaissances à partir d'une base de données orientée-objet géographique fournie par l'IGN pour sa thèse [94]. Nos travaux ont donné lieu à la proposition d'un langage de requête, FuSOQL (pour Fuzzy Spatial Object-oriented Query Language) que nous avons mis en œuvre pour l'extraction de connaissances à partir d'une base de données spatiales orientée-objet (BDSOO).

Dans ce langage, nous avons introduit des éléments de la théorie des sous-ensemble flous afin d'obtenir une meilleure gestion des données spatiales. Puis, nous avons appliqué un algorithme de construction d'arbres de décision flous sur les données sélectionnées, afin d'en obtenir des connaissances plus synthétiques. Nous avons pu expérimenter cette approche sur des données géographiques décrivant une région française et pour laquelle un ensemble de règles de classification a été découvert afin de discriminer les maisons urbaines des maisons rurales.

Dans le domaine des systèmes d'information géographiques (GIS), des recherches en extraction de connaissances ont été menées sur des bases de données spatiales [51]. Dans ce cadre particulier, l'idée principale est d'extraire des motifs spatiaux intéressants, des caractéristiques, et des relations générales entre les données spatiales et non-spatiales, ainsi que des caractéristiques générales des données qui ne sont pas explicitement stockées dans la base de données spatiales [77]. La proposition de méthodes pour réaliser cela est devenue de plus en plus cruciale avec le développement important de ce type de données ainsi que de l'énorme quantité de données spatiales et non-spatiales qui sont actuellement collectées et stockées.

A l'époque de nos travaux, il existait déjà plusieurs langages de requêtes pour extraire des connaissances à partir de données spatiales, mais les plus significatifs avaient été développés pour une base de données relationnelle [49, 59, 70].

De notre côté, nous avons donc proposé une extension du langage de requêtes OQL (Object Query Language) permettant de sélectionner et de traiter des données contenues dans une BDSOO. De plus, nous y avons introduit l'utilisation de la théorie des sous-ensembles flous, afin de traiter les données numériques et imprécises, et nous y avons incorporé un algorithme d'apprentissage pour générer les connaissances. À titre d'expérimentation, nous avons choisi l'algorithme de construction d'arbres de décision flous pour la génération des connaissances.

Pour découvrir des connaissances dans une base de données orienté-objet spatiale (BDOOS), il est nécessaire de prendre en compte simultanément à la fois la nature orientée-objet des données, et aussi leur spécificité spatiale.

Dans nos travaux avec Nara Martini Bigolin, nous avons donc proposé une extension de SQL capable de réaliser une telle prise en compte. Nous avons ainsi défini FuSOQL qui intègre, en plus de la prise en compte des données spatiales et orientées-objet, la possibilité d'utiliser la théorie des sous-ensembles flous pour prendre en compte les données spatiales numériques. Il intègre aussi le choix d'un algorithme de data mining à appliquer sur les données sélectionnées afin de découvrir de nouvelles connaissances.

Dans un premier temps, la sélection des données est faite à l'aide d'une requête spatiale qui est lancée sur la BDOOS afin d'extraire un sous-ensemble d'objets spatiaux susceptibles d'être pertinents pour l'étape de data mining. Un traitement appliqué sur chaque objet spatial permet ensuite de transformer ces objets en une base d'apprentissage classique. Le traitement peut être soit mathématique (calcul de mesures comme la distance Euclidienne, la longueur,...), ou bien à base de théorie des sous-ensembles flous (calcul de fonctions d'appartenance pour obtenir des valeurs floues)

Une fois préparé, le fichier d'apprentissage est utilisé par un algorithme d'apprentissage (dans notre application, nous avons choisi la construction d'arbres de décision flous) afin d'obtenir des connaissances interprétables et facilement exploitables par la suite. Ce sont ces connaissances qui sont fournies en réponse à la requête faite.

Application

Le modèle précédent a été appliqué sur des données fournies par l'Institut Géographique National (IGN) dans le cadre de la thèse de Nara Martini Bigolin. Le but de cette application était de trouver un ensemble de règles de classification afin de pouvoir déterminer, en fonction de sa description et de son entourage, si une maison est localisée en milieu rural ou à l'intérieur d'une ville.

Dans l'architecture globale du système utilisé, FuSOQL est une couche au dessus du système de gestion de base de données O_2 [5] qui utilise un composant géographique et spatial GIS GeO_2 [101]. Les requêtes sont réalisées en utilisant des connaissances sur le domaine, fournies sous la forme de sous-ensembles flous. Les arbres de décision flous fournis en réponse à une requête, sont construits grâce au programme Salammbô [79].

Conclusion

Au final, dans ces travaux, nous avons mis en œuvre un système pour extraire des connaissances qui intègre tout le processus d'extraction dans le SGBD de la base, offrant ainsi à l'utilisateur un moyen simple de réaliser de la fouille de données sur une base.

Si une telle approche est apparue intéressante pour réduire la mise en œuvre de l'extraction des connaissances du point de vue de l'utilisateur, elle laisse entrevoir quelques limitations dans l'implémentation qu'elle requiert. En effet, une telle méthode demande l'introduction, au niveau du SGBD, des nouveaux algorithmes que l'on souhaite utiliser. Cela entraîne donc une certaine lourdeur car on devient donc complètement dépendant du SGBD sur lequel on se place (schéma compris). Ce type de solution apparaît donc très adéquate pour des applications bien ciblées (comme celle sur les données géographiques présentée ici), mais peut être un peu moins efficace si l'on souhaite avoir une solution plus généraliste, fonctionnant quelle que soit la base de données à traiter.

4.2.2 Faire coopérer l'algorithme avec le SGBD

Je présente ici des travaux, menés en collaboration avec Anne Laurent et Stéphane Gançarski, qui nous ont permis de mettre en œuvre une coopération entre un système d'apprentissage flou et un système de gestion de bases de données multidimensionnelles.

L'idée maîtresse de ces travaux est de construire une architecture générique pour réaliser de l'extraction de connaissances à partir de grandes bases de données, éventuellement hétérogènes. À la différence de l'étude précédente, l'extension du langage de requêtes du SGBD n'est plus une solution vraiment intéressante dans ce cadre.

C'est pourquoi, dans la nouvelle approche que nous avons proposée, et qui est présentée ici, nous avons opté pour l'utilisation d'un moteur OLAP (*On Line Analytical Processing*) pour réaliser l'interface pour des requêtes sur tout type de bases de données, et obtenir ainsi très aisément un ensemble de statistiques sur les données d'une base.

Contexte

Le but de notre étude a été de proposer un système général, ouvert pour l'intégration de la technologie OLAP avec diverses méthodes d'apprentissage, y compris des méthodes d'apprentissage flou. Ce système général est construit sur une architecture de type client-serveur, considérant la méthode d'apprentissage comme un client demandant des services à l'application interface le reliant à la base de données. L'intérêt d'une telle architecture est de tirer avantage des deux composantes et de limiter leurs inconvénients. Par exemple, un algorithme d'apprentissage par construction d'arbres de décision a besoin d'informations relatives aux attributs afin de minimiser une mesure d'entropie ; en revanche, un algorithme d'apprentissage par construction de prototypes nécessite des informations relatives aux valeurs d'attributs pour maximiser une mesure de similarité [102].

Dans l'approche que nous avons proposée, nous faisons coopérer deux composantes spécifiques : un SGBD multidimensionnel (SGBDM), adapté à la manipulation des données, et une méthode de data mining, adaptée à l'extraction de connaissances à partir de données. Ainsi, la gestion de la base d'apprentissage ne se fait plus au niveau de la méthode d'apprentissage qui garde sa vocation première mais qui n'a plus à prendre en charge les contraintes de stockage et de manipulation des données, celles-ci étant prises en charge par le SGBDM. Pour cela, d'une part, l'algorithme d'apprentissage doit remplacer sa gestion interne de la base d'apprentissage (chargement, calcul de statistiques, ...), par un mécanisme de communication avec le SGBDM, et, d'autre part, le SGBDM doit être doté de capacités de communication pour lui permettre de recevoir des requêtes et de donner des réponses de façon automatique. Pour illustrer et étudier une telle approche, nous avons utilisé deux systèmes existants : le SGBDM *Oracle Express* (OE), de la société Oracle, et le logiciel d'apprentissage *Salammbô* [79] qui construit des arbres de décision flous.

Dans les entrepôts de données, le processus d'extraction de connaissances doit utiliser des données volumineuses et hétérogènes, ce qui rend le calcul des agrégats directement à partir des sources de l'entrepôt trop complexe pour être applicable. Cette nécessité de calculer et de maintenir des données agrégées selon plusieurs dimensions est commune à un grand nombre d'applications d'analyse en ligne (OLAP [32]). Elle a donné lieu à l'émergence de nouvelles technologies qui fournissent des moyens de collecter, stocker et traiter des données multidimensionnelles à des fins d'analyse. Ces données sont stockées dans des *hypercubes* (ou plus simplement cubes) [4]. Un cube est un ensemble de données organisées comme un tableau multidimensionnel. La *mesure* représente la valeur contenue dans les cellules du cube, chaque cellule étant définie par

une valeur (ou modalité) pour chaque dimension. De plus, des hiérarchies peuvent être définies sur ces dimensions de manière à agréger les données selon différents niveaux.

L'intérêt d'utiliser les SGBDMs comme interface pour permettre aux algorithmes d'apprentissage d'extraire des connaissances à partir des masses volumineuses de données des entrepôts apparaît donc naturel.

Pour la construction d'un arbre de décision flou, Salammbô, le logiciel d'apprentissage que nous avons considéré, calcule des agrégats et des probabilités floues afin de mettre en valeur le meilleur attribut pour partitionner la base d'apprentissage, et pour évaluer le critère d'arrêt pour chaque nœud de l'arbre [79, 99]. Le choix du meilleur attribut dépend de l'information liée à la classe apportée par cet attribut. Cette information est mesurée par l'entropie d'événements flous.

L'interfaçage de Salammbô avec un SGBDM pour construire les arbres de décision flous offre donc la possibilité de pouvoir s'affranchir de la représentation et de la taille des données, et de permettre ainsi de traiter un grand nombre de données provenant éventuellement de sources diverses comme c'est le cas dans les entrepôts de données.

Dans notre approche, nous avons mis en évidence qu'un des principaux problèmes pour la mise en œuvre d'une collaboration entre un SGBDM et un système d'apprentissage réside dans l'équilibrage du rôle de chaque composant dans le processus d'extraction. Ainsi, par exemple, les outils d'apprentissage flou effectuent eux-mêmes des agrégats et des calculs flous. Or un système multidimensionnel peut aussi être très efficace pour exécuter des calculs complexes sur les données.

Un SGBDM peut donc soit envoyer des agrégats simples sur les données (par exemple le nombre d'exemples correspondant à chaque classe), soit calculer des agrégats complexes incluant des opérations floues et des calculs d'agrégats et d'en transmettre le résultat. Si la première approche a l'avantage de faciliter l'intégration d'autres algorithmes d'apprentissage pour ce type de coopération, la seconde approche, elle, tire mieux parti de la puissance du SGBDM pour gérer les calculs d'agrégats.

Dans les deux cas, les données transmises par le système d'apprentissage au SGBDM sont les mêmes : le système d'apprentissage questionne le SGBDM en lui transmettant un ensemble de données décrivant son état courant de la phase de construction. Par exemple, *Salammbô* interroge le SGBDM pour avoir des informations sur la branche courante de l'arbre de décision qu'elle est en train de construire. Cette interrogation est composée par des couples [attribut, valeur] qui représentent le chemin actuellement en cours de développement (pour plus de détails sur la construction d'un arbre de décision flou voir [79]).

Application

L'architecture proposée a fait l'objet de tests sur deux bases de données. La première application a porté sur une base multidimensionnelle fournie avec le serveur Oracle Express. Puis une autre application a été réalisée avec un volume de données plus conséquent issue d'une base de données relationnelle fournie par le MENRT, contenant les résultats au baccalauréat sur deux années consécutives, soit environ un million d'enregistrements. A partir de cette base, un ensemble de règles (issues des arbres de décision flous que nous avons pu construire) a été obtenu. Deux exemples de règles extraites sont présentés ci-dessous :

R1 : Si l'établissement est public, et que la série du bac est ES, alors la proportion de candidats reçus avec mention (AB, B ou TB) est faible.

R2 : Si l'établissement est privé sous contrat, que l'année de rentrée est 1996, que la spécialité est LV2, et que la série du bac est STI, alors la proportion de candidats reçus avec mention (AB, B ou TB) est élevée.

Conclusion

Dans ces travaux, nous avons proposé une nouvelle approche d'OLAP Mining pour extraire des connaissances à partir de grandes bases de données. Cette approche intègre le couplage de deux composantes spécifiques : un SGBD multidimensionnel, adapté à la manipulation des données agrégées, et une méthode d'apprentissage, adaptée à la construction de connaissances à partir de données.

Les tests ont montré la viabilité d'une telle architecture. Elle constitue un excellent moyen de prendre en compte non seulement des volumes de données importants mais aussi des données numériques, qui peuvent être imprécises et/ou incertaines, ce que les systèmes d'OLAP Mining existants ne permettent pas. De plus, l'intérêt (scientifique et économique) porté aux entrepôts de données et aux technologies OLAP laissent envisager un accroissement rapide de l'efficacité de ces systèmes.

4.2.3 Bilan

Au final, que peut-on donc déduire de ces deux types d'approches ?

Dans les travaux menés avec Nara Martini Bigolin, nous avons pu mettre en évidence qu'avec ce type d'approches, on tend à résoudre l'adaptation de l'algorithme d'apprentissage pour la prise en compte de données gérées par un SGBD. Si l'application que nous avons réalisée a pu montrer l'intérêt de ce nouveau langage, la généricité de ce type de solutions semble plutôt limitée car ce nouveau langage de requêtes doit être réalisé sur un SGBD spécifique, ce qui rend donc la prise en compte d'une base dépendante de ce SGBD.

Dans les travaux menés avec Anne Laurent et Stéphane Gançarski, nous avons mis en œuvre une architecture distribuée permettant de faire collaborer un algorithme d'apprentissage avec un SGBD à travers un moteur OLAP. Ce type de solutions apparaît alors plus pérenne et générique car toute base de données, quel que soit son SGBD (sous réserve qu'il soit capable de s'interfacer avec un moteur OLAP), peut alors être prise en compte.

4.3 Caractérisation de patients dans le domaine médical

En apprentissage inductif, la caractérisation permet d'étudier les relations existantes entre un ensemble de caractéristiques descriptives (la description) et la classe. L'idée est d'en dégager des caractérisations de la classe à l'aide d'un ensemble de valeurs observables. Ainsi, on peut alors construire un modèle qui, de plus, peut être interprétable selon l'algorithme d'apprentissage choisi (par exemple, les arbres de décision flous).

Ce genre d'application des arbres de décision est très intéressant dans le domaine médical car on peut ainsi construire des caractérisations des patients selon une maladie particulière, un mauvais fonctionnement, l'observance d'une médication, ou tout autre fait à comprendre. Comme, de plus, de nombreuses variables qui permettent de décrire les patients sont continues, ou imprécises, l'utilisation d'un modèle traitant les valeurs floues est hautement recommandé.

Il existe de nombreux travaux réalisés dans le domaine médical pour la prise en compte des données à l'aide de la théorie des sous-ensembles flous et les outils de data mining flou. Par

exemple, [96] utilise des bases de règles floues comme classifieurs pour prendre en compte des données médicales. Dans cette application, une approche neuro-floue est utilisée et les résultats obtenus démontrent que des classifieurs flous de ce type sont très intéressants pour de l'aide au diagnostic. Dans [56], une approche floue est utilisée pour faire de la fouille de données médicales. Des algorithmes de clustering flou sont utilisés pour sélectionner des caractéristiques et construire un ensemble de règles de décision floues.

Dans cette partie, je présente deux applications dans le domaine médical dans lesquelles j'ai utilisé les arbres de décision flous pour faire de la caractérisation de patients. Ces deux applications ont été réalisées avec le concours de médecins spécialisés.

4.3.1 Caractérisation de risques cardio-vasculaires

L'objectif principal de cette application était de trouver des caractéristiques discriminantes pour pouvoir prévenir les maladies cardio-vasculaires. Le but était donc de construire un prédicteur qui puisse aider les médecins pour détecter ces risques chez les patients hypertendus. Cette application s'est déroulée dans le cadre d'un projet de recherche commun à plusieurs équipes.

Les données utilisées étaient issues de la base INDANA (INdividual Data ANalysis of Antihypertensive intervention) [57] à partir de laquelle a été construite une base d'apprentissage composée de 10 échantillons thérapeutiques de patients ayant un risque cardio-vasculaire. Chaque patient est décrit par un ensemble de caractéristiques classiques, qui sont combinées avec un ensemble de mesures médicales prises sur plusieurs années. La classe associée au patient est relative à sa mortalité à l'issue de la phase de mesure. Dans la base que nous avons utilisée pour ces expérimentations, 2230 patients étaient décrits par une vingtaine de caractéristiques (l'identifiant, le genre, l'âge, la taille, le poids, les mesures médicales, et la classe (décédé ou non)). Ces caractéristiques ont été mesurées sur une grande période temporelle, et la classe a été affectée à la fin de la période, selon que le patient en question était décédé ou non. Dans cette base, 107 patients sont décédés d'une maladie cardio-vasculaire, et 2123 étaient vivants à l'issue de la période de suivi.

Un des problèmes principaux à résoudre dans cette application vient du fait que la proportion des classes est très fortement déséquilibrée, ce qui rend impossible l'utilisation de l'algorithme de construction d'arbres de décision classique. En effet, dans ce cas-là, la classe minoritaire est complètement masquée par la classe majoritaire, ce qui peut fausser l'évaluation de l'information apportée par les attributs et provoquer un arrêt précoce du processus de construction de l'arbre. C'est pourquoi la solution que j'ai adoptée utilise une forêt d'arbres de décision flous. L'idée était d'extraire des échantillons de la base comportant un nombre égal de patients de chaque classe, et d'en construire un arbre de décision flou. Une base d'apprentissage est donc constituée de patients, choisis aléatoirement parmi les patients de la classe minoritaire (par exemple, 75% des patients de cette classe) et par un nombre équivalent de patients de la classe majoritaire. Cet échantillonnage est effectué 100 fois afin de couvrir au maximum, et un maximum de fois, tous les patients de la base originale. A partir de chacune de ces bases d'apprentissage, est construit un arbre de décision flou grâce au logiciel *Salammbô* [79]. L'ensemble des arbres construits avec toutes ces bases d'apprentissage constitue donc la forêt d'arbres de décision flous.

Afin d'étudier les arbres obtenus, chaque patient non utilisé dans la construction d'un arbre a été classé avec cet arbre afin de prédire sa classe (*décédé* ou non). L'ensemble des classifications obtenues par tous les arbres de la forêt a été ensuite agrégé par un vote majoritaire, afin de déterminer la classe du patient prédite par la forêt complète. Pour avoir une base de comparaison, j'ai testé de deux manières les forêts ainsi obtenues. D'une part, en considérant les décisions

prises lors de la classification comme floues (les arbres construits sont donc considérés comme des arbres de décision flous à part entière), et, d'autre part, en considérant les décisions prises dans l'arbre comme complètement précises (les arbres sont utilisés comme des arbres de décision classiques). Dans le premier cas, lors de sa classification, chaque patient est donc associé aux classes avec un degré d'appartenance, alors que dans le second cas, il est associé à une classe unique.

	Arbres classiques	Arbres flous
Sensibilité	59.8%	70.0%
Spécificité	59.4%	54.3%

Table 4.1 – Prédiction du risque cardio-vasculaire.

Les résultats obtenus, relatifs à la reconnaissance de la classe "*décédé*", sont présentés dans la Table 4.1. Comme je ne m'intéresse ici qu'aux apports des arbres flous par rapport aux arbres classiques, je ne détaillerais pas plus les résultats obtenus sur l'ensemble du projet INDANA. On peut observer que les forêts d'arbres flous sont meilleures que les forêts d'arbres classiques pour prédire correctement le risque de décès (haute sensibilité) mais que cela entraîne une augmentation du nombre de mauvaises détections du risque (la spécificité baisse).

Plus intéressante est l'analyse des détails de la classification. Elle laisse apparaître l'existence de disparités entre les patients. Il existe des patients qui sont très difficiles à classer, alors que, pour d'autres, le taux de bonnes classifications atteint les 100%. De plus, l'examen des attributs utilisés dans les arbres construits permet de comprendre les règles les liant à la prédiction de la classe.

4.3.2 Caractérisation de l'observance médicale

Cette expérimentation, qui a été menée conjointement avec le docteur Alain Lurie[1], avait pour objectif principal d'étudier la perception de la sévérité de leur maladie par des patients souffrant d'asthme, puis de comparer les variables identifiées comme étant déterminantes dans cette perception, avec les variables impliquées dans l'évaluation de la sévérité de l'asthme selon les recommandations du *National Asthma Education and Prevention Program (NAEPP).*

La base de données est composée d'un ensemble de 113 patients (62 femmes et 51 hommes), souffrant d'asthme à différents degrés (6.2% d'asthme léger, 15.9% d'asthme modéré, et 12.4% d'asthme sévère. Un questionnaire a été rempli par chaque patient, ce qui permet de leur associer un ensemble de caractéristiques (entre autres, caractéristiques socio-démographiques, et caractéristiques sur leur asthme). Les caractéristiques sur l'asthme se décomposent en deux parties : celles estimées par le patient lui-même, et celles évaluées par les médecins.

Les patients ont à estimer la perception qu'ils ont de la sévérité de leur maladie (évaluée comme "mild intermittent", "mild persistent", "moderate", ou "severe"), la perception qu'ils ont de l'efficacité de leur traitement, et ils remplissent des questionnaires permettant d'évaluer leur qualité de vie (questionnaires "Asthma Quality of Life Questionnaire" (AQLQ)). On leur associe alors une estimation de l'observance de leur médication.

De leur côté, les médecins fournissent des informations sur la sévérité objective de l'asthme de leur patient (aussi évaluée comme "mild intermittent", "mild persistent", "moderate", ou "severe") qui est estimée selon des critères médicaux et des mesures de fonctions respiratoires.

[1] Service de Pneumologie, Hôpital Cochin, Assistance Publique-Hôpitaux de Paris.

Toutes ces variables sont réunies pour décrire un patient, et le postulat de l'étude est qu'elles devraient entrer en compte dans la détermination de la sévérité perçue de son asthme par le patient (sévérité perçue qui servira donc de classe dans notre étude).

Afin d'analyser les relations entre ces caractéristiques et la classe choisie, nous avons utilisé l'algorithme de construction d'arbres de décision flous avec le logiciel *Salammbô*. Ainsi, un ensemble de variables et de valeurs caractéristiques a pu être mis en évidence pour la détermination de cette classe par le patient.

Rang	Attribut
#1	AQLQ - questions sur la santé
#2	Ancienneté de l'asthme
	Dyspnée le jour de l'étude
#3	AQLQ - questions sur la santé
	Ancienneté de l'asthme
	Court traitement à base de corticostéroïds (les 12 derniers mois)
	Sévérité perçues durant les 6 dernières semaines
#4	Âge
	Nombre total d'inhalations prises par jour
	Nombre d'inhalations de corticostéroïds
	AQLQ - questions sur l'essouflement
	Dyspnée le jour de l'étude
#5	Traitement longue durée de théophylline

Table 4.2 – Attributs principaux apparaissant dans l'arbre de décision flou

L'arbre ainsi construit a été analysé par le médecin afin d'en étudier les relations entre les caractéristiques et la classe mise en évidence. Il apparaît alors que les attributs présents dans l'arbre final montrent une différence entre le point de vue du patient et celui du médecin pour l'estimation de la sévérité de l'asthme. Ainsi, les variables principalement sélectionnées dans l'arbre (voir Table 4.2, où "rang" est la position de la variable depuis la racine de l'arbre) sont très différentes de celles utilisées généralement par les médecins pour estimer la sévérité de l'asthme de leurs patients [5].

D'autre part, cette étude met en évidence l'importance de la mesure AQLQ pour l'estimation de la sévérité de leur asthme par les patients. Cela renforce l'opinion selon laquelle les patients ont tendance à sous-estimer la sévérité de leur asthme. Cette tendance créée un comportement très dangereux, car cela peut inciter le patient à stopper sa médication, au risque d'avoir des complications très sérieuses pour sa santé.

Pour renforcer ces résultats, nous avons mené une cross-validation afin d'estimer les taux de reconnaissance obtenus par ce type d'arbre. L'ensemble des patients a été décomposé en 4 sous-ensembles qui ont servi tour à tour de base d'apprentissage (regroupés par 3) et de base de test. Ainsi, le taux de bonne classification atteint $73\% \pm 7\%$ ce qui montre donc une certaine fiabilité des arbres ainsi construits.

4.3.3 Bilan

Ces deux applications des arbres de décision flous dans le domaine médical, ont permis de mettre en évidence l'intérêt de ce type de modèle à base de règles floues dans ce domaine. De

plus, ces travaux ont été menés avec des experts médicaux qui ont pu ainsi valider les arbres obtenus sans avoir aucune connaissance préalable sur le modèle.

4.4 Video mining

Le traitement d'un grand volume de données est le challenge prépondérant de l'extraction de connaissances à partir de données. Dans le domaine de l'indexation vidéo, la taille des corpus disponibles est très importante et se prête donc bien à des études sur l'applicabilité des algorithmes de data mining.

Avec Marcin Detyniecki, nous avons donc mené un ensemble de travaux afin de mettre en œuvre un algorithme automatique pour extraire des caractéristiques de haut niveau dans des vidéos. Pour cela, nous nous sommes confrontés à un problème concret, dans le cadre de la compétition annuelle internationale TRECVid [104, 105]. De plus, nous nous sommes attachés à garder un modèle interprétable, à base d'arbres de décision flous, afin de pouvoir conserver une explicabilité des décisions de classification prises et de pouvoir aussi étudier l'influence des descripteurs et leur rôle pour la détermination de la classe.

La tâche du challenge TRECVid à laquelle nous avons participé (quatre années consécutives, de 2005 à 2008) avait pour but la reconnaissance de concepts de haut niveau dans des shots d'un corpus de vidéos hétérogènes, après une étape de prétraitement des vidéos afin d'obtenir une description de chaque shot utilisable par un algorithme d'apprentissage.

Je décris dans ce qui suit tout d'abord l'étape de prétraitement que nous avons élaborée avec Marcin Detyniecki, puis je donne le détail de la mise en œuvre de l'algorithme d'apprentissage que nous avons décidé d'utiliser pour cette approche. Pour illustrer, je me focaliserai sur la tâche d'extraction de caractéristiques de haut niveau du challenge TRECVid de l'année 2008. Ce que nous avons fait les années précédentes était plus ou moins similaire et la description de nos approches pourra être trouvée plus en détail dans les articles fournis en référence en fin de ce chapitre.

4.4.1 Extraction de caractéristiques de haut niveau

Le corpus sur lequel nous avons travaillé, dans le cadre du challenge TRECVid 2008, comporte près de 200 heures de vidéos, qui se décomposent en 100h à utiliser comme corpus de développement (base d'apprentissage), et 100h qui servent de benchmark (base de test). Cela représente 33726 shots de référence et 215 fichiers vidéo. Les informations sur les shots sont fournies dans le cadre du challenge et issues d'une décomposition automatique dans laquelle les frontières temporelles des shots sont déterminées en fonction de la similarité des images qui le composent. Un shot est aussi associé à un ensemble d'images caractéristiques (une *keyframe* et, éventuellement, d'autres *frames*) censées le représenter.

La tâche d'extraction de caractéristiques de haut niveau consiste à retrouver un certain nombre de caractéristiques présentes dans les shots. Ainsi, en 2008, il y avait 20 caractéristiques à reconnaître : *classroom, bridge, emergency vehicule, dog, kitchen, airplane flying, two people, bus, driver, cityscape, harbor, telephone, street, demonstration or protest, hand, mountain, nighttime, boat ship, flower, singing.* Chaque caractéristique est considérée comme soit absente, soit présente complètement dans un shot. Pour chaque caractéristique, le but de la tâche est de fournir un ordonnancement d'au plus 2000 shots du corpus de test qui sont susceptibles de la contenir.

Dans notre approche, nous utilisons les informations fournies sur les shots (durée, position, etc.) ainsi que les frames qui le représentent, pour constituer un ensemble de descripteurs qui

pourra être utilisé en apprentissage pour construire un classifieur, et en test pour déterminer la présence d'une caractéristique dans un shot.

Les descripteurs que nous avons proposés pour représenter un shot sont de deux ordres. Les *descripteurs visuels* sont construits à partir des frames présentes dans le shot. Chaque image est ainsi découpée en zones, qui peuvent se chevaucher, sur lesquelles un histogramme de couleurs, représentées dans l'espace HSV et selon différents niveaux de granularité selon la position de la zone dans l'image, est calculé. Les *descripteurs temporels* sont construits en fonction des informations temporelles associées au shot (sa position dans la vidéo, sa durée, la position temporelle de ces frames,...).

Pour chaque shot de la base de développement, il est possible d'associer les caractéristiques de haut niveau qui ont été détectées dans ce shot, à l'issue d'une indexation manuelle (réalisée conjointement par des équipes participant au challenge).

Durant nos différentes participations à ce challenge TRECVid, nous avons étudié différents types de descripteurs visuels et temporels. En particulier, nous avons essayé, au fil du temps, différentes façons de découper les images. Pour plus de détails, je renvoie à nos comptes-rendus sur chacune de nos participations à ce challenge [80, 79, 90, 92].

4.4.2 Forêts d'arbres de décision flous

L'utilisation d'un algorithme d'apprentissage pour induire un classifieur à partir de la base d'apprentissage construite à l'étape précédente n'est pas simple. Avec Marcin Detyniecki, nous nous sommes focalisés sur l'utilisation de l'algorithme de construction d'arbres de décision flous. Les autres équipes participantes à ce challenge possédant leurs propres méthodes, il était donc clair que nous pourrions ainsi effectuer une comparaison de ces approches à l'issue de la publication des résultats finaux.

L'avantage principal d'utiliser les arbres de décision flous réside dans leur interprétabilité qui permet, une fois construit, d'obtenir un ensemble d'informations claires sur les règles de décision qui les composent. D'autre part, c'est un algorithme suffisamment rapide à mettre en œuvre et à utiliser, même en présence d'une importante masse de données. Par contre, il doit être adapté pour pouvoir gérer des problèmes multi-classes ou des problèmes où les proportions des classes sont fortement déséquilibrées.

Dans notre approche, nous avons décidé d'utiliser des forêts d'arbres de décision flous, un peu sur le modèle de ce que j'avais réalisé dans l'application médicale de caractérisation de risque cardio-vasculaire qui a été décrite dans la section précédente. Pour chaque caractéristique, nous avons constitué une base d'apprentissage en libellant chaque shot par une classe binaire (caractéristique présente, ou non). Partant de là, une forêt d'arbres de décision flous est construite pour reconnaître l'occurrence d'une caractéristique dans un shot, selon la méthode décrite dans la Section 2.3.3. Ainsi, une forêt est construite indépendamment pour chacune des caractéristiques.

Pour déterminer si une caractéristique apparaît dans un shot, on classifie celui-ci à l'aide de la forêt construite pour cette caractéristique. La classification s'effectue selon la méthode décrite dans la Section 2.3.3 et permet d'obtenir un degré avec lequel la caractéristique est susceptible d'apparaître dans le shot. Ainsi, tous les shots du corpus de test peuvent être classés par la forêt afin d'obtenir un ordonnancement (sur les degrés d'appartenance des shots à la classe fournis par la forêt) de tous ces shots, et d'en faire ressortir les 2000 premiers pour lesquels le degré d'appartenance est le plus fort.

Les résultats que nous avons obtenus avec ce type d'approches sont très prometteurs et ils se situent un peu en dessous de la moyenne des méthodes participantes, ce qui est très intéressant

pour les descripteurs très élémentaires que nous utilisons (par rapport aux autres approches qui utilisent généralement des descripteurs plus complexes et plus coûteux à appliquer).

4.4.3 Bilan

Nos travaux avec Marcin Detyniecki dans le domaine du video mining lors des différents challenges TRECVid auxquels nous avons participé, nous ont permis de mettre en œuvre l'algorithme de construction d'arbres de décision flous et de proposer l'utilisation d'une nouvelle forme de forêts d'arbres flous pour ce type de problèmes. Les résultats que nous avons obtenus ont eu l'avantage de montrer que ce type de solutions, basée sur une méthode facilement interprétable et aisément mise en œuvre, était viable pour de telles applications.

4.5 Conclusion

Au cours des travaux décrits dans ce chapitre, j'ai donc eu l'opportunité, en collaborant avec d'autres chercheurs, de confronter l'algorithme des arbres de décision flous à des applications dans différents domaines.

Toutes ces applications m'ont permis de confronter mes recherches à des problèmes du monde réel concrets. J'ai ainsi eu l'occasion d'étudier le positionnement de l'algorithme d'apprentissage dans la chaîne de traitement complète mise en œuvre pour extraire de la connaissance à partir d'une base de données et de me confronter aux difficultés apparaissant lors de l'intégration d'un algorithme d'apprentissage pour le traitement de données complexes. Dans le domaine médical, l'étude de problèmes à classes fortement déséquilibrées m'a permis de développer mes travaux sur les forêts d'arbres de décisions flous et de proposer un nouveau modèle de construction et d'utilisation de telles forêts.

Finalement, le video mining pour l'extraction de descripteurs sémantiques de haut niveau m'a ainsi donné l'occasion de réaliser un véritable passage à l'échelle des algorithmes d'apprentissage artificiel que je proposais tout en me permettant de confronter mes résultats à d'autres équipes de recherche dans le cadre d'une compétition internationale. Cela m'a aussi permis d'adapter les forêts d'arbres de décision flous afin de les utiliser pour obtenir un ordonnancement de résultats au lieu d'une simple classification.

4.6 Références

Les travaux de ce chapitre sont issus de recherches conjointes, menées avec d'autres chercheurs. Ils ont donné lieu aux publications suivantes :

- travaux sur le langage FuSOQL et l'application aux données spatiales, avec Nara Martini Bigolin : [69, 70] ;
- travaux sur l'extraction de connaissances à partir de bases de données multi-dimensionnelles, avec Anne Laurent et Stéphane Gançarski : [74, 76] ;
- travaux sur la caractérisation de l'observance médicale, avec le Dr. Alain Lurie : [5] ;
- travaux en video mining, avec Marcin Detyniecki : [10, 42, 50, 79, 80, 90, 92].

Chapitre 5

Travaux des thèses supervisées

5.1 Introduction

Dans cette partie, je fais une présentation succincte des travaux des étudiants que j'ai co-encadrés en thèse. Je rappelle les thèmes principaux sur lesquels ils ont mené leurs travaux et je présente les publications communes que nous avons faites durant leur thèse.

5.2 Thèse de Thanh Ha Dang

Sujet : Mesures de discrimination et leurs applications en apprentissage inductif

Date de soutenance : le 10 juillet 2007.

Directrice de thèse : Bernadette Bouchon-Meunier

Résumé : Le sujet de cette thèse fait suite aux travaux de ma propre thèse et porte sur l'étude des mesures de discrimination, en particulier pour la construction des arbres de décision flous. Dans ce travail, Thanh Ha a proposé une extension du modèle hiérarchique des mesures pour l'appliquer à des mesures floues et il a étendu l'application de ce modèle à d'autres mesures d'infomation classiques. Il a étudié l'utilisation d'une mesure d'information pour la discrétisation des attributs numériques. Il a ensuite proposé l'utilisation de telles mesures pour l'évaluation de classifieurs et le traitement de valeurs manquantes dans des bases d'apprentissage.

Ces travaux ont donné lieu aux publications suivantes : [31, 32, 37, 38].

5.3 Thèse de Thomas Delavallade

Sujet : Évaluation des risques de crise, appliquée à la détection des conflits armés intra-étatiques

Date de soutenance : le 6 décembre 2007

Directrice de thèse : Bernadette Bouchon-Meunier

Thèse CIFRE avec Thalés (co-encadrant industriel : Philippe Capet).

Résumé : Cette thèse a pour cadre l'évaluation des risques et leur traitement. Pour cela, Thomas a réalisé une étude de la chaîne d'apprentissage dans son ensemble, du prétraitement des données au choix du classifieur le mieux adapté à les traiter selon le problème considéré. Il a proposé une analyse comparative empirique des méthodes de traitement des données manquantes et il a étudié les méthodes de sélection d'attributs. Cela l'a amené à proposer un nouveau modèle

d'évaluation des risques qu'il a pu ensuite appliqué dans le domaine de la détection des conflits armés intra-étatiques.

Ces travaux ont donné lieu à la publication suivante : [36].

5.4 Thèse de Marc Damez

Sujet : De l'apprentissage artificiel pour l'apprentissage humain : de la récolte de traces à la modélisation utilisateur

Date de soutenance : le 18 septembre 2008

Directrice de thèse : Bernadette Bouchon-Meunier

Thèse CIFRE avec Vivendi Publishing.

Résumé : Cette thèse a pour cadre l'utilisation de techniques d'apprentissage artificiel pour l'apprentissage humain par la prise en compte des interactions homme-machine. Pour cela, Marc a réalisé un état de l'art lui permettant de faire ressortir les différents éléments intervenants dans ce type d'interactions et il a étudié différentes approches pour récolter des traces d'utilisations, les visualiser et les analyser. Cela l'a amené à proposer ensuite une nouvelle méthode de récolte de traces d'interaction homme-machine et de nouveaux outils originaux d'analyses et de visualisation de ces traces. Il a, de plus, réalisé une étude et une application des approches d'apprentissage artificiel intéressante à mettre en œuvre dans ce type de problèmes.

Ces travaux ont donné lieu à la publication suivante : [34].

5.5 Thèse de Tri Duc Tran

Sujet : Conception et développement d'un assistant intelligent pour un accompagnement conatif des élèves en difficulté

Date de soutenance : le 25 janvier 2010

Directrice de thèse : Bernadette Bouchon-Meunier

Thèse CIFRE avec ILObject (co-encadrant industriel : Georges-Marie Putois)

Résumé : Cette thèse a pour cadre le milieu éducatif et la réalisation d'un assistant intelligent pour aider les élèves en difficulté lors de leur apprentissage. Tri Duc a, tout d'abord, réalisé un état de l'art sur l'usage d'agents intelligents pour l'éducation afin d'en dégager une taxonomie d'agents lui permettant de faire ressortir les propriétés essentielles et novatrices qu'il convient d'intégrer à de tels agents. Il a ensuite proposé un modèle général de conception d'assistant intelligent et personnel pour aider les apprenants. Il a pu mettre en œuvre et valider son modèle dans une expérimentation pour aider des élèves en difficulté.

Ces travaux ont donné lieu aux publications suivantes : [26, 27].

5.6 Thèse de Jean-René Coffi

Sujet : Traitement adaptatif d'événements complexes pour la protection d'infrastructures critiques

Date de soutenance : prévue fin 2011

Directrice de thèse : Bernadette Bouchon-Meunier

Thèse CIFRE avec Thalés TRT (co-encadrant industriel : Nicolas Museux)

Résumé : Cette thèse a pour cadre le traitement d'événements temporels avec, en particulier, une application pour la détection d'alarmes. L'utilisation de techniques d'apprentissage artificiel

est envisagée dans ce travail afin de permettre la génération automatique de règles de détection et l'enrichissement de bases de règles temporelles et causales.

5.7 Conclusion

Durant ces années, j'ai eu l'opportunité de co-encadrer plusieurs thèses qui avaient toutes pour point commun le domaine de l'apprentissage artificiel et la prise en compte de données réelles, imparfaites, et imprécises.

Tous ces doctorants, avec lesquels j'ai eu le plaisir de travailler, ont toujours su aller au delà du sujet qui leur avait été proposé pour l'imprégner de leurs propres idées et réaliser ainsi d'excellentes thèses, comme l'ont souligné leurs jurys de thèse. Ils m'ont ainsi permis d'obtenir des réponses concrètes sur les approches d'apprentissage artificiel qu'ils ont étudiées et mises en œuvre.

D'un autre point de vue, la collaboration que ces thèses m'ont permis d'avoir avec eux, par leur fraîcheur d'esprit, leur enthousiasme et leurs qualités scientifiques m'a ainsi énormément apporté pour ma propre réflexion et mes propres recherches.

Conclusion et perspectives

Dans ce mémoire, j'ai présenté les travaux que j'ai pu mener, seul, en collaborant avec d'autres collègues ou en co-encadrant des thèses, depuis la fin de ma thèse d'université. Dans la continuité de ce que j'avais réalisé pour mon doctorat, j'ai poursuivi durant ces années les travaux en apprentissage artificiel et leur mise en œuvre pour un mécanisme de raisonnement flou. Le fil conducteur principal de ces recherches a été l'étude et la proposition de méthodes permettant d'étendre des algorithmes classiques pour les doter d'une meilleure capacité de prise en compte des données issues du monde réel, qui sont, par essence, numériques, imparfaites, imprécises, incertaines, ou incomplètes. Ainsi, les thèmes de recherche pour intégrer la théorie des sous-ensembles flous dans l'apprentissage artificiel sont multiples : traiter les données, construire les règles, adapter les modèles d'apprentissage, et utiliser les connaissances ainsi générées. Durant ces années, j'ai effectué des recherches dans chacun de ces thèmes.

Le premier axe de mes recherches m'a conduit à poursuivre l'étude que j'avais débutée durant ma thèse sur les mesures de discrimination, leurs propriétés et leur rôle pour la sélection et l'ordonnancement d'attributs flous dans les algorithmes d'apprentissage. Cela m'a amené à généraliser le modèle hiérarchique de fonctions que j'avais proposé, en suivant une voie différente, mais à mon avis plus complète, de celle que j'ai étudiée en co-encadrant Thanh Ha Dang durant sa thèse.

En parallèle, les applications concrètes (dans le domaine médical, ou dans l'extraction de descripteurs à partir de vidéo) sur lesquelles j'ai travaillé m'ont amené à étudier plus en détails les systèmes de combinaison de modèles flous à base d'ensembles de classifieurs, et à proposer un modèle de construction et d'utilisation de forêts d'arbres de décision flous qui peut s'appliquer dans les problèmes où les distributions des classes sont très déséquilibrées.

Un autre axe important de mes recherches a été d'étudier d'autres apports possibles de l'apprentissage artificiel pour l'amélioration des mécanismes de raisonnement flou. Ainsi, j'ai mené des recherches sur la comparaison de deux algorithmes d'apprentissage artificiel utilisés pour la construction de règles floues. En faisant ressortir les points communs et les différences d'un algorithme de génération de règles d'association et d'un algorithme de construction d'arbres de décision flous, il est alors possible de proposer et d'envisager des améliorations de chacun des deux algorithmes. D'autre part, en constatant l'incomplétude des bases de règles susceptibles d'être construites par apprentissage artificiel, j'ai été amené à étudier le mécanisme de raisonnement flou par interpolation, tout à fait adapté à l'utilisation de telles connaissances incomplètes, et à proposer une approche pour l'utiliser dans un cadre multi-prémisses.

Toutes les recherches que j'ai menées ont toujours eu pour cadre un domaine applicatif. Cela m'a permis ainsi de valider les points théoriques que j'étudiais et de découvrir de nouveaux objectifs théoriques à étudier. En particulier, je veux mettre en exergue les trois domaines principaux dans lesquels j'ai pu réaliser des études réelles de data mining.

Le premier domaine applicatif, les bases de données géographiques ou multidimensionnelles,

m'a permis d'étudier le positionnement de l'algorithme d'apprentissage dans la chaîne de traitement complète mise en œuvre pour extraire de la connaissance à partir d'une base de données. Ainsi, j'ai eu l'occasion de me confronter aux difficultés inhérentes à l'intégration d'un algorithme d'apprentissage dans une telle chaîne de traitement de données complexes.

Pour le deuxième domaine applicatif, le domaine médical et la caractérisation de patients pour deux pathologies comme l'asthme et les maladies cardiaques, je me suis orienté vers l'étude des problèmes à classes fortement déséquilibrées pour lesquels j'ai développé mes travaux sur les forêts d'arbres de décisions flous. Cela m'a permis de proposer un nouveau modèle de construction et d'utilisation de telles forêts.

Finalement, le domaine du vidéo mining, et l'extraction de descripteurs sémantiques de haut niveau à partir de vidéos, est un des domaines auxquels j'ai consacré une part importante de recherche. Il m'a ainsi permis de réaliser un véritable passage à l'échelle des algorithmes d'apprentissage artificiel que je proposais tout en me permettant de confronter mes résultats à d'autres équipes de recherche dans le cadre d'une compétition internationale. D'autre part, ce domaine m'a amené à adapter les forêts d'arbres de décision flous afin de les utiliser pour obtenir un ordonnancement de résultats au lieu d'une simple classification.

Dans une moindre mesure, durant ces années, j'ai aussi eu l'opportunité de participer à des projets de recherche ce qui m'a permis d'alimenter mes recherches fondamentales sur l'apprentissage artificiel et le raisonnement flou.

Perspectives de recherches

Chacun des points que j'ai présentés dans ce mémoire offre, de mon point de vue, de nombreuses perspectives de recherches qui mériteront d'être poursuivies.

Mesures et sélection d'attributs

Il reste encore beaucoup de points à étudier sur les mesures de discrimination et la sélection d'attributs ou leur ordonnancement dans un processus d'apprentissage artificiel. Tout d'abord, l'application du modèle hiérarchique pour d'autres mesures reste à développer plus avant. Il sera alors possible de dresser une cartographie plus complète des mesures, de leurs apports, et de leur pertinence selon l'algorithme d'apprentissage dans lequel on l'utilise. Les premières réponses que j'ai pu apporter à la question principale qui se pose (comment choisir une mesure de discrimination en fonction du problème à traiter) restent encore à développer et demandent la poursuite des recherches dans ce sens afin d'avoir une vue plus complète de la liaison mesures / problèmes.

Toujours pour une meilleure connaissance des mesures de discrimination et leur extension dans le cadre du flou, un développement des recherches est aussi d'étudier l'extension des mesures non seulement sous l'angle de la mesure elle-même, mais aussi en raisonnant sur la notion d'événement conditionnel à partir de laquelle elle est définie.

Combinaison de modèles flous en apprentissage

L'étude des ensembles de classifieurs flous, menée par le biais de la construction et de l'utilisation des forêts d'arbres de décision flous, reste encore à développer et promet d'intéressantes perspectives de recherches. Ainsi, tout d'abord, il reste à compléter les travaux que j'ai présentés

ici par une gamme d'expérimentations sur des bases d'apprentissage qui permettrait d'avoir une meilleure connaissance de ce type de modèles flous.

La recherche de moyens théoriques et pratiques de réduire la taille des forêts d'arbres flous reste aussi à développer plus avant. Par exemple, en réfléchissant sur la génération des arbres d'une forêt ou sa réduction par des mécanismes d'élagage des arbres de la forêt.

D'autre part, le succès des ensembles d'arbres classiques (ou aléatoires) dans le domaine de l'ordonnancement offre des perspectives très intéressantes qu'il serait sûrement fructueux de développer pour les forêts d'arbres flous (comme on en a déjà eu la preuve de l'intérêt dans les applications que nous avons réalisées pour le challenge TRECVid).

Modèles pour le raisonnement flou

Comme on a pu le montrer pour l'extraction de connaissances à partir de données, le choix de l'algorithme d'apprentissage est lié aux données à traiter ainsi qu'aux connaissances que l'on souhaite générer. De même, l'algorithme d'apprentissage est lié au modèle de raisonnement flou que l'on désire mettre en œuvre. On a ainsi vu qu'il était possible d'exploiter une base de règles incomplètes générées par un algorithme d'apprentissage en utilisant un raisonnement interpolatif. Les recherches de ce type de liens entre l'algorithme et le modèle de raisonnement sont encore à leur début et demandent à être développées.

Ainsi, par exemple, il est possible d'enrichir la base de règles générées en dirigeant l'algorithme d'apprentissage afin qu'il se consacre à l'apprentissage des zones inconnues de l'espace décrit par les règles. Des recherches en ce sens doivent encore être menées pour étudier la meilleure façon d'intégrer dans l'algorithme une telle prise en compte de l'espace des règles.

Et plus généralement...

Les outils flous offrent une meilleure prise en compte des données réelles que l'on rencontre dans les applications courantes de nos jours. Dans ce mémoire, je me suis attaché à montrer de tels apports dans des cadres bien spécifiques de l'apprentissage artificiel et du raisonnement flou.

Mais d'autres perspectives restent encore à étudier. Par exemple, la prise en compte de connaissances du domaine afin d'améliorer l'apprentissage sur des données est un domaine encore à explorer pour la prise en compte de connaissances imprécises. Ainsi, l'apprentissage artificiel combinant des données brutes et des graphes ou des hiérarchies sur ces données (en tant que méta-connaissances du domaine) est une voie de recherche qu'il va être intéressant de creuser avec la théorie des sous-ensembles flous.

De même, l'étude des mécanismes d'apprentissage incrémentaux et la prise en compte de données temporelles est aussi une perspective à développer car peu de travaux actuels considèrent un tel apprentissage à partir de données imprécises.

Enfin, les apports possibles de la théorie des sous-ensembles flous sont très nombreux et beaucoup sont encore à être étudiés.

Bibliographie

[1] N. Abu-Halaweh and R. Harrison. Practical fuzzy decision trees. In *Proc. of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, pages 211 – 216, Nashville, (USA), March 2009.

[2] N. Abu-Halaweh and R. Harrison. Rule set reduction in fuzzy decision trees. In *Proc. of the 28th NAFIPS Annual Conference*, pages 1–4, Cincinnati, Ohio (USA), June 2009.

[3] J. Aczél and Z. Daróczy. *On Measures of Information and their Characterizations*, volume 115 of *Mathematics in Science and Engineering*. Academic Press, New York, 1975.

[4] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In *Proc. of the 13th Int. Conference on Data Engineering*, pages 232–243, Birmingham, U.K., April 1997.

[5] F. Bancilhon, C. Delobel, and P. Kanellakis. *Building an Object-Oriented Databases Systems : The story of 02.* Morgan Kaufmann, 1992.

[6] P. Baranyi, T. D. Gedeon, and L. T. Kóczy. A general for fuzzy rule interpolation : Specialized for crisp triangular and trapezoidal rules. In *EUFIT'95*, pages 99–102, 1995.

[7] P. Baranyi, D. Tikk, Y. Yam, and L. T. Kóczy. Investigation of a new α-cut based fuzzy interpolation method. Technical Report CUHK-MAE-99-06, The Chinese University of Hong Kong, 1999.

[8] P. Baranyi, Y. Yam, and L. T. Kóczy. Multi-variables singular value based rule interpolation. In *Proc. of the SMC'97 Conference : Computational, Cybernetics and Simulation*, pages 1598–1603, 1997.

[9] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms : Bagging, boosting, and variants. *Machine Learning*, 36 :105–139, 1999.

[10] U. Bodenhofer. Binary ordering-based modifiers. In *Proceedings of the 9th International Conference IPMU*, pages 1953–1959, Annecy, France, 2002.

[11] P. P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares. A fuzzy random forest : Fundamental for design and construction. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'08)*, pages 1231–1238, Malaga, Spain, July 2008.

[12] B. Bouchon-Meunier, G. Coletti, and C. Marsala. Independence and possibilistic conditioning. In *Proc. of the Conf. on Partial Knowledge and Uncertainty : Independence, Conditioning, Inference*, Roma, Italy, May 2000. (Extended abstract).

[13] B. Bouchon-Meunier, G. Coletti, and C. Marsala. Possibilistic conditional events. In *Proc. of the 8th IPMU'00 Conf.*, volume 3, pages 1561–1566, Madrid, Spain, June 2000.

[14] B. Bouchon-Meunier, G. Coletti, and C. Marsala. Conditional possibility and necessity. In B. Bouchon-Meunier, J. Gutiérrez-Ríos, L. Magdalena, and R. Yager, editors, *Technologies for Constructing Intelligent Systems*. Springer, 2002.

[15] B. Bouchon-Meunier, G. Coletti, and C. Marsala. Independence and possibilistic conditioning. *Annals of Mathematics and Artificial Intelligence*, 35(1-4) :107–123, May 2002.

[16] B. Bouchon-Meunier, T.-H. Dang, and C. Marsala. Comparison of techniques for the construction of decision trees. In *Proc. of the 13th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE'04)*, pages 58–62, Nice, France, July 2004.

[17] B. Bouchon-Meunier, J. Delechamp, C. Marsala, N. Mellouli, M. Rifqi, and L. Zerrouki. Analogy and interpolation in the case of sparse rules. In *Eurofuse - SIC'99*, pages 132–136, Budapest, Hungary, 1999.

[18] B. Bouchon-Meunier, J. Delechamp, C. Marsala, N. Mellouli, M. Rifqi, and L. Zerrouki. Raisonnement interpolatif à partir de schéma analogique flou. In *Rencontres JNMR'98, Journées Nationales sur les Modèles de Raisonnements*, Paris, France, Mars 1999. (Publication Web).

[19] B. Bouchon-Meunier, J. Delechamp, C. Marsala, and M. Rifqi. Several forms of fuzzy analogical reasoning. In *Proceeding of the Sixth IEEE International Conference on Fuzzy Systems, FUZZ'IEEE'97*, volume 1, pages 45–50, Barcelona, Spain, July 1997.

[20] B. Bouchon-Meunier, J. Delechamp, C. Marsala, and M. Rifqi. Analogical reasoning as a basis for various forms of approximate reasoning. In B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, editors, *Uncertainty in Intelligent and Information Systems*. World Scientific, *in press*.

[21] B. Bouchon-Meunier, D. Dubois, C. Marsala, H. Prade, and L. Ughetto. A comparative view of interpolation methods between sparse fuzzy rules. In *Proc. of the IFSA'01 World Congress*, pages 2499–2504, Vancouver, Canada, July 2001.

[22] B. Bouchon-Meunier and C. Marsala. Improvement of the interpretability of fuzzy rules constructed by means of fuzzy decision tree based systems. In *Proceedings of the FSTA'2002 conference*, 2002. extended abstract.

[23] B. Bouchon-Meunier and C. Marsala. Measures of discrimination for the construction of fuzzy decision trees. In *Proc. of the FIP'03 conference*, pages 709–714, Beijing, China, March 2003.

[24] B. Bouchon-Meunier, C. Marsala, and M. Ramdani. Learning from imperfect data. In D. Dubois, H. Prade, and R. R. Yager, editors, *Fuzzy Information Engineering : a Guided Tour of Applications*, pages 139–148. John Wileys and Sons, 1997.

[25] B. Bouchon-Meunier, C. Marsala, and M. Rifqi. Interpolative reasoning based on graduality. In *Proc. of the 9th IEEE Int. Conf. on Fuzzy Systems*, pages 483–487, San Antonio, Texas, May 2000.

[26] X. Boyen and L. Wehenkel. Automatic induction of fuzzy decision tree and its application to power system security assessmen. *Fuzzy Sets and Systems*, 102(1) :3–19, 1999.

[27] L. Breiman. Bagging predictors. *Machine Learning*, 24 :123–140, 1996.

[28] L. Breiman. Random forests. *Machine Learning*, 45 :5–32, 2001.

[29] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification And Regression Trees*. Chapman and Hall, New York, 1984.

[30] B. Chandra and P. Paul Varghese. Fuzzifying Gini index based decision trees. *Expert Systems with Applications*, 36 :8549–8559, 2009.

[31] K. Cios and L. Sztandera. Continuous ID3 algorithm with fuzzy entropy measures. In *Proceedings of the first International IEEE Conference on Fuzzy Systems*, San Diego, 1992.

[32] E. Codd, S. Codd, and C. Salley. Providing OLAP (on-line analytical processing) to user-analysts : an IT mandate. Technical report, Arbor Software Corporation, 1993.

[33] K. Crockett, Z. Bandar, and D. Mclean. Growing a fuzzy decision forest. In *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, pages 614–617, December 2001.

[34] V. Cross and Y. Sudkamp. Geometric compatibility modification. *Fuzzy Sets and Systems*, 84(3) :283–299, 1996.

[35] M. Damez, T.-H. Dang, C. Marsala, and B. Bouchon-Meunier. Fuzzy decision tree for user modeling from human-computer interactions. In *Proc. of the 5th International Conference on Human System Learning*, page (ï¿½ paraï¿½tre), Marrakech (Maroc), Novembre 2005.

[36] T.-H. Dang. *Mesures de discrimination et leurs applications en apprentissage inductif*. PhD thesis, Université Pierre et Marie Curie - Paris 6, Paris, Juillet 2007.

[37] T.-H. Dang, B. Bouchon-Meunier, and C. Marsala. Measures of information for inductive learning. In *Proc. of the 10th IPMU'04 Conf.*, pages 1495–1502, Perugia, Italy, July 2004.

[38] T.-H. Dang and C. Marsala. Extension of hierarchical model for fuzzy measures of discrimination. In *Proc. of the 11th IPMU'06 Conf.*, pages 1284–1291, Paris, France, July 2006.

[39] T.-H. Dang, C. Marsala, B. Bouchon-Meunier, and A. Boucher. Discrimination-based criteria for the evaluation of classifiers. In *Proc. of the 7th Int. Conf. FQAS'06*, pages 552–563, Milano, Italy, June 2006.

[40] T. Delavallade, B. Bouchon-Meunier, C. Marsala, and P. Capet. Country risk ratings : A new methodology to asses internal conflicts risk. In *Proc. of the Deloitte Risk Management Conference*, Anvers, Belgique, May 2005.

[41] M. Detyniecki and C. Marsala. Fuzzy inductive learning for multimedia mining. In *Proc. of the EUSFLAT'01 conference*, pages 390–393, Leicester (UK), September 2001.

[42] M. Detyniecki and C. Marsala. Fuzzy multimedia mining applied to video news. In *Proceedings of the IPMU'02 Conference*, pages 1001–1008, Annecy, France, July 2002.

[43] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization. *Machine Learning*, 40 :139–157, 2000.

[44] D. Dubois and H. Prade. On fuzzy interpolation. In *Proceedings of the 3rd International Conference on Fuzzy Logic & Neural Networks*, pages 353–354, Iizuka, Japan, August 1994.

[45] B. Efron. Bootstrap methods : Another look at the jackknife. *The Annals of Statistics*, 7(1) :1–26, 1979.

[46] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, CRC Press, 1993.

[47] B. Efron and R. Tibshirani. Cross-validation and the bootstrap : Estimating the error rate of a prediction rule. Technical Report TR-477, Dept. of Statistics, Stanford University, 1995.

[48] B. Efron and R. Tibshirani. Improvements on cross-validation : The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438) :548–560, 1997.

[49] M. Ester, H.-P. Kriegel, and J. Sander. Spatial data mining : A database approach. *Proc. 5th Symp. on Spatial Databases, Berlin, Germany*, 1997.

[50] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3) :37–54, Fall 1996.

[51] A. Fotheringham and S. P. Rogerson. *Spatial analysis and GIS : applications in GIS*. London Washington, 1993.

[52] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4 :933–969, 2003.

[53] Y. Freund and R. Shapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[54] Y. Freund and R. Shapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5) :771–780, September 1999.

[55] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1) :3–42, April 2006.

[56] S. N. Ghazavi and T. W. Liao. Medical data mining by fuzzy modeling with selected features. *Artificial Intelligence in Medicine*, 43 :195–206, 2008.

[57] F. Gueyffier, F. Boutitie, J. P. Boissel, J. Coope, J. Cutler, T. Ekbom, R. Fagard, L. Friedman, H. M. Perry, and S. Pocock. INDANA : a meta-analysis on individual patient data in hypertension. protocol and preliminary results. *Thérapie*, 50(4) :353–62, 1995.

[58] R. C. Guiaşu and S. Guiaşu. Conditional and weighted measures of ecological diversity. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(3) :283–300, 2003.

[59] J. Han, K. Koperski, and N. Stefanovic. Geominer : A system prototype for spatial data mining. *Proc. 1997 ACM-SIGMOD Int'l Conf. on Management of Data(SIGMOD'97), Tucson, Arizona*, May 1997.

[60] T. K. Ho. *A Theory of Multiple Classifier Systems And Its Application to Visual Word Recognigtion*. PhD thesis, Faculty of the Graduate School of State University of New York at Buffalo, May 1992.

[61] T. K. Ho. Random decision forests. In *Proc. of the Third Int. Conf. on Document Analysis and Recognition*, volume 1, pages 278–282, 1995.

[62] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Learning*, 20(8) :832–844, August 1998.

[63] S. Jaillet, A. Laurent, and M. Teisseire. Sequential patterns for text categorization. *Intelligent Data Analysis*, 10(3) :199–214, 2006.

[64] C. Z. Janikow. Fuzzy decision trees : Issues and methods. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1) :1–14, 1998.

[65] C. Z. Janikow. Fuzzy decision forest. In *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS'03)*, pages 480–483, July 2003.

[66] C. Z. Janikow and M. Faifer. Fuzzy decision forest. In *Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS'00)*, pages 218–221, July 2000.

[67] S. Jenei. A new approach for interpolation and extrapolation of compact fuzzy quantities. In *Proc. of the 21th Linz Sem. on Fuzzy Sets theory*, pages 13–18, 2000.

[68] L. T. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9 :197–225, 1993.

[69] L. T. Kóczy and K. Hirota. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Science*, 71 :169–201, 1993.

[70] K. Koperski, J. Han, and J. Adhikary. Mining knowledge in geographic data. *Comm. ACM (to appear)*, 1998.

[71] A. Laurent. Generating fuzzy summaries from multidimensional databases. In *Proc. of the Intelligent Data Analysis (IDA) Conference*, pages 24–33, 2001.

[72] A. Laurent. *Bases de données multidimensionnelles floues et leur utilisation pour la fouille de données*. PhD thesis, Université P. et M. Curie, Paris, France, September 2002.

[73] A. Laurent. FUB et FUB Miner : deux systèmes pour la représentation, la manipulation et la fouille de données multidimensionnelles floues. *revue I3 - Information, Interaction, Intelligence*, 3(1) :37–83, 2003.

[74] A. Laurent, B. Bouchon-Meunier, A. Doucet, S. Gançarski, and C. Marsala. Fuzzy data mining from multidimensional databases. In J. Kacprzyk, editor, *Proc. of the Int. Symposium on Computational Intelligence - ISCI'00*, volume 54 of *Studies CI*, pages 278–283, Kosice, Slovakia, August 2000. Springer Verlag.

[75] A. Laurent, S. Gançarski, and C. Marsala. Coopération entre un système d'extraction de connaissances floues et un système de gestion de base de données multidimensionnelles. In *Actes de la conférence LFA'98*, pages 325–332, La Rochelle, France, Octobre 2000. Cépaduès éditions.

[76] A. Laurent, C. Marsala, and B. Bouchon-Meunier. Improvement of the interpretability of fuzzy rule based systems : Quantifiers, similarities and aggregators. In J. Lawry, J. Shanahan, and A. Ralescu, editors, *Modelling with Words*, volume 2873 of *Lecture Notes in Computer Science*, pages 102–123. Springer-Verlag, Heidelberg, 2003.

[77] W. Lu, J. Han, and B. C. Ooi. Discovery of general knowledge in large spatial databases. *Proc. of 1993 Far East Workshop on Geographic Information Systems- (FEGIS'93), Singapore*, pages 275–289, June 1993.

[78] A. Lurie, C. Marsala, S. Hartley, B. Bouchon-Meunier, and D. Dusser. Patients' perception of asthma severity. *Respiratory Medecine*, 101(10) :2145–2152, October 2007.

[79] C. Marsala. *Apprentissage inductif en présence de données imprécises : construction et utilisation d'arbres de décision flous.* Thèse de doctorat, Université Pierre et Marie Curie, Paris, France, Janvier 1998. Rapport LIP6 n^o 1998/014.

[80] C. Marsala. Fuzzy decision trees to help flexible querying. *Kybernetika*, 36(6) :689–705, 2000.

[81] C. Marsala. Data mining with ensembles of fuzzy decision trees. In *Proceedings of the Symposium on Computational Intelligence and Data Mining*, pages 348–354, Nashville, USA, March 2009.

[82] C. Marsala and B. Bouchon-Meunier. Forest of fuzzy decision trees. In M. Mareš, R. Mesiar, V. Novák, J. Ramik, and A. Stupňanová, editors, *Proceedings of the Seventh International Fuzzy Systems Association World Congress*, volume 1, pages 369–374, Prague, Czech Republic, June 1997.

[83] C. Marsala and B. Bouchon-Meunier. Interpolative reasoning with multi-variable rules. In *Proc. of the IFSA'01 World Congress*, pages 2476–2481, Vancouver, Canada, July 2001.

[84] C. Marsala and B. Bouchon-Meunier. Choice of a method for the construction of fuzzy decision trees. In *Proc. of the IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE*, pages 584–589, St Louis, USA, May 2003.

[85] C. Marsala and B. Bouchon-Meunier. Selection of attributes for fuzzy decision trees. In E. Hullermeier, F. Klawon, and R. Kruse, editors, *Workshop "Soft Computing for Information Mining", in 27th conference on Artificial Intelligence*, Ulm, Germany, September 2004.

[86] C. Marsala and B. Bouchon-Meunier. Ranking attributes to build fuzzy decision trees : a comparative study of measures. In *Proc. of the IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE*, pages 1777–1783, Vancouver, Canada, July 2006.

[87] C. Marsala, B. Bouchon-Meunier, and A. Ramer. Hierarchical model for discrimination measures. In *Proc. of the IFSA'99 World Congress*, pages 339–343, Taiwan, 1999.

[88] C. Marsala and M. Detyniecki. University of Paris 6 at TRECVid 2005 : High-level feature extraction. In *TREC Video Retrieval Evaluation Online Proceedings*, November 2005. http ://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[89] C. Marsala and M. Detyniecki. University of Paris 6 at TRECVid 2006 : Forest of fuzzy decision trees for high-level feature extraction. In *TREC Video Retrieval Evaluation Online Proceedings*, November 2006. http ://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[90] C. Marsala and M. Detyniecki. UPMC-LIP6 at TRECVid'08 : Balanced and unbalanced forests of fuzzy decision trees for high-level feature detection. In *TREC Video Retrieval Evaluation Online Proceedings*, November 2008. http ://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[91] C. Marsala and M. Detyniecki. High scale fuzzy video mining. In A. Laurent and M.-J. Lesot, editors, *Scalable Fuzzy Algorithms for Data Management and Analysis : Methods and Design*, chapter 15, pages 365–378. IGI, 2009.

[92] C. Marsala, M. Detyniecki, N. Usunier, and M. R. Amini. High-level feature detection with forests of fuzzy decision trees combined with the rankboost algorithm. In *TREC Video Retrieval Evaluation Online Proceedings*, November 2007. http ://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[93] C. Marsala and N. Martini Bigolin. Spatial data mining with fuzzy decision trees. In N. F. F. Ebecken, editor, *Data Mining*, pages 235–248. WIT Press, 1998. Proceedings of the International Conference on Data Mining, Rio de Janeiro, Sept. 1998.

[94] N. Martini Bigolin. *Méthodes pour la découverte de connaissances à partir d'une base de données spatiales orientée objet : le système LARECOS*. PhD thesis, Université P. et M. Curie, Paris 6, 1999.

[95] N. Martini Bigolin and C. Marsala. Fuzzy spatial OQL for fuzzy knowledge discovery. In *Principles of Data Mining and Knowledge Discovery*, volume 1510 of *Lecture Notes in*

AI, pages 246–254. Springer Verlag, 1998. Proceedings of the 2nd European Symposium PKDD'98.

[96] D. Nauck and R. Kruse. Obtaining interpretable fuzzy classification rules from medical data. *Artificial Intelligence in Medicine*, 16 :149–169, 1999.

[97] C. Olaru and L. Wehenkel. A complete fuzzy decision tree technique. *Fuzzy Sets and Systems*, 138(2) :221–254, September 2003.

[98] W. Z. Qiao, M. Masaharu, and Y. Shi. An improvement to kóczy and hirota's interpolative reasoning in sparse fuzzy rule bases. *International Journal of Approximate Reasoning*, 15 :185–201, 1996.

[99] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1) :86–106, 1986.

[100] M. Ramdani. Une approche floue pour traiter les valeurs numériques en apprentissage. In *Journées Francophones d'apprentissage et d'explication des connaissances*, 1992.

[101] L. Raynal and G. Schorter. Geo2. Technical report, COGIT - IGN, 1995.

[102] M. Rifqi. *Mesures de comparaison, typicalité et classification d'objets flous : théorie et pratique.* PhD thesis, Université P. et M. Curie, Paris, France, Décembre 1996. Aussi publiée en rapport du LAFORIA-IBP n^o TH96/15.

[103] R. Schapire. The strengh of weak learnability. *Machine Learning*, 5 :197–227, 1990.

[104] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR '06 : Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[105] A. F. Smeaton, P. Over, and W. Kraaij. High-Level Feature Detection from Video in TRECVid : a 5-Year Retrospective of Achievements. In A. Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[106] T. D. Tran, B. Bouchon-Meunier, C. Marsala, and G.-M. Putois. An intelligent assistant to support students and to prevent them from dropout. In *Proceedings of the 1st International Conference on Computer Supported Education (CSEDU)*, 2009.

[107] T.-D. Tran, C. Marsala, B. Bouchon-Meunier, and G.-M. Putois. A model to manage learner's motivation : a use-case for an academic schooling intelligent assistant. In *Proceedings of 4rd European Conference on Technology Enhanced Learning (EC-TEL 09)*, 2009.

[108] L. Ughetto, D. Dubois, and H. Prade. Interpolation linéaire par ajout de règles dans une base incomplète – Une discussion. In *Actes des 10e rencontres francophones sur la Logique Floue et ses Applications (LFA'00)*, pages 71–78, La Rochelle, France, Nov 2000. Cépaduès Editions.

[109] X. Wang, B. Chen, G. Qian, and F. Ye. On the optimization of fuzzy decision trees. *Fuzzy Sets and Systems*, 112(1) :117–125, May 2000.

[110] R. Weber. Fuzzy-ID3 : A class of methods for automatic knowledge acquisition. In *IIZUKA'92 Proceedings of the 2nd International Conference on Fuzzy Logic*, pages 265–268, 1992.

[111] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1) :183–190, January/February 1988.

[112] Y. Yuan and M. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and systems*, 69 :125–139, 1995.

[113] L. Zadeh. Probability measures of fuzzy events. *Journal Math. Anal. Applic.*, 23, 1968. reprinted in "*Fuzzy Sets and Applications : selected papers by L. A. Zadeh*", R. R. Yager, S. Ovchinnikov, R. M. Tong and H. T. Nguyen eds, pp. 45–51.

Annexes

Autres travaux (depuis 1998)

Participation à des projets

Financements Ministériels

- 2001-2004 : ACI "technologies pour la santé"
  Thème : *Méthodes d'apprentissage pour la prédiction d'un risque cardio-vasculaire*
  Responsable scientifique : M.-C. Jaulent (INSERM ERM 0202)
  Participants : INSERM ERM 0202, LRI, Polytechnique, LIP6, LIFL, SPC.

- 2002-2004 : Projet RNTL (pré-compétitif)
  Thème : *ACEDU : Adaptation du Cartable Electronique à ses Divers Utilisateurs*
  Responsable scientifique : B. Bouchon-Meunier (LIP6)
  Partenaires : Vivendi Universal Éducation France (Éditions Bordas et Éditions Nathan), LIP6, Laboratoire Cognition et Activités Finalisées Paris 8 (CAF).

- 2002-2006 : Projet RNRT
  Thème : *LUTIN : Laboratoire des Usages en Technologies d'Information Numérique*
  Partenaires : Vivendi Universal Éducation France (Éditions Bordas et Éditions Nathan), IRCAM, LIP6, Laboratoires universitaires.

- 2005-2008 : Projet Infomagic
  Thème : *Analyse et fusion d'informations*
  Responsable : Thalès L & J
  Partenaires : pôle de compétitivité CAP Digital.

- 2009-2011 : Projet DOXA
  Thème : *Traitement automatique des opinions et des sentiments*
  Responsable : Thalès L & J
  Partenaires : pôle de compétitivité CAP Digital.

Financements CNRS

- 1996-2001 : projet de la DRI du CNRS (accord CNRS/CNCPRST)
  Thème : *Connaissances expertes floues et apprentissage*
  Coopération franco-marocaine (accord CNRS/CNCPRST) entre le LAFORIA (ex. LIP6) et la Faculté des Sciences et Techniques de Mohammadia (Maroc).
  Programme de recherche SPI3397 lancé en 1996.

Reconduit en 1997 (accord n°9709), en 1998 (accord n°5188), en 2000 (accord n°8296 entre le LIP6 et la FST de Mohammadia (Maroc)), puis en 2001 (accord n°8296).
Responsable scientifique : B. Bouchon-Meunier (LIP6) et M. Ramdani (FST Mohammedia)
Participants : LAFORIA UPMC (ex. LIP6) et FST de Mohammadia (Maroc)

Autres activités liées à la recherche

Membre de comités d'organisation de congrès

- Membre du comité d'organisation de la 7^e conférence internationale IPMU (Information Processing and Management of Uncertainty in Knowledge-Based Systems), (300 participants), Paris, Juil. 98.
- Co-président du comité d'organisation de la 11^e conférence internationale IPMU (Information Processing and Management of Uncertainty in Knowledge-Based Systems), (350 participants), Paris, Juil. 06.

Organisation de sessions dans des conférences (depuis 2000)

- Chairman de la session "Decision Trees under uncertainty", à la 7e Conférence IPMU'98, Paris, Juil. 98.
- Organisateur et chairman d'une session invitée sur le thème "Fuzzy Inductive Learning", à la 8e Conference IPMU'00, Madrid (Espagne), Juil. 00.
- Co-organisateur avec M. Detyniecki et chairman de la session "Data mining and Multimedia Systems", Conférence EUSFLAT'01, Leicester (UK), Sept. 01.
- Co-organisateur avec M. Detyniecki et chairman de la session "Data mining and Multimedia Systems", Conférence IPMU'02, Annecy, Juil. 02.

Organisation de journées de recherche

- Co-organisateur avec F. d'Alché-Buc et V. Corruble de la journée du GTRA sur l'" Extraction et découverte de connaissances à partir de données structurées, incomplètes, imparfaites ou distribuées.", Sept. 00.
- Organisateur de deux journées de séminaires dans le cadre du pôle oral de la revue *Information - Interaction - Intelligence* du GDR I3 du CNRS. (Mars 2004, et sept. 2004).

Membre de jurys de thèse

- Nara Martini Bigolin, *Méthodes pour la découverte de connaissances à partir d'une base de données spatiales orientée objet : le système LARECOS*. Université Paris VI. Oct. 99.
- Anne Laurent, *Bases de données multidimensionnelles floues et leur utilisation pour la fouille de données*. Université Paris VI. Sept. 02.
- Cristina Olaru, *Contributions to automatic learning - Soft decision tree induction*. Université de Liège (Belgique). Oct. 03.

Membre de comités de programme (depuis 2000)

- Conférence internationale FQAS (Flexible Query Answering Systems)

  - 2000, 2002, 2004, 2006,2009, 2011.
- Conférence internationale ECSQARU (European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty)
  - 2001, 2003.
- Conférence WSC (Online World Conf. on Soft Computing in Industrial Applications) (sur internet)
  - 2002, 2003, 2004, 2006, 2009, 2010.
- Conférence LFA (Rencontres francophones sur la logique floue et ses applications)
  - 2002, 2003, 2007, 2008, 2009, 2010.
- Conférence internationale EUSFLAT (European Society for Fuzzy Logic and Technology)
  - 2003, 2005 (jointe avec LFA).
- Conférence internationale RASC (Recent Advances in Soft Computing)
  - 2002, 2004, 2006.
- Conférence internationale Fuzz-IEEE (IEEE International Conf. on Fuzzy Systems)
  - de 2002 à 2005, puis au titre de reviewer depuis 2006.
- Conférence internationale IPMU (Information Processing and Management of Uncertainty in knowledge-based systems)
  - depuis 2002.

Activités d'édition

- Co-éditeur avec B. Bouchon-Meunier de deux volumes sur la théorie des sous-ensembles flous de la collection I3, Hermès, Jan. 03.
- Associate editor de la revue internationale "Mathware & Soft Computing" (dep. Jan. 02).
- Co-éditeur avec B. Bouchon-Meunier, M. Rifqi, et R. Yager du livre "Uncertainty and intelligent information systems" (World Scientific), 2008.

Charges de recherche

- Suppléant au conseil scientifique du LIP6 (Juin 99 - Mai 06, Jan. 09- aujourd'hui).
- Membre du bureau de direction en tant que secrétaire (élu en 2001, réélu en 2003) de la société européenne de logique floue (EUSFLAT), (Sept. 01 - Sept. 05).
  La société EUSFLAT est une association scientifique européenne qui compte plus de 200 adhérents (en 2005), chercheurs européens en majeure partie.
- Reviewer d'articles pour différentes revues et conférences (depuis 1998) :
  Revues : Int. Journal of Uncertainty, Fuzziness and Knowledge-based Systems, IEEE Trans. on Knowledge and Data Engineering, Iranian Journal of Science and Technology, Fuzzy Sets and Systems, IEEE Trans. on S.M.C.
  Conférences : principalement celles dont je suis dans le comité de programme (FUZZ-IEEE, IPMU, LFA, FQAS, ECSQARU...)

Séminaires et conférences invitées

- Fév. 99, *Mesures de discrimination pour apprentissage inductif*, séminaire de l'équipe de recherche ÉRIC, Université Lyon II.
- Fév. 99, *Arbres de décision et sous-ensembles flous*, séminaire à la Faculté des Sciences et Techniques de Mohammadia, Maroc.

- Sep. 99, *Fuzziness and Data Mining*, conférence invitée, conférence EURO-PRIME for young researchers in Operational Research, Warsaw, Poland.Varsovie (Pologne).
- Mai 00, *Fuzzy Data Mining*, séminaire à l'Université de Perugia. Perugia (Italie).
- Avril 01, *Fuzzy Sets Theory and Data Mining*, série de 5 séminaires de 1h à l'Université de Perugia. Perugia (Italie).
- Juil. 03, *Fuzzy Methods for Inductive Learning*, (avec B. Bouchon-Meunier), BISC Special Seminar, Berkeley (USA).
- Oct. 03, *Arbres de décision flous et leur application au vidéo mining*, séminaire à l'Institut Montefiore, Université de Liège (Belgique).
- Déc. 05, *Fuzzy Data Mining*, série de 3 conférences de 2h en tant que conférencier invité, 5^e International School on Reasoning under Partial Knowledge (Reason Park), Foligno (Italie).
- Mai 10, *Fuzzy Decision Trees : issues, methods and applications*, conférence plénière invitée, 6^e International Conference on Intelligent Systems : Theory and Applications, Rabat (Maroc).

Encadrements d'étudiants (hors thèses)

Stages de fin d'étude (5 mois) (niveau M2 rech. ou M2 pro)

- Anne Laurent. *Extraction de connaissances à partir d'une base de données multidimensionnelles*. Co-encadrement avec B. Bouchon-Meunier, A. Doucet et S. Gançarski. Avr. - Sept. 99. Stage de DEA.
- Laure Mouillet. *Interpolation en logique Floue*. Co-encadrement avec B. Bouchon-Meunier. Avr. - Sept. 01. Stage de DEA.
- Alexandre Pitti. *Combinaison de classifieurs dans le cas de classes déséquilibrées : application à la prévision du risque cardio-vasculaire*. Co-encadrement avec F. d'Alché-Buc. Avr. - Sept. 02. Stage de DEA.
- Thanh Ha Dang. *Mesures d'information et arbres de décision*. Avr. - Sept. 03. Stage de DEA.
- Nicolas Longuet. *Graphe d'association situation-tâche pour la portabilité de dispositifs numériques*. Mai - Sept. 03. Stage de DESS.
- Lionel Yaffi. *Usages d'un terminal*. Mai - Sept. 04. Stage de DESS.

Projet de fin d'étude (4 mois) (niveau M2 pro)

- Réalisation d'un système de simulation de réseau routier sous ILOG Rules. *1 binôme*. Jan.- Avr. 99.
- Réalisation d'un générateur de plans de combinaisons au jeu d'échecs sous ILOG Rules. *1 binôme*. Jan.- Avr. 99.
- Réalisation d'un système de simulation de réseau routier sous ILOG Rules. *1 binôme*. Jan.- Avr. 00.
- Extension de ILOG C++ Rules pour l'implémentation de systèmes à base de connaissances floues. *1 binôme*. Jan.- Avr. 00.
- Extension de ILOG Java Rules pour l'implémentation de systèmes à base de connaissances floues. *1 binôme*. Jan.- Avr. 01.

- Réalisation de packages Java pour la construction et l'utilisation d'arbres de décision. *1 binôme*. Jan.- Avr. 03.
- Implémentation d'un système de programmation de systèmes à base de connaissances. *1 binôme*. Jan.- Avr. 03.
- Réalisation d'une Intelligence Artificielle pour un bot de jeu vidéo. *1 étudiant*. Jan.- Avr. 04.
- Réalisation d'une plateforme de traitement de données vidéo. *4 étudiants*. Jan.- Avr. 04.
- Conception d'un livre électronique. *1 binôme*. Jan.- Avr. 05.
- Réalisation d'une API Java pour la construction incrémentale d'arbres de décision. *1 binôme*. Jan.- Avr. 06.

Stage de fin d'étude de magistère d'informatique (5 mois)

- Aurélien Tabard. *Usages d'un terminal : interfaces et visualiation*. Avr. - Sept. 04.

Liste des publications depuis 1998

Ouvrages édités

1. B. Bouchon-Meunier, C. Marsala, M. Rifqi, and R. Yager, eds. *Uncertainty and Intelligent Information Systems*. World Scientific, 2008. ISBN-13 : 978-9812792341.
2. B. Bouchon-Meunier and C. Marsala, eds. *Logique floue : principes, aide à la décision*, volume 1. Hermès Sciences Publications, 2003.
3. B. Bouchon-Meunier and C. Marsala, eds. *Traitement de données complexes et commande en logique floue*, volume 2. Hermès Sciences Publications, 2003.

Revues internationales

4. M. Detyniecki and C. Marsala. Automatic video annotation with forests of fuzzy decision trees. *Mathware and Soft Computing*, 15(1) :61–74, 2008.
5. A. Lurie, C. Marsala, S. Hartley, B. Bouchon-Meunier, and D. Dusser. Patients' perception of asthma severity. *Respiratory Medecine*, 101(10) :2145–2152, October 2007.
6. B. Bouchon-Meunier, R. Mesiar, C. Marsala, and M. Rifqi. Compositional rule of inference as an analogical scheme. *Fuzzy Sets and Systems*, 138(1) :53–65, August 2003.
7. B. Bouchon-Meunier, G. Coletti, and C. Marsala. Independence and possibilistic conditioning. *Annals of Mathematics and Artificial Intelligence*, 35(1-4) :107–123, May 2002.
8. C. Marsala. Fuzzy decision trees to help flexible querying. *Kybernetika*, 36(6) :689–705, 2000.
9. O. Gascuel, B. Bouchon-Meunier, G. Caraux, P. Gallinari, A. Guénoche, Y. Guermeur, Y. Lechevallier, C. Marsala, L. Miclet, J. Nicolas, R. Nock, M. Ramdani, M. Sebag, B. Tallur, G. Venturini, and P. Vitte. Twelve numerical, symbolic and hybrid supervised classification methods. *Int. Jour. of Pattern Recognition and A. I.*, 12(5) :517–572, 1998.

Chapitres dans des recueils (ouvrages édités)

10. C. Marsala and M. Detyniecki. High scale fuzzy video mining. In A. Laurent and M.-J. Lesot, eds, *Scalable Fuzzy Algorithms for Data Management and Analysis : Methods and Design*, chapter 15, pp. 365–378. IGI, 2009.

11. B. Bouchon-Meunier, M. Detyniecki, M.-J. Lesot, C. Marsala, and M. Rifqi. Real world fuzzy logic applications in data mining and information retrieval. In P.P. Wang, D. Ruan, and E.E. Kerre, editors, *Fuzzy Logic - A Spectrum of Theoretical and Practical Issues*, number 215 in Studies in Fuzziness and Soft Computing, pp. 219–247. Springer, 2007.

12. A. Laurent, C. Marsala, and B. Bouchon-Meunier. Improvement of the interpretability of fuzzy rule based systems : Quantifiers, similarities and aggregators. In J. Lawry, J.G. Shanahan, and A.L. Ralescu, eds, *Modelling with Words*, volume 2873 of *Lecture Notes in Computer Science*, pp. 102–123. Springer-Verlag, Heidelberg, 2003.

13. B. Bouchon-Meunier, C. Marsala, and M. Rifqi. Introduction. In *Logique floue : principes, aide à la décision*, volume 1, pp. 17–39. Hermès Sciences Publications, 2003.

14. B. Bouchon-Meunier and C. Marsala. Méthodes de raisonnement. In *Logique floue : principes, aide à la décision*, volume 1, pp. 121–147. Hermès Sciences Publications, 2003.

15. C. Marsala and B. Bouchon-Meunier. Apprentissage et extraction de connaissances. In *Traitement de données complexes et commande en logique floue*, volume 2, pp. 153–198. Hermès Sciences Publications, 2003.

16. B. Bouchon-Meunier, G. Coletti, and C. Marsala. Conditional possibility and necessity. In B. Bouchon-Meunier, J. Gutiérrez-Ríos, L. Magdalena, and R.R. Yager, eds., *Technologies for Constructing Intelligent Systems*. Springer, 2002.

17. C. Marsala. Fuzzy partitioning methods. In W. Pedrycz, ed., *Granular Computing : an Emerging Paradigm*, Studies in Fuzziness and Soft Computing, pp. 163–186. Springer-Verlag, 2001.

18. C. Marsala, M. Ramdani, D. Zakaria, and M. Toullabi. Fuzzy decision trees to extract features of odorous molecules. In B. Bouchon-Meunier, R.R. Yager, and L.A. Zadeh, eds., *Uncertainty in Intelligent and Information Systems*, volume 20 of *Advances in Fuzzy Systems - Applications and Theory*, pp. 235–249. World Scientific, 2000.

19. B. Bouchon-Meunier and C. Marsala. Learning fuzzy decision rules. In J. Bezdek, D. Dubois, and H. Prade, eds., *Fuzzy Sets in Approximate Reasoning and Information Systems*, volume 3 of *Handbook of Fuzzy Sets*, chapter 4, pp. 279–304. Kluwer Ac. Pub., 1999.

Publications dans des conférences internationales

20. N. Labroche and C. Marsala. Optimization of a fuzzy decision trees forest with artificial ant based clustering. In *Proceedings of the International Conference SOCPAR*, 2010.

21. C. Marsala and B. Bouchon-Meunier. Quality of measures for attribute selection in fuzzy decision trees. In *Proceedings of the International Conference on Fuzzy Systems (WCCI-2010)*, page *to appear*, Barcelona, Spain, July 2010.

22. C. Marsala. Data mining with ensembles of fuzzy decision trees. In *Proceedings of the Symposium on Computational Intelligence and Data Mining*, pp. 348–354, Nashville, USA, March 2009.

23. C. Marsala. A fuzzy decision tree based approach to characterize medical data. In *Proceedings of the IEEE International Conference on Fuzzy Systems (Fuzz-IEEE)*, pp. 1332–1337, Jeju Island, Korea, August 2009.

24. B. Bouchon-Meunier, C. Marsala, and M. Rifqi. Fuzzy analogical model of adaptation for case-based reasoning. In *Proc. of the International Conf. IFSA-EUSFLAT*, 2009.

25. J. Bu, S.-Y. Lao, L. Bai, S. Tollari, and C. Marsala. Goalmouth detection in field-ball game video using fuzzy decision tree. In *International Conference on Image and Graphics (ICIG)*, 2009.

26. Tri Duc Tran, Bernadette Bouchon-Meunier, Christophe Marsala, and Georges-Marie Putois. An intelligent assistant to support students and to prevent them from dropout. In *Proc. of the 1st International Conf. on Computer Supported Education (CSEDU)*, 2009.

27. T.-D. Tran, C. Marsala, B. Bouchon-Meunier, and G.-M. Putois. A model to manage learner's motivation : a use-case for an academic schooling intelligent assistant. In *Proceedings of 4rd European Conference on Technology Enhanced Learning (EC-TEL)*, 2009.

28. M. Detyniecki and C. Marsala. Adaptive acceleration and shot stacking for video rushes summarization. In *Proceeding of the 2nd ACM workshop on Video summarization*, 2008.

29. M. Detyniecki and C. Marsala. Forest of fuzzy decision trees and their application in video mining. In *Proc. of the EUSFLAT'07 conference*, Ostrava, Czech Rep., September 2007.

30. C. Marsala and B. Bouchon-Meunier. Ranking attributes to build fuzzy decision trees : a comparative study of measures. In *Proc. of the IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE*, Vancouver, Canada, August 2006.

31. T.-H. Dang, C. Marsala, B. Bouchon-Meunier, and A. Boucher. Discrimination-based criteria for the evaluation of classifiers. In *Proc. of the 7th Int. Conf. FQAS'06*, pp. 552–563, Milano, Italy, June 2006.

32. T.-H. Dang and C. Marsala. Extension of hierarchical model for fuzzy measures of discrimination. In *Proc. of the 11th IPMU'06 Conf.*, pp. 1284–1291, Paris, France, July 2006.

33. B. Bouchon-Meunier, G. Coletti, and C. Marsala. A general theory of conditional decomposable information measures. In *Proc. of the 11th IPMU'06 Conf.*, Paris, France, July 2006.

34. M. Damez, T.-H. Dang, C. Marsala, and B. Bouchon-Meunier. Fuzzy decision tree for user modeling from human-computer interactions. In *Proc. of the 5th International Conference on Human System Learning*, Marrakech (Maroc), Novembre 2005.

35. B. Bouchon-Meunier and C. Marsala. From fuzzy questionnaires to fuzzy decision trees : 30 years of research in fuzzy learning. In *Proceedings of the BISCSE'2005 workshop*, Berkeley (USA), November 2005. Conférence invitée.

36. T. Delavallade, B. Bouchon-Meunier, C. Marsala, and P. Capet. Country risk ratings : A new methodology to asses internal conflicts risk. In *Proc. of the Deloitte Risk Management Conference*, Anvers, May 2005.

37. B. Bouchon-Meunier, T.-H. Dang, and C. Marsala. Comparison of techniques for the construction of decision trees. In *Proc. of the 13th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE'04)*, pp. 58–62, Nice, France, July 2004.

38. T.-H. Dang, B. Bouchon-Meunier, and C. Marsala. Measures of information for inductive learning. In *Proc. of the 10th IPMU'04 Conf.*, pp. 1495–1502, Perugia, Italy, July 2004.

39. C. Marsala and B. Bouchon-Meunier. Selection of attributes for fuzzy decision trees. In E. Hullermeier, F. Klawon, and R. Kruse, eds., *Workshop "Soft Computing for Information Mining", in 27th conference on Artificial Intelligence*, Ulm, Germany, September 2004.

40. B. Bouchon-Meunier and C. Marsala. Measures of discrimination for the construction of fuzzy decision trees. In *Proc. of the FIP'03 conference*, pp. 709–714, Beijing, China, March 2003.

41. M. Detyniecki and C. Marsala. Discovering knowledge for better video indexing based on colors. In *Proc. of the Fuzz-IEEE'03 conference*, pp. 1177–1181, St Louis (USA), May 2003.

42. C. Marsala and M. Detyniecki. Fuzzy data mining for video. In *Proc. of the EUSFLAT'03 conference*, pp. 73–78, Zittau, (Germany), September 2003.

43. C. Marsala and B. Bouchon-Meunier. Choice of a method for the construction of fuzzy decision trees. In *Proc. of the IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE*, pp. 584–589, St Louis, USA, May 2003.

44. B. Bouchon-Meunier and C. Marsala. A comparative study of methods of construction of fuzzy decision trees. In *Proc. of the 4th International Workshop on Preferences and Decisions*, pp. 9–14, Trento (Italie), septembre 2003.

45. B. Bouchon-Meunier and C. Marsala. Improving the interpretability of fuzzy models by means of linguistic modifiers. In *Proc. of the 6th International Conference on Fuzzy Sets Theory and its Application (FSTA)*, Liptovsky Mikulas (Slovak Republic), 2002.

46. M. Detyniecki and C. Marsala. Fuzzy multimedia mining applied to video news. In *Proc. of the 9th IPMU'00 Conf.*, pp. 1001–1008, Annecy, France, July 2002.

47. B. Bouchon-Meunier and C. Marsala. Linguistic modifiers and measures of similarity or resemblance. In *Proc. of the IFSA'01 World Congress*, pp. 2195–2199, Vancouver, Canada, July 2001.

48. B. Bouchon-Meunier, D. Dubois, C. Marsala, H. Prade, and L. Ughetto. A comparative view of interpolation methods between sparse fuzzy rules. In *Proc. of the IFSA'01 World Congress*, pp. 2499–2504, Vancouver, Canada, July 2001.

49. C. Marsala and B. Bouchon-Meunier. Interpolative reasoning with multi-variable rules. In *Proc. of the IFSA'01 World Congress*, pp. 2476–2481, Vancouver, Canada, July 2001.

50. M. Detyniecki and C. Marsala. Fuzzy inductive learning for multimedia mining. In *Proc. of the EUSFLAT'01 conference*, pp. 390–393, Leicester (UK), September 2001.

51. B. Bouchon-Meunier, G. Coletti, and C. Marsala. Independence and possibilistic conditioning. In *Proc. of the Conf. on Partial Knowledge and Uncertainty : Independence, Conditioning, Inference*, Roma, Italy, May 2000. (Extended abstract).

52. B. Bouchon-Meunier, C. Marsala, and M. Rifqi. Interpolative reasoning based on graduality. In *Proc. of the 9th IEEE Int. Conf. on Fuzzy Systems*, pp. 483–487, San Antonio, Texas, May 2000.

53. B. Bouchon-Meunier, G. Coletti, and C. Marsala. Possibilistic conditional events. In *Proc. of the 8th IPMU'00 Conf.*, volume 3, pp. 1561–1566, Madrid, Spain, June 2000.

54. A. Ramer, B. Bouchon-Meunier, M. do Carmo Nicoletti, C. Marsala, and M. Rifqi. Interpolative model for fuzzy arithmetic. In *Proc. of the 9th IEEE Int. Conf. on Fuzzy Systems*, pp. 633–635, San Antonio, Texas, May 2000.

55. B. Bouchon-Meunier, E. Di Crescenzo, C. Marsala, N. Mellouli, and M. Rifqi. Uncertainty management in the recognition of a new odor. In *Proc. of the 8th IPMU'00 Conf.*, volume 3, pp. 1924–1927, Madrid, Spain, June 2000. (Poster).

56. B. Bouchon-Meunier, J. Delechamp, C. Marsala, and M. Rifqi. Analogy as a basis for various forms of approximate reasoning. In B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, eds., *Uncertainty in Intelligent and Information Systems*, pp. 70–79. World Scientific, 2000.

57. A. Laurent, B. Bouchon-Meunier, A. Doucet, S. Gançarski, and C. **Marsala**. Fuzzy data mining from multidimensional databases. In *Proc. of the Int. Symposium on Computational Intelligence - ISCI'00*, Kosice - Slovakia, August 2000.

58. C. Marsala and B. Bouchon-Meunier. Construction of fuzzy classes by fuzzy partitioning. In *Proc. of the 4^{rd} Int. Conf. FQAS'00*, Warsaw, Poland, October 2000.

59. B. Bouchon-Meunier, J. Delechamp, C. Marsala, N. Mellouli, M. Rifqi, and L. Zerrouki. Analogy and interpolation in the case of sparse rules. In *Proc. of the Eurofuse - SIC'99 Conf.*, pp. 132–136, Budapest, Hungary, Mai 1999.

60. C. Marsala and B. Bouchon-Meunier. An adaptable system to construct fuzzy decision trees. In *Proc. of the NAFIPS'99 (North American Fuzzy Information Processing Society)*, pp. 223–227, New York, USA, June 1999.

61. C. Marsala, B. Bouchon-Meunier, and A. Ramer. Hierarchical model for discrimination measures. In *Proc. of the IFSA'99 World Congress*, pp. 339–343, Taiwan, August 1999.

62. A. Ramer, B. Bouchon-Meunier, and C. Marsala. Analytical structure of hierarchical discrimination. In *Proc. of the 8th IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE*, volume 2, pp. 1050–1053, Seoul, Korea, August 1999.

63. B. Bouchon-Meunier, J. Delechamp, C. Marsala, R. Mesiar, and M. Rifqi. Fuzzy deductive reasoning and analogical scheme. In *Proc. EUSFLAT-ESTYL Joint Conference*, Palma de Mallorca (Spain), 1999.

64. C. Marsala. Application of fuzzy rule induction to data mining. In T. Andreasen, H. Christiansen, and H. L. Larsen, eds., *Proc. of the 3rd Int. Conf. FQAS'98 - LNAI 1495*, pp. 260–271, Roskilde, Denmark, May 1998. Springer-Verlag.

65. B. Bouchon-Meunier, J. Delechamp, C. Marsala, and M. Rifqi. Analogy as a basis for various forms of reasoning. In *Proc. of the 7th Int. Conf. on Intelligent Systems, ISCA'98*, Fontainebleau, France, Juillet 1998.

66. C. Marsala, M. Ramdani, M. Toullabi, and D. Zakaria. Fuzzy decision trees applied to the recognition of odors. In *Proc. of the 7th IPMU'98 Conf.*, volume 1, pp. 532–539, Paris, July 1998. Editions EDK.

67. B. Bouchon-Meunier and C. Marsala. Analogy and fuzzy deductive reasoning. In *Proc. of the NAFIPS'98 (North American Fuzzy Information Processing Society)*, Pensacola (USA), August 1998.

68. C. Marsala. Stability of fuzzy decision trees when classifying evolving observations. In D. Ruan, H. Aït Abderrahim, P. D'hondt, and E. E. Kerre, eds., *Fuzzy Logic and Intelligent Technologies for Nuclear Science and Industry*, pp. 83–90, Antwerp (Belgium), September 1998. World Scientific.

69. N. Martini Bigolin and C. Marsala. Fuzzy spatial oql for fuzzy knowledge discovery. In *Principles of Data Mining and Knowledge Discovery*, volume 1510 of *Lecture Notes in AI*, pp. 246–254. Springer Verlag, 1998. (Proc. of the 2nd European Symposium PKDD'98).

70. C. Marsala and N. Martini Bigolin. Spatial data mining with fuzzy decision trees. In N. F. F. Ebecken, ed., *Data Mining*, pp. 235–248. WIT Press, 1998. (Proc. of the Int. Conf. on Data Mining, Rio de Janeiro, Sept. 1998).

71. C. Marsala and B. Bouchon-Meunier. Construction methods of fuzzy decision trees. In *Proc. of the JCIS'98 Conf.*, volume 4, pp. 17–20, Durham, USA, October 1998.

Publications dans des conférences nationales

72. C. Marsala and B. Bouchon-Meunier. Validation de mesures de sélection d'attributs en apprentissage inductif. In *Actes de la conférence LFA 2010*, 2010.

73. S. Tollari, M. Detyniecki, A. Fakeri-Tabrizi, C. Marsala, M.-R. Amini, and P. Gallinari. Using visual concepts and fast visual diversity to improve image retrieval. In *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of the Cross-Language Evaluation Forum, Revised Selected Papers*, 2009.

74. S. Tollari, M. Detyniecki, A. Fakeri-Tabrizi, C. Marsala, M.-R. Amini, and P. Gallinari. Utilisation de concepts visuels et de la diversité visuelle pour améliorer la recherche d'images. In *Actes de Conférence en Recherche d'Informations et Applications (CORIA'09)*, 2009.

75. B. Bouchon-Meunier, C. Marsala, and M. Rifqi. Modèles flous d'adaptation en raisonnement par cas. In *Actes de la conférence LFA 2009*, 2009.

76. A. Laurent, S. Gançarski, and C. Marsala. Coopération entre un système d'extraction de connaissances floues et un système de gestion de base de données multidimensionnelles. In *Actes de la conférences LFA'00*, pp. 325–332, La Rochelle, France, Octobre 2000.

77. B. Bouchon-Meunier, J. Delechamp, C. Marsala, N. Mellouli, M. Rifqi, and L. Zerrouki. Raisonnement interpolatif à partir de schéma analogique flou. In *Rencontres JNMR, Journées Nat. sur les Modèles de Raisonnements*, Paris, France, Mars 1999. (Publication Web).

78. C. Marsala. Construction d'arbres de décision flous : le système *salammbô*. In *Actes de la conférence LFA'98*, pp. 171–176, Rennes, Novembre 1998. Cépaduès Éditions.

Autres publications

79. C. Marsala and M. Detyniecki. University of Paris 6 at TRECVID 2006 : Forest of fuzzy decision trees for high-level feature extraction. In *TREC Video Retrieval Evaluation Online Proceedings*, Novembre 2006. http ://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

80. C. Marsala and M. Detyniecki. University of Paris 6 at TRECVID 2005 : High-level feature extraction. In *TREC Video Retrieval Evaluation Online Proceedings*, Novembre 2005. http ://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

Notations

Notations pour le chapitre 1

Les notations suivantes sont utilisées :

- $\mathcal{E} = \{e_1, ..., e_n\}$, un ensemble d'exemples décrits par des attributs et associés à une valeur de classe
- e^j : valeur pour l'attribut A_j associée à l'exemple $e \in \mathcal{E}$
- e^c : valeur pour la classe C associée à l'exemple $e \in \mathcal{E}$
- $\mathcal{A} = \cup_j A_j$: ensemble de tous les attributs existants pour décrire les exemples de $\mathcal{E}$
- C : ensemble de toutes les valeurs c_k possibles associées aux exemples de $\mathcal{E}$ pour la classe. On considère que C contient K valeurs (éventuellement floues) $c_1, ..., c_K$.
- A_j : ensemble de toutes les valeurs associées aux exemples de $\mathcal{E}$ pour l'attribut A_j. On considère que A_j contient m_j valeurs (éventuellement floues) $v_1, ..., v_{m_j}$
- $\mathcal{E}_{v_{jl}}$: ensemble des exemples de $\mathcal{E}$ qui possède la valeur v_{jl} pour l'attribut A_j. On a $\mathcal{E}_{v_{jl}} = \{e \in \mathcal{E} \mid e^j = v_{jl}\}$, $\forall v_{jl} \in A_j$
- $\mathcal{E}_{c_k}$: ensemble des exemples de $\mathcal{E}$ qui possède la classe c_k. On a $\mathcal{E}_{c_k} = \{e \in \mathcal{E} \mid e^c = c_k\}$, $\forall c_k \in C$.
- étant donné un ensemble d'objets $\mathcal{S}$, éventuellement flou, on note $\mathbb{P}[\mathcal{S}]$ l'ensemble des sous-ensembles (éventuellement flous) de $\mathcal{S}$, soit $\mathbb{P}[\mathcal{S}] = \{U \mid U \subseteq \mathcal{S}\}$.

Comme nous travaillons sur des sous-ensembles flous, nous allons préciser dans un premier temps ici ce que nous allons choisir pour définir nos opérations de base. Soit un A et B, deux sous-ensembles flous définis sur un univers X :

- l'inclusion de sous-ensembles flous : on dit que A est inclus dans B ($A \subseteq B$) si $\forall x \in X,\ \mu_A(x) \leq \mu_B(x)$.
- l'intersection $A \cap B$ de deux sous-ensembles flous A et B de X est définie à partir d'une t-norme $\top$: $\mu_{A \cap B}(x) = \top(\mu_A(x), \mu_B(x))$.
- l'union $A \cup B$ de deux sous-ensembles flous A et B de X est définie à partir d'une t-conorme $\perp$: $\mu_{A \cup B}(x) = \perp(\mu_A(x), \mu_B(x))$.
- la cardinalité $|A|$ d'un sous-ensemble flou A est donné en sommant les degrés d'appartenance des éléments de A. Pour A fini (notre cas en apprentissage inductif) : $|A| = \sum_{x \in X} \mu_A(x)$.
- une partition U de X est un ensemble de $n > 1$ sous-ensembles flous $u_1, ..., u_n$ de X telle que pour tout $x \in X$, $\sum_{i=1}^{n} u_i(x) = 1$.
- à partir d'un sous-ensemble flou A, pour tout $\alpha \in [0, 1]$, on définit ${}^{\alpha}A$, l'alpha-coupe de A de niveau α par : ${}^{\alpha}A = \{x \in X \mid \mu_A(x) \geq \alpha\}$.

Rappels

Probabilités d'événements flous

Probabilités d'événements flous

Quand on se place en théorie des sous-ensembles flous, il faut alors considérer des probabilités d'événements flous. La mesure de probabilité d'événements flous généralement utilisée est celle introduite et étudiée par Zadeh [113] :

Définition 11 *Un sous-ensemble flou U de $\mathcal{X}$, de fonction d'appartenance μ peut être assimilé à un événement flou et la* probabilité de l'événement flou U *est définie par :*

$$p^*(U) = \sum_{i=1}^{n} \mu(x_i)p(x_i) \tag{1}$$

On vérifie aisément que l'on retrouve une probabilité classique si U n'est pas flou. Comme Zadeh l'a montré, la mesure p^* est bien une mesure de probabilité [113], elle vérifie :

1. $p^*(\mathcal{X}) = 1$,
2. pour tout événement flou U : $0 \leq p^*(U) \leq 1$,
3. si U_1 et U_2 sont deux événements flous disjoints, on a[1] $p^*(U_1 \cup U_2) = p^*(U_1) + p^*(U_2)$.

Probabilité conditionnelle d'événements flous

Par extension de la probabilité conditionnelle classique, on peut donner la définition suivante :

Définition 12 *Soit U_1 et U_2, deux événements flous de $\mathcal{X}$. On suppose $p^*(U_2) \neq 0$. La* probabilité conditionnelle de U_1 sachant U_2 *est définie par :*

$$p^*(U_1|U_2) = \frac{p^*(U_1 \cap U_2)}{p^*(U_2)} \tag{2}$$

Dans son article [113], Zadeh introduit cette définition de la probabilité conditionnelle mais en choisissant la t-norme produit comme opérateur d'intersection afin de respecter la notion d'indépendance qu'il a introduit et qui est basée sur le produit d'ensembles flous et non sur leur intersection.

Nous allons par la suite conserver cette définition plus générale de probabilité conditionnelle floue en ne déterminant la t-norme que lorsque ce sera vraiment nécessaire.

On vérifie que la mesure de probabilité d'événements flous $p^*_{U_2}$ définie pour tout événement flou U de $\mathcal{X}$, par $p^*_{U_2}(U) = p^*(U|U_2)$ est bien une mesure de probabilité :

[1] Zadeh utilise la t-norme min pour l'intersection et la t-conorme max pour l'union de sous-ensembles flous.

1. $p^*_{U_2}(\mathcal{X}) = \dfrac{p^*(\mathcal{X} \cap U_2)}{p^*(U_2)} = \dfrac{p^*(U_2)}{p^*(U_2)} = 1$,
2. pour tout événement flou $U : 0 \leq p^*_{U_2}(U) \leq 1$ car $U \cap U_2 \subseteq U_2$ et donc $p^*(U_1 \cap U_2) \leq p^*(U_2)$,
3. si U et V sont deux événements flous disjoints, on a $p^*((U \cup V) \cap U_2) = p^*((U \cap U_2) \cup (V \cap U_2))$. Et étant donné que U et V sont disjoints, $U \cap U_2$ et $V \cap U_2$ sont aussi disjoints et donc $p^*((U \cap U_2) \cup (V \cap U_2)) = p^*((U \cap U_2)) + p^*(V \cap U_2))$, ce qui donne donc $p^*_{U_2}(U \cup V) = p^*_{U_2}(U) + p^*_{U_2}(V)$.

Le seul inconvénient de définir ainsi la probabilité conditionnelle pour n'importe quelle t-norme, est que la notion d'indépendance d'événements qui en découle est alors plus complexe à définir. L'avantage de l'utilisation de la t-norme produit permet de conserver la même définition d'indépendance que dans le cas classique : U_1 et U_2 sont indépendants si $p^*(U_1|U_2) = p^*(U_1)$ (avec, de façon implicite, l'égalité $p^*(U_1 \cap U_2) = p^*(U_1)p^*(U_2)$ qui doit être vérifiée). On verra par la suite (travaux avec Bernadette et Giuliannella décrits en fin de ce chapitre) que la recherche d'une meilleure définition pour cette indépendance amène l'introduction d'une nouvelle définition de ce que l'on entend par "événement conditionnel".