



HAL
open science

Une méthode d'analyse discriminante pour des concepts imprécis

Dominique Loutchmia

► **To cite this version:**

Dominique Loutchmia. Une méthode d'analyse discriminante pour des concepts imprécis. Informatique [cs]. Université de la Réunion, 1998. Français. NNT : . tel-00473292

HAL Id: tel-00473292

<https://theses.hal.science/tel-00473292>

Submitted on 15 Apr 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TH
161

Une Méthode d'Analyse Discriminante
pour des Concepts Imprécis

Thèse de doctorat de l'Université de La Réunion

Spécialité: Informatique

Dominique Loutchmia

25 Septembre 1998

— UNIVERSITÉ DE LA RÉUNION —

Institut de Recherche en Mathématiques et Informatique Appliquées

École Doctorale de Mathématiques et Informatique de Marseille

THÈSE

Présentée pour obtenir le titre de

Docteur en Sciences

Spécialité : Informatique

par

DOMINIQUE LOUTCHMIA

Une Méthode d'Analyse Discriminante pour des Concepts Imprécis

Soutenue le 25 Septembre devant le jury composé de :

Me. BERNADETTE BOUCHON-MEUNIER	Directeur de Recherche	CNRS	Rapporteur
M. K.CHIDANANDA GOWDA	Professeur	S.J.College of Engineering	Rapporteur
M. ALAIN GUÉNOCHE	Chargé de Recherche	CNRS, Marseille	Examineur
M. HENRI RALAMBONDRAINY	Professeur	Univ. de La Réunion	Dir. de thèse

Remerciements. . .

Mes tous premiers remerciements vont naturellement à Henri Ralambondrainy, mon directeur de thèse, pour m'avoir encadré et suivi pendant ces quatre années.

Je remercie Me Bernadette Bouchon-Meunier, Mr Chidananda Gowda, Mr Alain Guénoche pour l'intérêt qu'ils ont manifesté à mon travail en acceptant d'être membres de mon Jury.

Je remercie tous les membres de l'IREMIA pour leur sympathie et en particulier les deux "Pierre" piliers du labo.

Enfin, je tiens à remercier mes amis, ma famille, mes parents et Réjane pour leurs encouragements et leur soutien.

Table des matières

Introduction	3
1 Les approches numériques et symboliques	5
1.1 Méthode de segmentation non paramétrique	6
1.2 Induce 1-2	7
1.3 Discrimination par des fonctions complètes et consistantes	9
1.4 Une procédure de caractérisation à partir de données mixtes	10
2 Les connaissances imprécises et incertaines	13
2.1 L'imprécision	13
2.1.1 Définition d'un sous-ensemble flou	14
2.1.2 Normes et conormes triangulaires	15
2.1.3 Les α coupes	17
2.2 L'incertitude	18
2.2.1 La théorie des probabilités	18
2.2.2 La théorie des possibilités	19
2.2.3 La théorie de l'évidence	20
3 Les méthodes de discrimination floue	23
3.1 Arbre de décision flou	23
3.1.1 Le système FID3	23
3.1.2 Le système SAFI	26
3.2 Processus d'induction avec quantificateurs linguistiques	28
4 La représentation des données	31
4.1 Les nuances	32
4.2 Rappels sur les treillis	34
4.3 La représentation des propriétés	35
4.3.1 Les attributs de type simple	36
4.3.2 Les attributs de type symbolique	36
4.3.3 Les attributs de type taxonomique	38

4.4	Les objets structurés	42
4.4.1	Les attributs de type structuré	42
4.4.2	Structure sur l'ensemble des valeurs	43
4.5	Représentation structurelle des objets	45
5	Distances et mesures d'agrégation	49
5.1	Modularité et Distances	50
5.1.1	Les treillis modulaires	50
5.1.2	La distance de <i>Hamming</i>	51
5.1.3	La distance <i>Bi-partite</i>	52
5.1.4	La distance <i>Min-Max</i>	53
5.2	Différentes métriques pour les attributs de type simple	54
5.3	Fonctions d'agrégation sur des données structurées	58
5.3.1	Fonction d'agrégation	59
5.3.2	Distance structurelle	60
6	La représentation des concepts	65
6.1	La caractérisation des observations	65
6.1.1	" <i>L'appartenance</i> " à une description	65
6.1.2	Fonctions caractéristiques floues	67
6.2	Généralités sur les concepts imprécis	70
7	Qualité de la discrimination	73
7.1	Introduction	73
7.2	Définition de la complétude et de la consistance	74
7.2.1	Complétude et consistance	76
7.2.2	Les propositions floues quantifiées	77
7.3	Evaluation des propositions floues de types " <i>les A sont B</i> "	78
7.3.1	Implications floues et degré d'inclusion	79
7.3.2	Utilisation de mesures de similarité	80
7.3.3	Utilisation des α -coupes	82
7.4	Application sur un exemple "jouet"	84
7.5	Conclusion	87
8	Méthode de discrimination	89
8.1	L'espace de recherche : Le treillis $\mathcal{T}_\alpha[A]$	90
8.1.1	Définitions et propriétés	90
8.1.2	Le treillis des descriptions discriminantes	92
8.1.3	Méthode de construction du treillis $\mathcal{T}_\alpha[A]$	95
8.1.4	Exemple	98

8.2	Caractérisation de la classe des exemples	100
8.2.1	Critère de qualité d'un concept	100
8.2.2	Méthode de parcours du treillis \mathcal{T}_α	101
8.2.3	Exemple	103
8.3	Complexité des algorithmes	104
8.4	Optimisation de la discrimination	106
8.5	Méthode de recherche des attributs discriminants	108
8.5.1	Pouvoir discriminant d'un ensemble d'attributs	109
8.5.2	Algorithme de sélection des attributs discriminants	111
9	Le système DFI	113
9.1	Présentation du système	113
9.1.1	Le traitement des données et des connaissances	114
9.1.2	Sélection des attributs les plus discriminants	115
9.1.3	La discrimination	115
9.2	Les paramètres d'entrée	116
10	Applications	119
10.1	Application sur les <i>Hyalonema</i>	119
10.1.1	La représentation des éponges	119
10.1.2	Sélection des attributs discriminants	121
10.1.3	Caractérisation de la classe des <i>Cyliconema</i>	122
10.2	Application sur les <i>Votes</i>	126
10.2.1	Les votes	126
10.2.2	Du numérique au symbolique	126
10.2.3	Caractérisation des classes	129
	Conclusion	133
A	Les opérateurs de généralisation et de spécialisation pour les taxonomies	137
A.1	La généralisation de taxonomies	137
A.1.1	Preuve de la proposition 4.3.1	137
A.1.2	Procédure de généralisation	139
A.2	La spécialisation de taxonomies	140
	Bibliographie	145

Table des figures

1.1	<i>Illustration de la caractérisation</i>	12
3.1	<i>Forme de l'arbre de décision flou possibiliste</i>	25
3.2	<i>Exemple de partition floue</i>	26
3.3	<i>Effet des similarités</i>	27
4.1	<i>Des nuances de croyance</i>	33
4.2	<i>Des nuances d'intensité</i>	34
4.3	<i>Représentation symbolique de l'attribut Taille</i>	36
4.4	<i>Description taxonomique de régions</i>	38
4.5	<i>Instances hiérarchiques mal définies</i>	39
4.6	<i>Définition de l'opérateur ∇</i>	40
4.7	<i>Instances hiérarchiques bien définies</i>	41
4.8	<i>Exemples de généralisations</i>	42
4.9	<i>Instances de l'attribut Corps</i>	44
4.10	<i>Exemples de descriptions structurelles</i>	46
5.1	<i>Exemples de treillis modulaires</i>	51
5.2	<i>Treillis non modulaires</i>	52
5.3	<i>Les différents cas</i>	58
5.4	<i>Répartition des poids en fonction de la hauteur</i>	60
6.1	<i>Représentativité d'une description en fonction de la relation d'ordre</i>	69
6.2	<i>Intention et extension d'une classe imprécise</i>	71
7.1	<i>La classe dans son contexte</i>	75
7.2	<i>Recouvrement de la classe H</i>	76
7.3	<i>Une représentation du quantificateur "la plupart"</i>	77
7.4	<i>Exemples de modificateurs linguistiques</i>	78
7.5	<i>Inclusion de A dans B</i>	81
7.6	<i>Exemples de coupes</i>	82
7.7	<i>Exemples de recouvrements par coupes</i>	86

8.1	<i>Insertion d'un nouvel exemple dans $T_\alpha[A]$</i>	95
8.2	<i>Insertion de l'exemple $\{a, b\}$</i>	98
8.3	<i>Insertion de l'exemple $\{d\}$</i>	99
8.4	<i>Insertion de l'exemple $\{b, c\}$</i>	99
8.5	<i>Recherche de concepts</i>	101
8.6	<i>Initialisation de la caractérisation</i>	103
8.7	<i>Choix de la 1^{ère} description de concept</i>	103
8.8	<i>Liste Pge à renouveler</i>	104
8.9	<i>Valeur maximale atteinte, fin</i>	104
8.10	<i>Initialisation</i>	107
8.11	<i>Insertion des premiers éléments</i>	107
8.12	<i>Exemple de réduction du treillis</i>	108
8.13	<i>Le genre Hyalonema</i>	110
9.1	<i>Le système DFI</i>	114
9.2	<i>Exemples de modificateurs</i>	116
10.1	<i>Le modèle de l'expérimentation</i>	120
10.2	<i>Les différentes classes de Hyalonema</i>	121
10.3	<i>Modèle à 19 attributs</i>	122
10.4	<i>Modèle à 6 attributs</i>	123
10.5	<i>Intention du concept \mathcal{H}_1</i>	123
10.6	<i>Intention du concept \mathcal{H}_2</i>	124
10.7	<i>Intention du concept \mathcal{H}_3</i>	124
10.8	<i>Intention du concept \mathcal{H}_4</i>	125
10.9	<i>Extension des concepts \mathcal{H}_3 et \mathcal{H}_4</i>	125
10.10	<i>Etape de fuzzification des valeurs</i>	127
10.11	<i>Du flou au symbolique</i>	128
10.12	<i>Le modèle de représentation des Votes</i>	129
10.13	<i>Intention du concept \mathcal{H}'_4</i>	130
10.14	<i>Intention du concept \mathcal{H}'_3</i>	130
10.15	<i>Intention du concept \mathcal{H}'_2</i>	131
10.16	<i>Intention du concept \mathcal{H}'_1</i>	132
A.1	<i>Exemple de spécialisation appliquée aux sous-hiérarchies</i>	142
A.2	<i>Exemple de spécialisation sur les régions</i>	143

Liste des symboles

\perp, \top	borne inférieure et supérieure d'un treillis
\wedge	opérateur de spécialisation
\vee	opérateur de généralisation
$h()$	fonction hauteur
$[A]$	treillis des descriptions possibles de l'attribut A
$\mathbf{[A]}$	treillis des descriptions structurelles de l'attribut A
$\rho(A_i), w_i$	pois de l'attribut simple A_i
D_H	distance de Hamming généralisée relative
d_m	distance Min-Max
d_b	distance bi-partite
D	distance moyenne
D_S	distance pondérée par la structure
∇	distance structurelle
S	similarité structurelle
χ_d	fonction de reconnaissance associée à la description d
$\mathcal{E}, C_{\mathcal{E}}$	ensemble des exemples et des contre-exemples
Q	quantificateur linguistique
m	modificateur linguistique
$V(A \subset B)$	degré d'inclusion de A dans B
$\mathcal{T}_{\alpha}[A]$	treillis des descriptions α -discriminantes générées par A
$G(\mathcal{H})$	fonction de qualité du concept $calH$
$I_s(e, f)$	indice de séparabilité entre e et f
$I_c(e)$	indice de cohérence de la description e
$Pw(\{A_1, \dots, A_n\})$	pouvoir de discrimination de l'ensemble des attributs $\{A_1, \dots, A_n\}$

L'accession à la conscience présuppose le franchissement des portes de l'imagination, dont les clés sont de nature symbolique. On peut faire passer ses idées avec soi... mais uniquement sous forme de symboles.

Raja Flatterie
Psychiatre-aumônier
Frank Herbert in *L'incident Jésus*

Introduction

L'acquisition de connaissances à partir d'observations est une préoccupation commune à de nombreux domaines tels que les Sciences de la Nature. Différentes approches ont été développées selon la manière de représenter les données ou encore selon les objectifs à réaliser. Dans bien des cas, les observations sont décrites par un ensemble de propriétés caractérisées par des attributs, appelés également variables.

En Analyse des Données, les variables descriptives sont généralement numériques. Lorsque les données comportent des variables symboliques, elles sont codées sous forme numérique avant traitement. Les techniques d'Analyse de Données s'appuient sur les espaces métriques où les distances jouent un rôle prépondérant pour classer les observations. Cependant, les résultats obtenus ne sont pas toujours faciles à interpréter pour l'utilisateur.

L'Approche Symbolique s'intéresse à des techniques permettant d'extraire de la connaissance à partir d'observations caractérisées par des attributs symboliques. Les formalismes utilisés pour la représentation des données sont des extensions de la logique des propositions, du calcul des prédicats, ... Les mécanismes mis en œuvre pour acquérir de la connaissance sont fondés sur des méthodes de raisonnement telles que l'induction, la déduction, l'analogie, ... Contrairement à l'Analyse des Données, les résultats obtenus sont plus faciles à comprendre à cause de leurs représentations symboliques.

Cependant, ces approches présentent certaines limites, elles ne traitent que de données précises et ne déterminent que des concepts précis ou de nature probabiliste. Les méthodes d'acquisition de connaissance s'appuyant sur la théorie des sous-ensembles flous, sont capables de prendre en compte l'imprécision aux niveaux des données et des concepts.

Les travaux de cette thèse se situent à l'intersection de ces trois approches. Le problème auquel nous nous intéressons est la discrimination de concepts ou de classes. Notre objectif est de pouvoir caractériser des classes d'observations imprécises et struct-

urées, par un ensemble de règles lisibles car symboliques. La méthode de discrimination conceptuelle proposée est de nature inductive et s'appuie sur des opérateurs de généralisation définies dans un espace métrique. Ces méthodes de discrimination ont été ensuite implantées dans le système DFI et nous présentons des applications montrant la pertinence de l'approche proposée.

Le plan de la thèse se décompose de la manière suivante :

- Dans les trois premiers chapitres, nous présentons quelques méthodes de discrimination utilisées en Analyse des Données et en Apprentissage Symbolique pour des données précises et des extensions pour des données vagues. Pour cela, nous faisons un rappel sur les différentes théories utilisées permettant de modéliser les connaissances incertaines et imprécises.
- Après cet état de l'art, les chapitres quatre et cinq sont consacrés d'abord au modèle de représentation des données pour des objets imprécis et structurés, puis aux métriques et mesures de similarité prenant en compte les différents aspects du modèle.
- Dans le chapitre six, nous introduisons les fonctions de reconnaissance floue pour ensuite définir les notions de concepts imprécis.
- Dans le chapitre sept, nous nous intéressons à évaluer à l'aide de critères la complétude et la consistance d'une fonction caractéristique associée à un ensemble de descriptions. Ces critères prennent en compte la présence, éventuelle, d'erreurs de classement d'observations.
- Dans le chapitre huit, nous proposons une méthode de discrimination, ainsi qu'une méthode de sélection des attributs les plus discriminants pour une base de données.
- Après la présentation du système DFI dans le chapitre neuf, nous présentons une expérimentation du système sur une base de cas d'éponges appartenant au genre *Hyalonema*, ainsi qu'une expérimentation sur des données d'enquête.

Chapitre 1

Les approches numériques et symboliques

Sommaire

1.1	Méthode de segmentation non paramétrique	6
1.2	Induce 1-2	7
1.3	Discrimination par des fonctions complètes et consistantes	9
1.4	Une procédure de caractérisation à partir de données mixtes	10

Les problèmes d'acquisition de concepts regroupent principalement la classification appelée "*apprentissage par la découverte*" et la discrimination appelée également "*apprentissage supervisé*". Les problèmes de classification sont concernés par la détermination des catégories ou classes d'individus à partir d'un ensemble d'observations, tandis que la discrimination consiste à caractériser une classe d'individus par un ensemble de règles.

Les méthodes d'Analyse des Données traitent de variables quantitatives en se plaçant dans l'espace vectoriel \mathbb{R}^k . Les méthodes de classification de données telles que les nuées dynamiques, les hiérarchies, les pyramides ou encore les méthodes d'analyse discriminante telles que les segmentations, l'analyse factorielle [9],[2] sont des méthodes de traitement efficaces pour des bases de données volumineuses mais donnent des résultats qui sont souvent difficile à interpréter.

Par opposition, l'approche symbolique a l'avantage de simplifier l'interprétation des descriptions et permet en plus une représentation plus complexe des données. Certaines méthodes ont été développées pour traiter des données décrites à la fois par des variables qualitatives et quantitatives [17], mais dans beaucoup de cas les variables quantitatives

sont transformées par des algorithmes de codage en variables qualitatives [34]. La méthode de l'étoile introduite par Michalsky [40], les méthodes de Guénoche [18], la méthode Lerman et Nicolas [25] sont des exemples de classification conceptuelle. Par ailleurs, L'espace des versions de Mitchell [31], Induce de Michalsky basée sur l'étoile, les arbres de décision telles que ID_3 de Quinlan [19] sont à la base de différentes méthodes de classification.

Comme nous nous intéressons dans cette thèse à la discrimination, nous présentons dans ce chapitre quelques méthodes de discrimination numérique et symbolique.

1.1 Méthode de segmentation non paramétrique

La méthode de J. Friedman décrite dans [9], détermine un arbre de décision binaire sur des individus séparés en deux classes. Les méthodes de construction d'arbres de décisions ont l'avantage de fournir rapidement des résultats faciles à interpréter, mais leur qualité de discrimination ne s'avère pas toujours satisfaisante.

Notations et terminologie

(x_1, \dots, x_n) sont n individus décrits sur p variables quantitatives.

P_1 et P_2 partitionnent les n individus en deux classes possédant respectivement n_1 et n_2 individus.

π_1 et π_2 sont les probabilités pour un élément d'appartenir à P_1 ou P_2 et vérifient $\pi_1 + \pi_2 = 1$.

l_1 (resp. l_2) représente le coût de mauvaise classification d'un élément de P_1 (resp. P_2) dans P_2 (resp. P_1).

F_1 et F_2 sont des fonctions de répartitions des groupes P_1 et P_2 .

Les risques

Dans le cas où les coûts sont proportionnels aux probabilités d'apparition des groupes (ie. $\pi_1 l_2 = \pi_2 l_1$), les fonctions suivantes évaluent le risque de mal classer un individu pour le point de coupure z par rapport à l'échantillon (x_1, \dots, x_n) :

- Risque de Bayes : $R(z) = (\pi_1 l_1 (1 - F_1(z)) + \pi_2 l_2 F_2(z))$

- Risque estimé : $\hat{R}(z) = (\pi_1 l_1 (1 - \hat{F}_1(z)) + \pi_2 l_2 \hat{F}_2(z))$ où,

$$\hat{F}_1(x) = \begin{cases} 0 & \text{si } x < x_1^1 \\ \frac{k}{n_1} & \text{si } x_k^1 \leq x < x_{k+1}^1 \\ 1 & \text{si } x_{n_1}^1 \leq x \end{cases}$$

et

$$\hat{F}_2(x) = \begin{cases} 0 & \text{si } x < x_1^2 \\ \frac{k}{n_2} & \text{si } x_k^2 \leq x < x_{k+1}^2 \\ 1 & \text{si } x_{n_2}^2 \leq x \end{cases}$$

où x_k est le $k^{\text{ième}}$ élément de la classe i rangée par ordre croissant. Les répartition F_1 et F_2 ne sont pas souvent connues et sont estimées par les fonctions \hat{F}_1 et \hat{F}_2 .

Principe et Méthode

Dans l'hypothèse que $\pi_1 l_2 = \pi_2 l_1$, il a été montré que le risque estimé $\hat{R}(\hat{c})$ converge en probabilité vers le risque de Bayes $R(c)$. \hat{c} désigne le point de coupure minimisant le risque estimé et c celui qui minimise le risque de Bayes. Ainsi dans le cas d'une variable, le point c permet de séparer les individus en minimisant les mauvais classements. A noter que sous l'hypothèse $\pi_1 l_2 = \pi_2 l_1$, le point minimisant le risque de Bayes correspond à la distance de Kolmogorov-Smirnov pour les deux classes.

Dans le cas de plusieurs variables, l'arbre de décision est construit de la manière suivante :

- Rechercher la variable qui minimise le risque estimé $\hat{R}(\hat{c})$.
- Séparer le groupe d'individus en deux sous-groupes en fonction de la coupure \hat{c} . Chaque sous-groupe, et par conséquent, chaque branche est caractérisée par le segment qui le porte.
- Répéter le processus sur chaque sous groupe tant que le test d'arrêt n'est pas vérifié. Ce test vérifie que le partitionnement précédent assure un échantillon de taille minimum pour chaque classe. La taille minimale est fixée par l'utilisateur.

Dans le cas de plusieurs classes, il est possible de réaliser un arbre de décision en se ramenant à chaque étape en deux groupes de classes. Il s'agira alors de définir une fonction de répartition empirique pour un groupe de variables, une fonction permettant de déterminer la meilleure partition des variables et un point de coupure séparant les individus pour chaque variable.

1.2 Induce 1-2

Le programme Induce 1-2 proposé par Dietterich et Michalsky [12] s'appuie sur une méthode d'apprentissage à partir d'exemples dont l'objectif est de trouver une description de concept à partir des exemples positifs du concept. Cette méthode s'applique à des descriptions structurées. Le langage de représentation des descriptions est une extension de la logique des prédicats du premier ordre.

Notation et Terminologie

Les événements sont décrits comme une conjonction de sélecteurs représentés par des prédicats unaires ou n -aires.

Les prédicats unaires concernent les descripteurs d'attribut, tandis que les prédicats n -aires concernent les descripteurs de structure.

Principe

L'espace des descriptions est séparé en deux classes, la classe des événements élémentaires regroupant les descripteurs d'attribut et l'espace des événements de structure regroupant les descripteurs de structure.

L'approche utilisée consiste d'abord à explorer l'espace des structures pour déterminer des généralisations plausibles, puis à explorer l'espace des événements élémentaires pour compléter la généralisation.

Les critères utilisés pour mesurer la qualité d'une généralisation peuvent être :

- la maximisation du nombre d'événements d'entrée couverts par la généralisation,
- la maximisation du nombre de sélecteurs dans une généralisation,
- etc ...

Méthode

L'exploration de l'espace des structures

- B_0 est composé initialement d'un échantillon d'événements de structure issus des événements d'entrée.
- A chaque étape, B_{i+1} est formé des généralisations possibles des éléments de B_i dont on a supprimé les sélecteurs isolés.
- B_{i+1} est réduit à un nombre fixe de généralisations "*moins prometteuses*" en fonction de critères mesurant la qualité de la généralisation.
- Les éléments de B_{i+1} recouvrant les exemples, lui sont retirés et ajoutés à l'ensemble des généralisations C .
- Tant que C n'atteint pas une taille fixée au préalable, l'étape suivante est effectuée.

Chaque description g de C va permettre de définir un espace d'attribut qui sera parcouru de la même manière que l'espace des structures. L'espace des descriptions engendré par g est composé de tous les objets décrits sur les variables qui apparaissent dans g . La méthode précédente est appliquée et les meilleurs généralisations sont conservées dans l'ensemble C' .

La généralisation conjonctive G est obtenue en combinant les éléments de C' avec les descripteurs de la structure g .

Pour des problèmes de discrimination, les exemples négatifs sont pris en compte au niveau des critères de qualité des généralisations. Par exemple, les généralisations qui recouvrent des contre-exemples seront écartées.

1.3 Discrimination par des fonctions complètes et consistantes

Dans le cadre de la discrimination conceptuelle, Guénoche [17] propose une méthode de construction de fonctions discriminantes caractérisant un ensemble d'exemples par rapport à un ensemble de contre-exemples.

Ces fonctions affectent une valeur "*vraie*" à tous les exemples et une valeur "*fausse*" à tous les contre-exemples. Elles désignent habituellement des fonctions complètes et consistantes. Dans ce qui suit, nous limitons la présentation aux algorithmes fonctionnant pour des objets décrits sur des attributs binaires.

Notations et terminologie

Soit X un ensemble de n objets décrits par un ensemble $A = \{A, B, C, \dots\}$ de m variables.

A une variable A correspond deux modalités a (dite forme directe) et a' (dite forme conjuguée). L'ensemble a^* des valeurs prises pour les m variables est de cardinal 2^m . Un *monôme* de longueur k est une conjonction de k éléments de a^* , où chaque variable est décrite une seule fois.

Les monômes *générateurs* sont les monômes de a^* de longueur 1. Tous les objets de X sont identifiés par un monôme.

Soit $Y \subset X$ et un monôme μ .

- $Y(\mu)$ est l'ensemble des monômes de Y contenant μ .
- μ est dit *Y-vide* si $|Y(\mu)| = 0$.
- μ *couvre* $Y(\mu)$ lorsque μ n'est pas *Y-vide*.

L'ensemble X est partitionné en deux classes disjointes, la classe des exemples notée C et la classe des contre-exemples notée Q .

Principes et méthodes

La proposition ci-dessous affirme que toute fonction caractéristique construite comme

une disjonction de conjonctions d'attributs, est un recouvrement de C par des monômes Q vides.

Proposition 1 Soit F une disjonction de monômes : $F = \bigvee_{j=1..r} \mu_j$.

La fonction F est caractéristique de C relativement à Q si et seulement si les monômes μ_j vérifient :

$$|Q(\mu_j)| = 0 \text{ et } \bigcup_{j=1..r} C(\mu_j) = C$$

Pour obtenir une fonction caractéristique de C , l'algorithme présenté par Guénoche produit, dans un premier temps, l'ensemble de tous les monômes vides de Q à l'aide de l'algorithme de Kuntzmann, pour ensuite en extraire une couverture minimale de la classe C . Chaque itération de la procédure de couverture consiste à conserver le monôme recouvrant le plus d'exemples non déjà recouverts, jusqu'à obtenir un recouvrement de C .

Proposition 2 Soit μ un monôme non vide de Q et G un générateur qui prend sur $Q(\mu)$ une valeur constante g , alors $g'\mu$ est un monôme vide de Q .

Lorsque le nombre de générateurs est élevé, l'algorithme de Kuntzmann est rendu inefficace. Guénoche propose alors un algorithme qui couple l'étape de recherche des monômes Q -vides (qui seront au plus de longueur 3) et l'étape de couverture de la classe C . L'algorithme consiste à produire les monômes de longueur 1 pour construire une couverture de C discriminant les contre-exemples. Ensuite, il s'agit de couvrir les éléments restants à partir de monômes de longueur 2. Les monômes de longueur 3 sont obtenus à partir de ceux de longueur 2 et de la proposition 2. Les éléments de C non recouverts complètent la fonction caractéristique.

Guénoche signale que les algorithmes fournissent une couverture satisfaisante bien qu'ils ne garantissent pas de couverture minimale. Lors des expérimentations, l'heuristique proposée pour les grands tableaux de données se montre 20 fois plus rapide. Afin de déterminer des fonctions caractéristiques plus robustes, Guénoche suggère de prendre en compte des monômes *presque* Q -vides, ou encore de chercher à construire un recouvrement partiel, etc ...

1.4 Une procédure de caractérisation à partir de données mixtes

Ralambondrainy dans [34] propose d'une part, une version étendue de l'algorithme des K -centres mobiles à des données décrites à la fois sur des attributs numériques et symboliques, et d'autre part une procédure de caractérisation sur les classes obtenues.

L'algorithme des centres mobiles sera étendu aux données mixtes à l'aide d'une distance entre objets prenant en compte les deux groupes d'attributs. L'influence des groupes intervient dans la distance comme des poids calculés à partir d'un critère d'inertie sur les observations. L'algorithme fournit en sortie une partition de l'ensemble des observations en classes. Nous nous intéressons ici, à la caractérisation des classes obtenues à l'étape précédente.

Représentation de la connaissance

Un attribut symbolique ou numérique¹ est représenté par (X, D, L) où :

- X identifie l'attribut,
- D est le domaine de valeur de l'attribut,
- L est l'ensemble des nuances possibles.

L'espace de recherche associé à X est un sous-ensemble de l'ensemble des parties de D . Une observation est un n -uplet de valeurs décrites sur les attributs $(X_1, D_1, L_1), \dots, (X_n, D_n, L_n)$. $\Omega = E_1 \times \dots \times E_n$ appelé espace des objets, possède une structure de treillis.

Un prédicat A_w associé à une description w est une application de Ω dans l'ensemble $\{true, false\}$ associant *vraie* à toute valeur inférieure à w dans le treillis Ω .

Les observations les plus typiques de la classe à décrire constituent l'ensemble des exemples noté C . Les autres observations représentent les contre-exemples.

Conditions de sélection des prédicats

Afin d'optimiser la procédure de caractérisation, les prédicats doivent satisfaire des critères de qualité de recouvrement des exemples et de discrimination des contre-exemples.

- (C1) Un prédicat A doit être α -discriminant. Le nombre de contre-exemples reconnus doit être inférieur à α .
- (C2) Le nombre d'élément de la classe C reconnu par un prédicat, doit être supérieur à β .

Principes et méthodes

La procédure de caractérisation est une méthode ascendante basée sur la généralisation

1. Les attributs numériques (X, \mathbb{R}) sont transformés en attribut symbolique (X, D) , où $D = \{inf, typical, sup\}$, par un processus de codage. Chaque valeur numérique de l'attribut X correspond à un élément de D en fonction de sa typicité pour X .

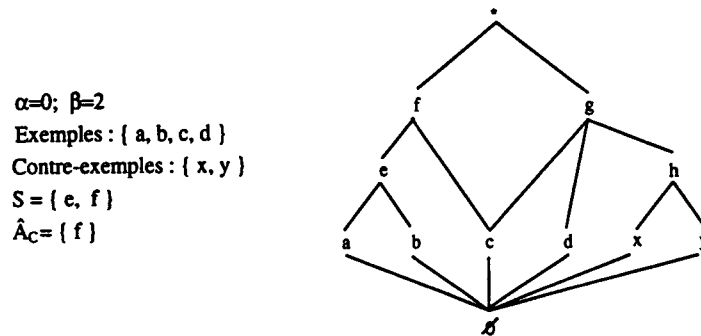


FIG. 1.1 – Illustration de la caractérisation

de prédicats et se divise en deux étapes.

- La première étape consiste à générer l'ensemble $D_{\alpha-disc}$ des descriptions vérifiant la condition (C1).

$$D_{\alpha-disc} = \bigcup_{i \in \mathbb{N}} \otimes_i EX$$

où $\otimes_{i+1} EX = \{x \vee y \mid x, y \in \otimes_i EX, x \neq y \text{ et } x \vee y \text{ } \alpha\text{-discriminante}\}$ et \vee est l'opérateur de généralisation correspondant à la borne supérieure des deux éléments dans le treillis des prédicats.

Ensuite, seuls les prédicats de $D_{\alpha-disc}$ satisfaisant la condition (C2) sont conservés et forment l'ensemble S .

- Dans la deuxième étape, une caractérisation de la classe C est établie à partir des prédicats de S . De manière linéaire, la caractérisation \hat{A}_C est construite par ajouts successifs de prédicats de S non contenus dans \hat{A}_C .

\hat{A}_C représente une description de concept de la classe C .

Comme nous l'avons précisé en introduction, ces méthodes ne traitent pas de données imprécises. Nous allons présenter dans le chapitre suivant certaines théories de l'incertain, afin de décrire des méthodes de discrimination floue.

Chapitre 2

Les connaissances imprécises et incertaines

Sommaire

2.1	L'imprécision	13
2.1.1	Définition d'un sous-ensemble flou	14
2.1.2	Normes et conormes triangulaires	15
2.1.3	Les α coupes	17
2.2	L'incertitude	18
2.2.1	La théorie des probabilités	18
2.2.2	La théorie des possibilités	19
2.2.3	La théorie de l'évidence	20

Dans ce chapitre, nous allons présenter différentes théories permettant de modéliser certaines formes d'imperfection issues de la connaissance, tels que l'imprécision, le vague, l'incertitude. . .

Ces imperfections se situent à différents niveaux. L'imprécision est généralement liée au contenu de l'information alors que l'incertitude concerne la vraisemblance de l'information. Prendre en compte ces différentes formes d'imperfection a permis d'améliorer, d'une part la qualité de la représentation des connaissances, et d'autre part la compréhension des résultats issus de systèmes d'apprentissage automatique.

2.1 L'imprécision

La théorie des sous-ensembles flous introduite par Zadeh [41], a permis une modélisation simple des connaissances imprécises. Cette imprécision exprime l'appartenance partielle à une classe, ou encore la continuité de passage d'un état à un autre, etc

2.1.1 Définition d'un sous-ensemble flou

Définition 2.1.1 *Un sous-ensemble flou A sur un référentiel Ω est caractérisé par une fonction d'appartenance μ_A qui associe à chaque élément x de Ω un nombre réel dans l'intervalle $[0, 1]$.*

$$\mu_A : \Omega \longrightarrow [0, 1]$$

La valeur $\mu_A(x)$ représente le degré d'appartenance de x à A .

La théorie des sous-ensembles flous se place comme une généralisation de la théorie des ensembles. Les opérations ensemblistes sont étendues aux sous-ensembles flous de la manière suivante :

- Deux sous-ensembles flous A et B sont dits égaux si et seulement si leurs fonctions caractéristiques sont identiques,

$$\forall x \in \Omega, \quad \mu_A(x) = \mu_B(x)$$

- A est un sous-ensemble de B si et seulement si,

$$\forall x \in \Omega, \quad \mu_A(x) \leq \mu_B(x)$$

- L'intersection $A \cap B$ de A et B est le plus grand sous-ensemble contenu respectivement dans A et B . Sa fonction caractéristique est définie par :

$$\forall x \in \Omega, \quad \mu_{A \cap B} = \min(\mu_A(x), \mu_B(x))$$

- L'union $A \cup B$ de A et B est le plus petit sous-ensemble contenant respectivement A et B . Sa fonction caractéristique est définie par :

$$\forall x \in \Omega, \quad \mu_{A \cup B} = \max(\mu_A(x), \mu_B(x))$$

- Le complément \bar{A} d'un sous-ensemble flou A , est défini par :

$$\forall x \in \Omega, \quad \mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Nous verrons dans la section suivante d'autres manières de définir les opérateurs d'union, d'intersection ou encore de complément. Les propriétés classiques des ensembles ordinaires, telles que les lois de Morgan, la distributivité des opérateurs d'union et d'intersection, etc... sont conservées pour les opérateurs définis ci-dessus, à l'exception des expressions suivantes,

- $A \cap \bar{A} = \emptyset$, *non-contradiction*

- $A \cup \bar{A} = \Omega$, tiers exclus

Le cardinal, le support, le noyau ou encore la hauteur d'un sous-ensemble flou A sont des caractéristiques permettant de décrire partiellement la forme du sous-ensemble. Ces caractéristiques répondent aux définitions suivantes :

- le **cardinal** d'un sous-ensemble flou A d'un ensemble de référence fini Ω donne un aperçu du degré d'inclusion de A dans Ω , et correspond à la valeur :

$$|A| = \sum_{x \in \Omega} \mu_A(x)$$

- le **support** d'un sous-ensemble flou A , indique l'ensemble des éléments de Ω appartenant un peu à A , soit :

$$\text{support}(A) = \{x \in \Omega, \mu_A(x) > 0\}$$

- le **noyau** de A correspond à l'ensemble des éléments de Ω appartenant totalement à A , et

$$\text{noyau}(A) = \{x \in \Omega, \mu_A(x) = 1\}$$

- la **hauteur** d'un sous-ensemble flou A noté h , indique le degré d'appartenance maximale d'un élément de Ω à A ,

$$h(A) = \sup_{x \in \Omega} \mu_A(x)$$

Il existe une définition plus générale des sous-ensembles flous, où l'ensemble d'appartenance n'est pas forcément l'intervalle $[0, 1]$ mais un ensemble structuré permettant de conserver les propriétés élémentaires de la théorie des ensembles.

2.1.2 Normes et conormes triangulaires

Les normes et conormes triangulaires définissent des familles d'opérateurs vérifiant des propriétés qui caractérisent respectivement l'intersection et l'union. Selon le domaine d'application, d'autres opérateurs que le **min** et le **max** pourront être utilisés comme opérateurs d'intersection ou d'union.

Définition 2.1.2 Une norme triangulaire est une application \top de $[0, 1] \times [0, 1]$ dans $[0, 1]$ vérifiant :

- i) $\top(x, y) = \top(y, x)$ (commutativité)
- ii) $\top(x, \top(y, z)) = \top(\top(x, y), z)$ (associativité)
- iii) $\top(x, y) \leq \top(z, t)$ si $x \leq z$ et $y \leq t$ (monotonie)

$$\text{iv) } \top(x, 1) = x$$

(élément neutre 1)

Les normes et conormes triangulaires ne sont pas toutes compatibles avec les lois classiques de la théorie des ensembles. La monotonie de ces opérateurs par rapport à l'inclusion est respectée. Ainsi, une intersection définie avec une norme \top vérifie :

$$A \subseteq B \implies A \cap_{\top} C \subseteq B \cap_{\top} C.$$

Parallèlement, une union définie par une conorme vérifie :

$$A \subseteq B \implies A \cap_{\perp} C \subseteq B \cap_{\perp} C.$$

Normes et conormes triangulaires vérifient les mêmes propriétés de base et se distinguent au niveau de leur élément neutre.

Définition 2.1.3 Une conorme triangulaire est une application \perp de $[0, 1] \times [0, 1]$ dans $[0, 1]$ vérifiant :

$$\text{i) } \top(x, y) = \top(y, x)$$

(commutativité)

$$\text{ii) } \top(x, \top(y, z)) = \top(\top(x, y), z)$$

(associativité)

$$\text{iii) } \top(x, y) \leq \top(z, t) \text{ si } x \leq z \text{ et } y \leq t$$

(monotonie)

$$\text{iv) } \top(x, 0) = x$$

(élément neutre 0)

L'application \top définie ci-dessous est un exemple de norme appelée norme de Weber.

$$\top_{Weber}(x, y) = \begin{cases} x & \text{si } y = 1 \\ y & \text{si } x = 1 \\ 0 & \text{sinon} \end{cases}$$

Sa conorme duale par rapport à l'opérateur de négation usuel $n(x) = 1 - x$ est l'application suivante :

$$\perp_{Weber}(x, y) = \begin{cases} x & \text{si } y = 0 \\ y & \text{si } x = 0 \\ 1 & \text{sinon} \end{cases}$$

Comme pour les opérateurs d'union et d'intersection, les opérateurs de complément forment une famille [7] dont chaque élément est une fonction n de $[0, 1]$ dans $[0, 1]$ vérifiant :

$$- n(0) = 1 \text{ et } n(1) = 0,$$

$$- x \geq y \implies n(x) \leq n(y).$$

Cette négation est involutive si $n(n(x)) = x$ pour tout x de Ω .

La fonction dite "para-consistant complément" utilisé par Prade dans [32] est un exemple de négation involutive, ou :

$$n(x) = \begin{cases} 1 & \text{si } x < 1 \\ 0 & \text{si } x = 1 \end{cases}$$

2.1.3 Les α coupes

Une α -coupe d'un sous-ensemble flou A , représente le sous-ensemble (ordinaire) des éléments de Ω qui appartiennent au sous-ensemble A , avec un degré d'appartenance supérieur de α .

$$A_\alpha = \{x \in \Omega, \mu_A(x) \geq \alpha\}$$

Les coupes α et β d'un même sous-ensemble A sont ordonnées de manière inverse aux niveaux des coupes, c'est à dire

$$\alpha < \beta \implies A_\beta \subseteq A_\alpha$$

La compatibilité des opérateurs d'intersection, d'union et d'inclusion entre les ensembles ordinaires et flous, confère aux α -coupes les propriétés suivantes :

- $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$
- $(A \cup B)_\alpha = A_\alpha \cup B_\alpha$
- Si $A \subset B$ alors $A_\alpha \subset B_\alpha$

Dans le cas d'un ensemble de référence Ω fini, un sous-ensemble flou peut être décrit par un ensemble fini de coupe de niveau α .

Théorème de décomposition – *Tout sous-ensemble flou A peut être décomposé sous la forme suivante où apparaissent des produits de sous-ensembles ordinaires par des coefficients α_i :*

$$A = \bigcup_i \alpha_i A_{\alpha_i}$$

Sur un ensemble de référence $\Omega = \{a, b, c, d, f\}$, voici un exemple de décomposition :

$$\begin{aligned} \{0.2/a, 0/b, 0.2/c, 1/d, 0.7/f\} &= 0.2 \{1/a, 0/b, 1/c, 1/d, 1/f\} \\ &\cup 0.7 \{0/a, 0/b, 0/c, 1/d, 1/f\} \\ &\cup 1 \{0/a, 0/b, 0/c, 1/d, 0/f\} \end{aligned}$$

2.2 L'incertitude

Les théories de probabilité, des possibilités et de l'évidence développées dans cette section, font parties des approches les plus courantes pour l'attribution d'une notion de *vraisemblance* à des éventualités dont l'issue est incertaine.

2.2.1 La théorie des probabilités

Les probabilités permettent de quantifier la *vraisemblance* d'un événement aléatoire, comme une fréquence limite de répétition de l'événement, réalisé dans des conditions d'expériences identiques. Contrairement au langage courant, la notion de *hasard* relative à la réalisation d'un événement, est liée à l'expérimentation et définir une probabilité nécessite de connaître toutes les éventualités possibles de l'expérience ou de l'observation.

Définition 2.2.1 Soient Ω un univers et $\mathcal{P}(\Omega)$ l'ensemble de ses parties.

On appelle probabilité sur Ω , une application P de $\mathcal{P}(\Omega)$ dans $[0, 1]$ vérifiant :

$$i) P(\Omega) = 1$$

$$ii) \forall A \in \mathcal{P}(\Omega), \forall B \in \mathcal{P}(\Omega), \quad A \cap B = \emptyset \implies P(A \cup B) = P(A) + P(B).$$

Les mesures de probabilité vérifient les propriétés suivantes :

Proposition 2.2.1 $\forall A \in \mathcal{P}(\Omega), \forall B \in \mathcal{P}(\Omega),$

$$1. P(\emptyset) = 0$$

$$2. B \subseteq A \implies P(A) \leq P(B)$$

$$3. P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$4. P(A) + P(\bar{A}) = 1$$

La probabilité d'un événement A est déterminée par celle de ses singletons, où

$$\forall A \in \mathcal{P}(\Omega), \quad P(A) = \sum_{\omega \in \Omega} P(\{\omega\})$$

Les mesures de probabilité nécessitent de connaître toutes les probabilités élémentaires. Lorsque les événements suivent des lois mal définies, l'incertitude de leur réalisation se formalise difficilement à l'aide de mesures de probabilité.

Pour la théorie des probabilités, tous les événements sont possibles mais pas forcément probables. Cependant dans certains cas, il est souhaitable de différencier la croyance en un événement de la certitude de sa réalisation. De plus, la propriété 4.

issue de l'axiome d'additivité *ii*) affirme que la probabilité d'un événement détermine celle de l'événement contraire. Or il est parfois possible de croire en un événement et en son contraire.

2.2.2 La théorie des possibilités

La théorie des possibilités proposée par Zadeh [42] et développée par Dubois et Prade [13] a permis le traitement de connaissances incertaines de nature non probabiliste. Un événement A est caractérisé par deux fonctions dont l'une mesure la possibilité de l'événement et l'autre sa certitude.

Définition 2.2.2 Une mesure de possibilité définie sur ensemble de référence fini Ω , est une fonction Π de $\mathcal{P}(\Omega)$ dans $[0, 1]$ telle que,

$$i) \Pi(\emptyset) = 0$$

$$ii) \Pi(\Omega) = 1$$

$$iii) \forall A \in \mathcal{P}(\Omega), \forall B \in \mathcal{P}(\Omega), \Pi(A \cup B) = \max(\Pi(A), \Pi(B))$$

Comme pour les probabilités, la possibilité d'un événement A peut être construite à partir des possibilités des événements élémentaires de Ω . Cette distribution, appelée *distribution de possibilité*, est une fonction π de Ω dans $[0, 1]$, vérifiant la condition de normalisation $\sup_{\omega \in \Omega} \pi(\omega) = 1$. La possibilité d'un événement A peut alors être définie comme la possibilité maximale des éléments qui la composent, c'est à dire :

$$\Pi(A) = \sup_{\omega \in A} \pi(\omega)$$

Définition 2.2.3 Une mesure de nécessité définie sur ensemble de référence fini Ω , est une fonction N de $\mathcal{P}(\Omega)$ dans $[0, 1]$ telle que,

$$i) N(\emptyset) = 0$$

$$ii) N(\Omega) = 1$$

$$iii) \forall A \in \mathcal{P}(\Omega), \forall B \in \mathcal{P}(\Omega), N(A \cap B) = \min(N(A), N(B))$$

Dans la pratique, pour que la possibilité et la nécessité se complètent dans l'estimation de l'incertitude d'un événement, ces mesures vérifient la contrainte suivante :

$$\forall A \in \mathcal{P}(\Omega), \quad N(A) = 1 - \Pi(\bar{A})$$

Ainsi, plus un événement est possible, moins son *contraire* est certain. Dans ce cas, ces deux mesures vérifient les propriétés décrites ci-dessous,

$$\forall A \in \mathcal{P}(\Omega) \quad - \Pi(A) \geq N(A)$$

$$- \max(\Pi(A), 1 - N(A)) = 1$$

Ces propriétés assurent qu'un événement doit être totalement possible pour être au moins un peu certain.

2.2.3 La théorie de l'évidence

La théorie de l'évidence de Dempster-Shafer [38] est construite sur un ensemble de référence fini Ω à partir du degré de croyance attribué à des parties de Ω .

Cette distribution, appelée masse de probabilité, est l'application

$$m : \mathcal{P}(\Omega) \longrightarrow [0, 1]$$

vérifiant les propriétés :

$$- m(\emptyset) = 0$$

$$- \sum_{A \subseteq \Omega} m(A) = 1$$

Chaque partie de Ω est affectée d'une masse qui est déterminée par l'utilisateur. Contrairement à une distribution de probabilité, la masse d'un sous-ensemble A ne dépend pas des masses attribuées aux événements élémentaires. Toutes parties de Ω possédant une masse non nulle est appelée *élément focal* de m . A partir de cette masse de probabilité, l'incertitude d'un événement est déterminée par sa *crédibilité* et sa *plausibilité*.

- La *crédibilité* d'un événement A est la somme de toutes les croyances aux événements qui le composent. La fonction de *crédibilité* Cr est définie sur Ω par :

$$\forall A \in \mathcal{P}(\Omega), \quad Cr(A) = \sum_{B \subseteq A} m(B)$$

- La *plausibilité* d'un événement A est la somme de toutes les croyances aux événements en rapport avec A . La fonction de *plausibilité* Pl est définie sur Ω par :

$$\forall A \in \mathcal{P}(\Omega), \quad Pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$$

Plausibilité et crédibilité sont liées par la propriété $Pl(A) + Cr(\bar{A}) = 1$. Cette propriété permet d'expliquer la plausibilité d'un événement A comme la somme des

événements ne rendant pas l'événement contraire \bar{A} crédible. Crédibilité et plausibilité vérifient les propriétés suivantes :

Proposition 2.2.2 $\forall A \in \mathcal{P}(\Omega)$, $\forall B \in \mathcal{P}(\Omega)$,

1. $Cr(\emptyset) = 0$ et $Pl(\emptyset) = 0$
2. $Cr(\Omega) = 1$ et $Pl(\Omega) = 1$
3. $A \subseteq B \implies Cr(A) \leq Cr(B)$ et $Pl(A) \leq Pl(B)$
4. $Pl(A) \geq Cr(A)$
5. $Cr(A) + Cr(\bar{A}) \leq 1$ et $Pl(A) + Pl(\bar{A}) \geq 1$

En fonction de la distribution des masses de probabilité, les fonctions de crédibilité et de plausibilité coïncident avec les deux autres mesures d'incertitude.

- Lorsque les éléments focaux sont restreints aux singletons de Ω , la fonction de crédibilité est une mesure de probabilité vérifiant :

$$\forall A \in \mathcal{P}(\Omega), Cr(A) = Pl(A) = \sum_{\omega \in \Omega} m(\{\omega\})$$

- Lorsque les éléments focaux $\{f_1, f_2, \dots, f_n\}$ sont des sous-ensembles emboîtés,

$$f_1 \subset f_2 \dots \subset f_n$$

les fonctions de crédibilité et de plausibilité correspondent respectivement à une mesure de nécessité et à une mesure de possibilité définies à partir de la distribution de possibilité π , où :

$$\forall x \in \Omega, \quad \pi(x) = \begin{cases} 0 & \text{si } x \notin \Omega \\ \sum_{i=k+1}^n m(A_i) & \text{si } x \notin f_k \text{ et } x \in f_{k+1} \end{cases}$$

La théorie des sous-ensembles flous et la théorie des possibilités vont contribuer au développement de méthodes d'apprentissage flou. Nous présenterons quelques une de ces méthodes dans le chapitre suivant.

Chapitre 3

Les méthodes de discrimination floue

Sommaire

3.1	Arbre de décision flou	23
3.1.1	Le système FID3	23
3.1.2	Le système SAFI	26
3.2	Processus d'induction avec quantificateurs linguistiques . .	28

Contrairement aux méthodes vues précédemment, les méthodes de ce chapitre traitent de données imprécises ou incertaines. Ces méthodes sont des extensions de méthodes de classification ou discrimination d'Analyse de Donnée ou d'Apprentissage Symbolique. Elles se différencient principalement par les critères et heuristiques utilisés pour le traitement de l'information.

Nous présenterons deux méthodes de construction d'arbre de décision qui prennent en compte l'incertitude dans les données pour la première et l'imprécision pour la deuxième. Ensuite nous verrons une méthode de discrimination appliquée à des sous-ensembles flous d'exemples et de contre-exemples.

3.1 Arbre de décision flou

3.1.1 Le système FID3

Généralement, les méthodes de construction d'arbres de décision minimisent la taille des arbres et en particulier la profondeur. Par opposition à ces méthodes, l'approche *FID₃* de Rives [35] consiste à produire l'arbre le plus informatif sans prendre en considération la taille du développement.

Notations et terminologie

E représente une table d'exemples décrits sur un ensemble fini A_1, \dots, A_n de n attributs, où chaque exemple est en plus caractérisé par une distribution de possibilité sur un attribut de sortie. Sur l'exemple suivant, les classes d'individus à séparer sont caractérisées par les modalités de l'attribut *Appearance*.

Exemple 3.1.1

	<i>Height</i>	<i>Weight</i>	<i>Age</i>	<i>Appearance</i>		
				<i>bad</i>	<i>normal</i>	<i>good</i>
ex_1	<i>tall</i>	<i>fat</i>	<i>lover</i>	0.78	0.33	0
ex_2	<i>tall</i>	<i>fat</i>	<i>lover</i>	1	0.11	0

Tableau E d'exemples

Pour chaque attribut A_i , $U(A_i)$ correspond à une U -incertitude [23] basée sur la distribution de possibilité relative à l'attribut A_i .

Par analogie, $U(A_1, \dots, A_n)$ est l'union des U -incertitudes des variables impliquées.

Les mesures d'information

La méthode se base sur des mesures de transmission d'information. Pour un tableau de données E décrit sur p attributs A_1, \dots, A_p , la fonction :

$$\hat{I}(E) = \frac{1}{p+1} (U(A_0) + \dots + U(A_p) - U(A_0, \dots, A_p))$$

mesure l'information potentielle contenue dans la base d'exemples E .

Pour définir une fonction mesurant le gain d'information pour chaque attribut, l'auteur détermine une fonction de transmission d'information :

$$\hat{I}(A) = \sup_{x \in A} [\top(\hat{I}(E_x), Poss(x))]$$

où

- \top est une norme quelconque,
- $Poss(x)$ est le maximum des possibilités de la variable x de A par rapport à la table E ,
- $\hat{I}(E_x)$ est l'information calculée sur la partition E_x de E relative à la variable x de A .

La fonction de gain d'information pour une table d'exemples E se mesure par :

$$\partial \hat{I}(A) = \hat{I}(E) - \hat{I}(A)$$

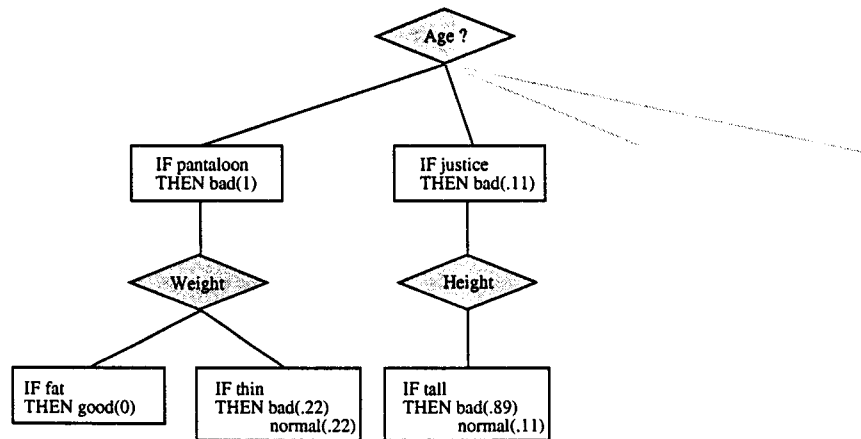


FIG. 3.1 – Forme de l'arbre de décision flou possibiliste

Interprétation d'une branche

Après avoir choisi l'attribut le plus informatif A , il faut pouvoir donner un sens aux branches en fonction des variables de l'attribut de sortie. Il s'agit de déterminer la variable de sortie caractérisant le mieux chaque branche. Ainsi, pour chaque partition E_x où x est une variable de A , l'auteur évalue la *distinguabilité* de chaque variable de l'attribut de sortie s par la fonction

$$D(B_s) = \frac{\sum_{a \in B_s} |2\mu(a) - 1|}{|B_s|}$$

où B_s est le sous-ensemble flou correspondant à la variable de sortie s tel que bad . D mesure avec quel degré un sous-ensemble B_s et son complémentaire \bar{B}_s se différencient. Ainsi, plus une valeur est proche de 1 ou 0, plus leur *distinguabilité* est forte.

Méthodes

L'algorithme FID_3 se divise en deux étapes :

- L'attribut A maximisant la fonction de gain d'information $\partial \hat{I}(A)$ est sélectionné.
- Pour chaque partition issue des variables de l'attribut A , l'algorithme choisit la variable de sortie maximisant le critère de *distinguabilité*. Comme le montre la figure 3.3, lorsque s est la variable de sortie caractérisant le mieux la valeur x , une branche peut s'interpréter par : Si x alors $s(Poss(x))$ où $Poss(x)$ est la possibilité de x relativement à la partition E_x .

Ces deux étapes sont répétées sur chaque partition obtenue. Chaque branche se termine lorsque tous les attributs ont été sélectionnés.

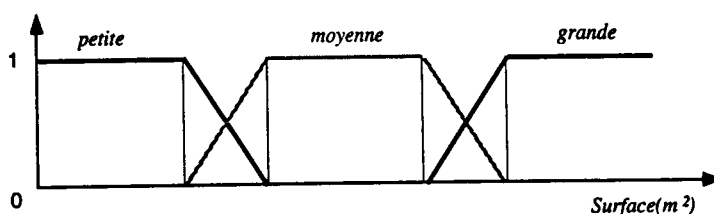


FIG. 3.2 – Exemple de partition floue

3.1.2 Le système SAFI

Le système SAFI ou système d'apprentissage flou inter-actif, développé par Ramdani [4], est une méthode basée sur les arbres de décisions permettant de traiter des attributs numériques caractérisés par des sous-ensembles flous. Le système SAFI est composé de plusieurs modules concernant la représentation des connaissances imprécises, l'optimisation des paramètres de cette représentation, le processus d'apprentissage, ainsi qu'une procédure de filtrage.

L'entropie-star

L'entropie-star est une généralisation de l'entropie conditionnelle classique aux probabilités floues. Pour l'attribut de décision par rapport à un attribut A , cette entropie notée I^*_A , est définie par :

$$I^*_A = - \sum_{k=1}^m P^*(v_k) \sum_{i=1}^l P^*(c_i|v_k) \log P^*(c_i|v_k)$$

où

- $\{c_1, \dots, c_l\}$ est l'ensemble des valeurs de l'attribut de décision.
- $P^*(v_k)$ mesure la probabilité floue de la modalité v_k de l'attribut A .
- $P^*(c_i|v_k)$ est la probabilité conditionnelle floue de la modalité c_i en fonction de la modalité v_k .

Cette mesure exprime l'indétermination à prendre une décision lorsqu'il y a de l'imprécision dans la répartition des valeurs de l'expérience réalisée.

Représentation de la connaissance

La représentation de la connaissance se fait à travers un module inter-actif dont le rôle est de décomposer pour tous les attributs numériques leur domaine de valeurs en partition floue, en fonction des valeurs les plus représentatives du domaine pour l'expert. Les sous-ensembles flous utilisés par le système sont représentés par des fonctions de forme trapézoïdale. Pour un attribut numérique, les valeurs caractérisées par des

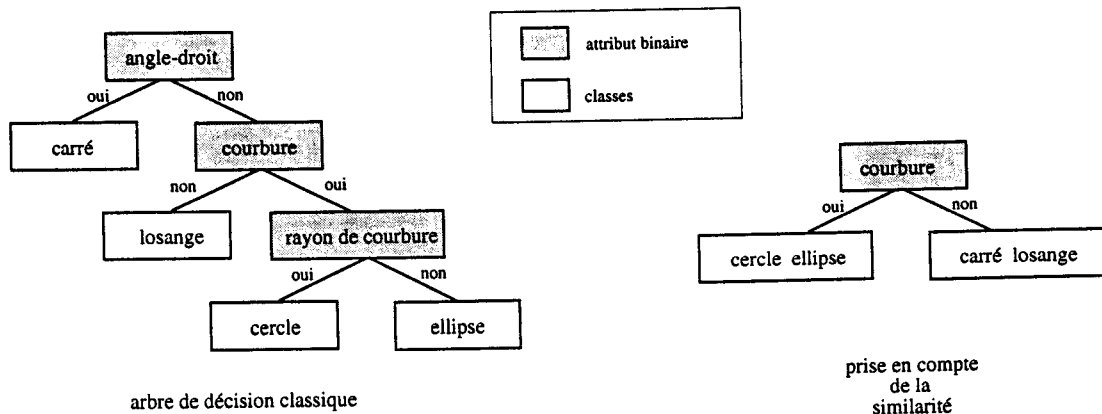


FIG. 3.3 - Effet des similarités

fonctions continues, sont transformées en valeurs symboliques, définies comme un sous-ensemble flou de référence la partition. Chaque valeur affectée de son degré de satisfaisabilité, exprime la compatibilité des valeurs. En minimisant une entropie-star, le module d'optimisation va permettre d'obtenir une partition floue plus informative. Cette entropie-star est fonction de l'écart minimal entre les noyaux de la partition encore appelé degré d'étalement minimal.

Développement de l'arbre de décision

La procédure de construction de l'arbre de décision est identique à celles des arbres de décision classiques. Le critère de sélection du meilleur attribut discriminant doit minimiser l'entropie-star.

Pour un nœud associé à un attribut symbolique, les branches correspondent aux différentes valeurs de l'attribut. Par contre, lorsqu'il est associé à un attribut numérique, les branches dépendent de la partition floue issue des modules précédents. A chaque nœud terminal de l'arbre est associée une probabilité conditionnelle floue correspondant au cheminement du nœud dans l'arbre. Le module de filtrage permet d'insérer de nouveaux exemples dans l'arbre en filtrant les différentes branches de l'arbre.

L'introduction de relations de similarité dans l'évaluation de l'entropie, va permettre de minimiser le développement de l'arbre. L'entropie basée sur ces relations favorise les attributs qui discriminent les décisions selon les classes de similarité. Cet arbre de décision permet à terme de cibler les recherches dans une des classes de similarités (fig. 3.2).

3.2 Processus d'induction avec quantificateurs linguistiques

Kacprzyk et Iwański proposent dans [20] une méthode de discrimination permettant de prendre en compte d'une part l'imprécision par l'usage de la logique floue, d'autre part les erreurs de classement par le biais de propositions linguistiques quantifiées.

Notations et terminologie

Une valeur d'attribut est un sélecteur $[A, r, 'a']$ où

- A est un attribut,
- r est une relation éventuellement floue telle que $'=, '≤, …$
- a est une valeur floue.

Une description est formée d'une liste de sélecteurs. Un complexe est une conjonction de sélecteurs. Enfin, un concept est une union de complexes.

Les observations sont réparties en deux classes, la classe des exemples P et la classe des contre-exemples N . Les auteurs posent $N = \bar{P}$.

L'imprécision

Le degré d'adéquation entre une description x et un sélecteur s est le résultat de procédures de filtrage flou entre les valeurs floues de x et s .

Le degré de recouvrement d'une description x par un complexe est mesuré comme la conjonction des degrés d'adéquation de leurs sélecteurs.

Pour finir, l'appartenance d'une description x à un concept R est l'union des recouvrements de x par chaque complexe de R .

Les opérateurs d'union et de conjonction sont respectivement des conormes et des normes.

L'erreur

Les auteurs utilisent des quantificateurs linguistiques tels que *"la plupart"*, *"peu de"* pour mesurer une complétude et une consistance partiellement satisfaites. Les propositions suivantes formalisent cette complétude et consistance partielles,

"La plupart des exemples sont dans R "

et *"peu de contre-exemples sont dans R "*

Ainsi des erreurs telles que de mauvaises descriptions ou encore des descriptions mal réparties entre les exemples et les contre-exemples, sont écartées de l'évaluation de la qualité du concept R dans la mesure du taux d'erreurs accepté par le quantificateur.

Le critère mesurant la qualité du concept R sera noté $truth[R]$.

Méthodes

Initialement, le concept R et le complexe C sont vides. L'algorithme se divise en trois étapes principales,

- Un complexe typique τ est construit relativement au concept R .

Les auteurs recherchent une sorte de prototype sous la forme d'un complexe dont chaque sélecteur recouvre au mieux tous les exemples qui ne sont pas déjà recouverts par le concept R .

- A partir de ce complexe typique, les auteurs recherchent un exemple typique x^* qui soit l'exemple le plus caractéristique de P (au sens de l'appartenance) et le plus proche du complexe x^* , c'est à dire que l'appartenance moyenne de x^* à τ soit maximale.

- Pour finir, un complexe C est construit en fonction des sélecteurs de x^* .

C est obtenu par ajouts successifs de sélecteurs de x^* qui maximisent le critère $truth[R \cup C]$. La construction de C s'arrête dès qu'aucun sélecteur restant de x^* n'augmente plus la qualité de $R \cup C$.

L'algorithme continue tant que la qualité $R \cup C$ augmente autant au niveau de la complétude qu'au niveau de la consistance. Dans le cas contraire, l'algorithme s'arrête et R représente une caractérisation de la classe des exemples.

Les méthodes proposées dans ce chapitre présentent certains inconvénients lorsqu'il s'agit de représenter des données issues des Sciences Naturelles. Par exemple, les relations de composition entre attributs ne sont pas prises en compte, ou encore certaines formes de nuances d'attribut ne se modélisent pas avec l'intervalle $[0, 1]$. Le modèle de représentation de données présenté dans le chapitre suivant tient compte de ces deux aspects.

Chapitre 4

La représentation des données

Sommaire

4.1	Les nuances	32
4.2	Rappels sur les treillis	34
4.3	La représentation des propriétés	35
4.3.1	Les attributs de type simple	36
4.3.2	Les attributs de type symbolique	36
4.3.3	Les attributs de type taxonomique	38
4.4	Les objets structurés	42
4.4.1	Les attributs de type structuré	42
4.4.2	Structure sur l'ensemble des valeurs	43
4.5	Représentation structurelle des objets	45

Savoir représenter la connaissance est une étape préalable à un processus d'apprentissage. Pour un être humain, apprendre nécessite de pouvoir communiquer et manipuler de l'information de manière intelligible. Pour qu'une machine puisse *apprendre* et *utiliser* automatiquement de la connaissance d'un domaine quelconque, il faut pouvoir de manière analogue lui communiquer cette connaissance. Le modèle de représentation des données définit un protocole de communication entre l'homme et la machine et plus précisément entre l'expert et la machine.

Comme le souligne Conruyt [10], l'action de décrire est une abstraction permettant de passer du monde réel dans un langage formel et représente un biais d'apprentissage. La manière de décrire un objet dépend de celui qui observe. Le modèle de représentation des connaissances est un outil permettant à l'expert de définir son modèle de description. Pour cette raison, différents langages de description ont été développés, tels que le calcul des prédicats, les descriptions hiérarchiques, les réseaux sémantiques, les représentations objets ou "frames", les logiques de l'incertain [39] qui sont plus ou moins adaptés

à certains domaines d'application.

Le modèle des objets structurés à valeurs imprécises a été proposé par H. Ralambondrainy [33, 34, 15], dans le but de prendre en compte certaines particularités de données issues des Sciences Naturelles que sont :

- *le jugement du spécialiste :*

Lors de la description d'un individu, les experts utilisent naturellement des termes linguistiques pour qualifier certaines propriétés de l'individu. Selon les cas, ces termes correspondent à des intensités, des fréquences, des croyances. . . Les langages de représentation usuels ne permettent pas de formaliser simplement cette vision discrète de l'expert. Par exemple, l'incompatibilité entre les nuances de croyance *vraie* et *fausse* des figures 4.1 est difficilement représentable dans les modèles classiques. La conséquence immédiate d'une mauvaise représentation, est la perte de compréhensibilité au sens de Michalsky [30].

- *l'aspect structurel des données :*

Dans bien des domaines et en particulier pour les sciences naturelles, il existe des relations de dépendance entre les attributs exprimant la connaissance du domaine ou "background knowlegde". Par exemple, une éponge est composée d'un corps et d'un pédoncule. Le pédoncule peut être torsadé ou non. Le corps d'une éponge est formé de micro-éléments et de macro-constituants décrits eux-mêmes par différentes propriétés,

$$\begin{array}{l} \text{éponge} = \langle \\ \quad \text{pédoncule,} \\ \quad \text{corps} = \langle \text{micro-éléments , macro-constituants} \rangle \rangle \end{array}$$

Cette relation de composition est difficilement retranscrite dans des modèles classiques par une représentation vectorielle des données. Les modèles orientés objets semblent plus adéquats pour représenter ce type de connaissance.

Nous allons présenter dans ce qui suit ce modèle des objets structurés . Par ailleurs, nous allons approfondir la modélisation des taxonomies nuancées et proposer un espace de représentation de structure d'objets.

4.1 Les nuances

Décrire un individu du monde réel nécessite de prendre en compte différents types d'imperfection. Ces imperfections se regroupent principalement en deux catégories qui sont l'imprécision et l'incertitude.

L'imprécision représente une connaissance sur les données :

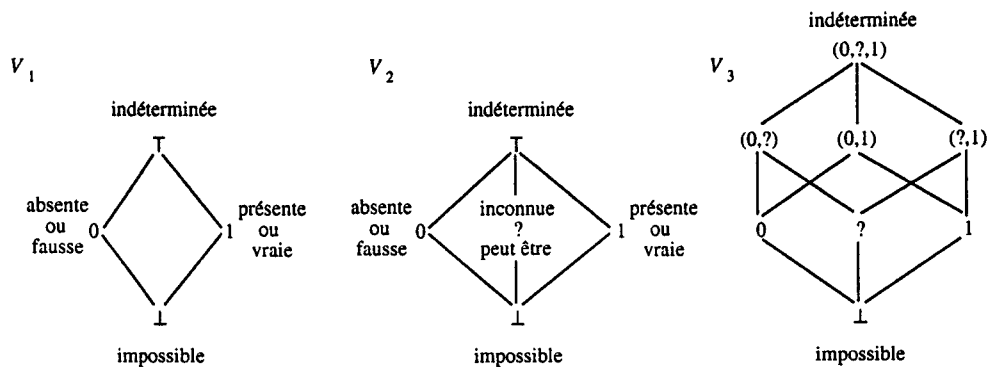


FIG. 4.1 – Des nuances de croyance

- approximative, caractérisée par des termes tels que “environ”, “presque”, “plus ou moins”,
- vague, exprimée par des concepts tels que “typiquement”, “habituellement”,
- quantitative, où la description est partiellement valide et qualifiée par “généralement”, “la plupart”, “peu de”.

L’incertitude elle, est de nature subjective et représente la croyance que l’on a d’un événement. Les termes tels “il est probable que”, “peut être” font référence à une connaissance préalablement acquise.

Ces différents types d’imperfection sont souvent présents lors de la description d’individus en sciences naturelles. Différentes raisons d’ordre matériel, financier, ou encore d’accessibilité à la ressource, etc. . . peuvent être à l’origine de mesures imprécises ou incomplètes. De plus, les descriptions faisant intervenir la connaissance de l’expert sont parfois à l’origine d’imprécisions (ex : *Amphidisques du type à umbella presque hémisphérique*).

La logique floue développée par L. Zadeh [41] permet une modélisation simple des descriptions imprécises ou vagues. Dans l’approche floue classique, chaque modalité d’une description est qualifiée par une valeur de l’intervalle $[0, 1]$. Dans l’approche que nous utilisons, une modalité est qualifiée par un terme linguistique appartenant à un treillis de nuances, telles que les nuances de croyance (fig. 4.1), d’intensité (fig. 4.2), de fréquence, etc. . . .

L’utilisation de treillis de nuances permet d’obtenir un langage de représentation de données plus proche du langage de description de l’expert, ce qui est un avantage pour la compréhensibilité des résultats.

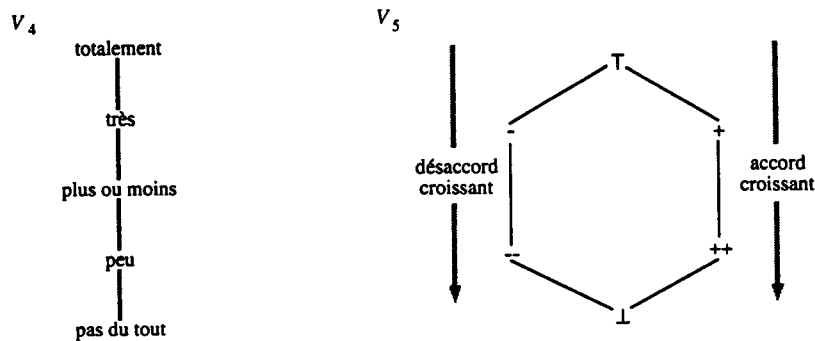


FIG. 4.2 – Des nuances d'intensité

De plus, l'utilisation de structure de treillis semble plus adéquate à certains types de description. Lorsqu'un individu est décrit, l'*absence* et la *présence* de valeurs pour une propriété, sont des informations de même importance. Dans le sens où, aucune de ces deux informations ne subsume l'autre, ces nuances doivent être incomparables. Les treillis de nuances de la figure 4.1 permettent une telle représentation. Ce type de description se retrouve fréquemment dans le cadre de questionnaires, où il est possible de nuancer la croyance d'une réponse.

Pour le treillis de croyance V_2 , la nuance "*peut être*" permet de modéliser l'*ignorance* comme une information à part entière, c'est à dire ni plus précise ni plus générale que les nuances *vraie* ou *fausse*. L'approche floue classique ne permet pas cette représentation car l'intervalle $[0, 1]$ est un ensemble totalement ordonné.

Dans notre approche, nous ne considérons que des attributs symboliques. Si les données comportent des attributs numériques, il est nécessaire de les coder sous forme symbolique ce qui peut entraîner une perte d'information. Pour le modèle des objets structurés, les attributs numériques sont transformés en attributs symboliques.

4.2 Rappels sur les treillis

Un treillis \mathcal{T} est un ensemble ordonné tel que chaque couple (x, y) d'éléments de \mathcal{T} possède un supremum (noté $x \vee y$) et un infimum (noté $x \wedge y$).

Le supremum d'un couple d'éléments de \mathcal{T} , appelé également borne supérieure, est le plus petit majorant commun à ces éléments. Leur infimum ou encore borne inférieure, est leur plus grand minorant commun.

Comme pour les opérations d'union et d'intersection sur les ensembles, les opérateurs \vee et \wedge confèrent aux treillis une structure algébrique [3, 1].

Toutes les lois sur les ensembles classiques ne sont pas généralisables pour un treillis quelconque. Par exemple, pour le treillis d'intensité V_5 de la figure 4.2,

- \vee et \wedge ne sont pas des opérateurs distributifs car,
 $(-\vee++) \wedge + = \top \wedge + = +$ alors que $(-\wedge+) \vee (++\wedge+) = \perp \vee ++ = ++$

- la négation n'est pas définie, il n'existe pas d'élément unique x dans V_5 tel que $x\vee+ = \top$ et $x\wedge+ = \perp$. En effet $-$ et $--$ sont solutions.

Définition 4.2.1

- x est un **successeur** de y (noté $x \succ y$) si et seulement si

$$\forall t \in \mathcal{T}, \quad y \leq t \leq x \implies t = y \text{ ou } t = x$$

- De manière duale, x est un **prédécesseur** de y (noté $x \prec y$) si et seulement si

$$\forall t \in \mathcal{T}, \quad x \leq t \leq y \implies t = y \text{ ou } t = x$$

- Un ensemble ordonné \mathcal{T} est dit de **longueur finie**, s'il existe un entier m vérifiant que toutes les chaînes de \mathcal{T} ont une longueur inférieure à m .

- Si \mathcal{T} est un treillis de longueur finie, alors \mathcal{T} possède un élément minimal (noté \perp) et un élément maximal (noté \top).

$$\forall x \in \mathcal{T}, \quad \left| \begin{array}{ll} x \wedge \perp = \perp & x \vee \perp = x \\ x \wedge \top = x & x \vee \top = \top \end{array} \right.$$

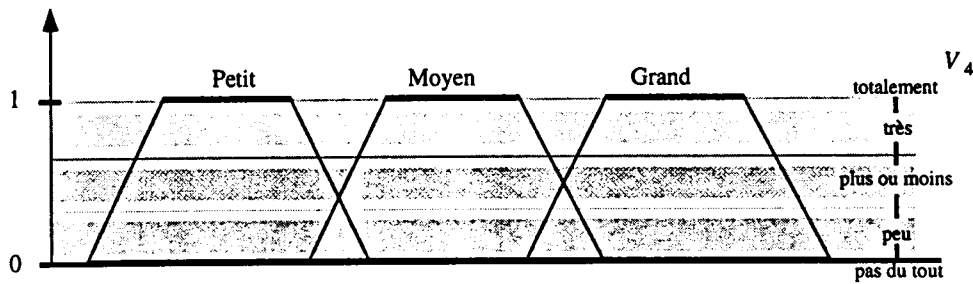
4.3 La représentation des propriétés

Les propriétés des objets sont décrites à l'aide d'attributs. Les attributs et leurs valeurs sont définis par les spécialistes du domaine. Nous supposons donné :

- un ensemble d'identificateurs $\mathcal{A} = \{A\}$ utilisés pour désigner une propriété.

- un ensemble de domaines de cardinal fini $\mathcal{D} = \{D\}$. Le domaine représente l'ensemble des valeurs possibles pour un attribut.

- un ensemble de nuances $\mathcal{V} = \{V\}$. Chaque élément d'un treillis de nuance V permet de qualifier une valeur d'attribut.

FIG. 4.3 – Représentation symbolique de l'attribut *Taille*

4.3.1 Les attributs de type simple

Les notions d'attribut et de valeur d'un attribut sont introduites de manière générale sans tenir compte de la nature des domaines.

Définition 4.3.1 *Un attribut de type simple est caractérisé par :*

- un identificateur A le désignant de manière unique,
- un domaine fini D regroupant l'ensemble des valeurs possibles,
- un treillis de nuances V permettant de qualifier une valeur,
- et l'ensemble des valeurs nuancées possibles pour l'attribut A , noté $[A]$.

Par abus de langage, un attribut $(A, D, V, [A])$ peut être désigné uniquement par son identificateur A .

Les attributs de type simple comprennent les attributs de type symbolique, taxonomique et d'une certaine manière numérique. En effet, ces derniers sont supposés préalablement discrétisés et seront considérés comme des attributs symboliques. L'attribut *taille* de la figure 4.3 est un exemple classique de discrétisation de valeurs numériques.

Dans la suite, l'ensemble $[A]$ des valeurs nuancées, sera défini comme un sous-ensemble de l'ensemble des applications de D dans V . L'ensemble $[A]$ représente toutes les descriptions *cohérentes* pour un attribut et sa définition varie en fonction de la nature de l'attribut.

4.3.2 Les attributs de type symbolique

Les domaines symboliques sont des ensembles de valeurs sans structure. Pour chaque attribut, le spécialiste définit le treillis de valuation qui doit être associé au domaine.

Définition 4.3.2 *Un attribut de type symbolique A de domaine D , qualifié par un treillis de nuance V possède comme ensemble de valeurs nuancées $[A]$, l'ensemble des*

applications de D dans V .

Pour un domaine $D = \{d_1, \dots, d_k, \dots, d_m\}$, une valeur nuancée μ de $[A] = V^D$ est une expression de la forme :

$$\mu = (n_1/d_1, \dots, n_k/d_k, \dots, n_m/d_m)$$

où $\forall k \in \{1, \dots, m\}$, $n_k \in V$.

Pour A. Kaufmann, "si l'on n'introduit plus la notion de complément, on peut bâtir une théorie des sous-ensembles flous pour n'importe quel sorte de treillis", [22]. Ainsi, une valeur nuancée est un sous-ensemble flou sur V de référence D .

L'ensemble des descriptions $[A]$ possède une structure de treillis induite par celle des nuances comme ensemble des applications d'un ensemble dans un treillis. Les opérateurs de ce treillis ($V^D, \leq, \vee, \wedge, \perp, \top$) s'expriment comme suit :

$\forall \mu \in [A], \forall \nu \in [A]$,

- \leq est la relation d'ordre partielle de $[A]$ où,

$$\mu \leq \nu \implies \forall d \in D \quad \mu(d) \leq \nu(d)$$

$\mu \leq \nu$ signifie que la description μ est moins générale ou plus précise que la description ν .

- μ et ν sont les opérateurs de supremum et d'infimum où,

$$\mu \vee \nu \text{ est l'application définie par } \forall d \in D \quad (\mu \vee \nu)(d) = \mu(d) \vee \nu(d)$$

$$\mu \wedge \nu \text{ est l'application définie par } \forall d \in D \quad (\mu \wedge \nu)(d) = \mu(d) \wedge \nu(d)$$

L'infimum et le supremum sont respectivement appelés opérateur de spécialisation et opérateur de généralisation.

- \perp et \top sont respectivement la borne inférieure et la borne supérieure du treillis des descriptions.

\perp est l'application qui a tout $d \in D$ associe la plus petite nuance \perp de V . La valeur \perp pour une description signifie que l'attribut lui est **inapplicable**.

\top est l'application qui a tout $d \in D$ associe la plus grande nuance \top de V . La valeur \top pour une description signifie que l'attribut lui est **indéterminé**.

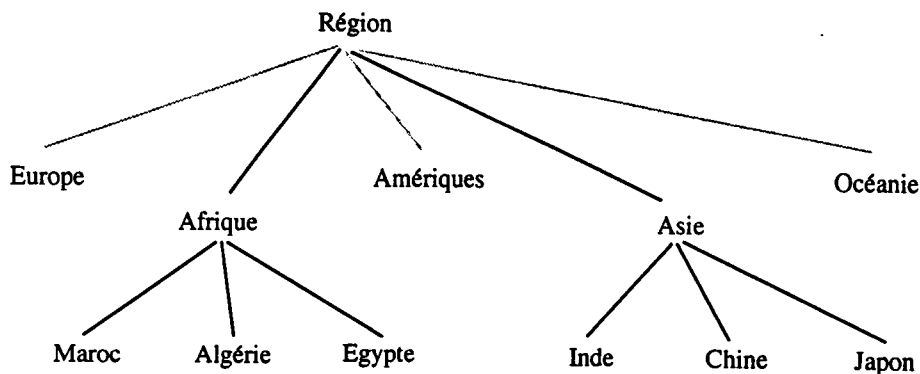


FIG. 4.4 – Description taxonomique de régions

4.3.3 Les attributs de type taxonomique

Les attributs taxonomiques sont relatifs à des domaines où les valeurs sont représentées par une structure arborescente et peut se formaliser comme hiérarchie de valeurs de la manière suivante :

Soit Ω un ensemble de feuilles. L'ensemble H des paliers est un ensemble de parties de Ω tel que,

- La hiérarchie H contient Ω .
- Les singletons formés des feuilles de Ω sont des paliers de H ,
- Pour tout palier h et h' de H , on a : $h \cap h' \neq \emptyset \implies h \subset h'$ ou $h' \subset h$.

Sur l'exemple de taxonomie de la figure 4.4, l'ensemble des pays constitue les feuilles de la hiérarchie et chaque région représente une partie de ces feuilles. Ainsi les paliers de la hiérarchie sont ordonnés par la relation d'inclusion.

Classiquement, chaque modalité d'une taxonomie est qualifiée par les valeurs "vraie" ou "fausse". L'intérêt d'une structure hiérarchique est la possibilité de pouvoir qualifier tous les paliers de la hiérarchie en fonction des nuances affectées à chaque feuille.

Soit μ une valeur hiérarchique, les paliers sont nuancés de la manière suivante :

$$\forall h \in H, \quad \mu(h) = \mu\left(\bigcup_{h' \subseteq h} h'\right) = \bigvee_{h' \subseteq h} \mu(h') \quad (*)$$

où \vee correspond au "ou" logique.

En d'autres termes, un palier de la hiérarchie est affecté de la valeur vraie dès qu'un de ses paliers fils est "vraie". Par exemple, lorsque la modalité *Chine* est "vraie"

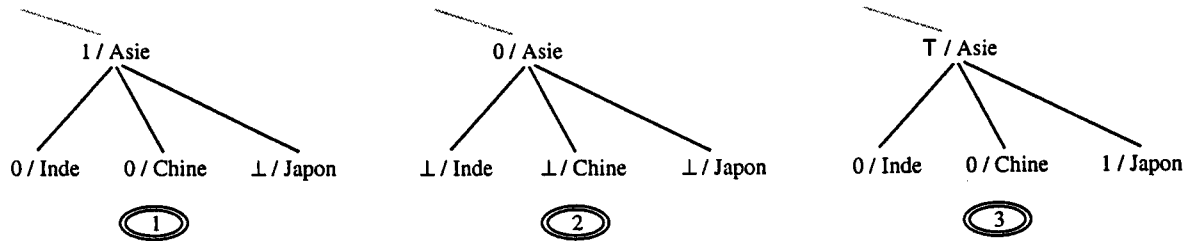


FIG. 4.5 – Instances hiérarchiques mal définies

pour une description, automatiquement la valeur *Asie* est “vraie” pour cette même description.

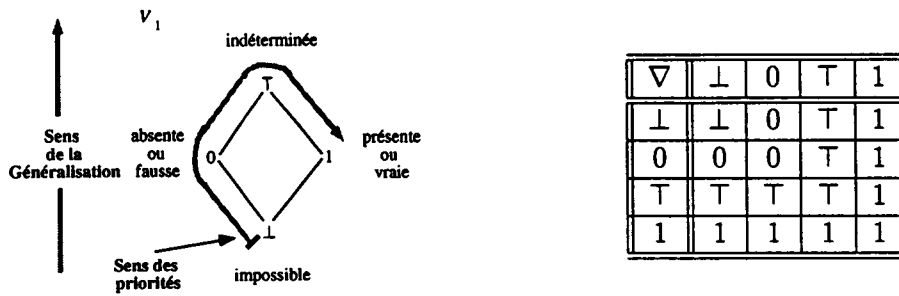
Lorsque l'on modélise les hiérarchies dans notre modèle, les nuances “vraie” et “fausse” sont représentées par la chaîne (“vraie” \leftarrow “fausse”). Ainsi le “ou” logique correspond à l'opérateur de borne supérieure du treillis de nuance.

Cependant, lorsqu'un treillis quelconque est utilisé pour nuancer les paliers, l'opérateur de borne supérieure ne convient pas forcément. Par exemple, en nuancant les régions par le treillis de croyances V_1 , on constate que dès que deux feuilles prennent comme valeurs “vraie” et “fausse”, on obtient par l'opérateur de généralisation un palier père “inconnu” or il serait souhaitable que comme pour les hiérarchies usuelles sa valeur soit “vraie”. Par exemple, la région *Asie* est “inconnue” lorsque la valeur *Japon* est “vraie” et la valeur *Chine* est “fausse”. Nous montrerons dans ce qui suit, qu'il est possible de définir des taxonomies nuancées sur le treillis de nuance V_1 , tout en respectant la structure des hiérarchies.

On considère un opérateur binaire ∇ sur les nuances du treillis V_1 , qui définit un ordre de priorité entre les différentes nuances. Cette ordre correspond à la chaîne croissante¹ $\{\perp, 0, \top, 1\}$.

L'idée est qu'un palier peut être qualifié par la nuance “vraie” dès qu'un de ses paliers fils est soit “vrai”, soit “indéterminé”. La notion de valeur hiérarchique cohérente

1. L'opérateur ∇ peut être vu comme le supremum de la chaîne $\{\perp, 0, \top, 1\}$. Par définition du supremum, ∇ est donc un opérateur commutatif et associatif. Par contre, les propriétés des treillis liées à l'ordre ne sont pas généralisables.

FIG. 4.6 – Définition de l'opérateur ∇

est introduite par la définition suivante :

Définition 4.3.3 - Une valeur hiérarchique nuancée cohérente est une application de V_1^D vérifiant² :

$$\text{Pour tout palier } h \text{ de } D, \begin{cases} \text{si } \mu(h) \neq 1 \text{ alors} & \mu(h) = \nabla_{h' \subset h} \mu(h') \\ \text{si } \mu(h) = 1 \text{ alors} & \nabla_{h' \subset h} \mu(h') = \top \text{ ou } \nabla_{h' \subset h} \mu(h') = 1 \end{cases}$$

- L'ensemble des descriptions taxonomiques nuancées $[A]$ est formé uniquement de valeurs hiérarchiques cohérentes.

L'instance 1 de la figure 4.5 n'est pas cohérente car la région *Asie* est *vraie* alors que tout ses pays sont qualifiés soit par la nuance *fausse*, soit par la nuance *impossible*.

L'instance 5 de la figure 4.7 illustre la particularité de la cohérence d'une valeur hiérarchique. La valeur *Asie* peut être *vraie* sans savoir quel pays en est l'origine.

Comme dans le cas des attributs symboliques, l'ensemble des applications de D dans V_1 possède une structure de treillis dont les opérateurs sont \leq , \vee , \wedge , \top , \perp . Ces opérateurs vont permettre de formaliser la structure de treillis de $[A]$.

Soient μ et ν deux éléments de $[A]$.

• $[A]$ est muni d'une relation d'ordre induite par celle de V_1 qui est identique à celle de V^D :

$$\mu \leq_{[A]} \nu \implies \mu \leq \nu$$

• La borne supérieure de $[A]$ notée $\top_{[A]}$ et la borne inférieure notée $\perp_{[A]}$ sont celles de V^D , car \top et \perp sont des valeurs hiérarchiques bien définies.

• Les opérateurs de supremum $\vee_{[A]}$ et d'infimum $\wedge_{[A]}$ du treillis $[A]$ sont définis ci-dessous. Dans les deux cas, il s'agit de démontrer l'existence et l'unicité des bornes

2. Le symbole \subset représente l'inclusion stricte

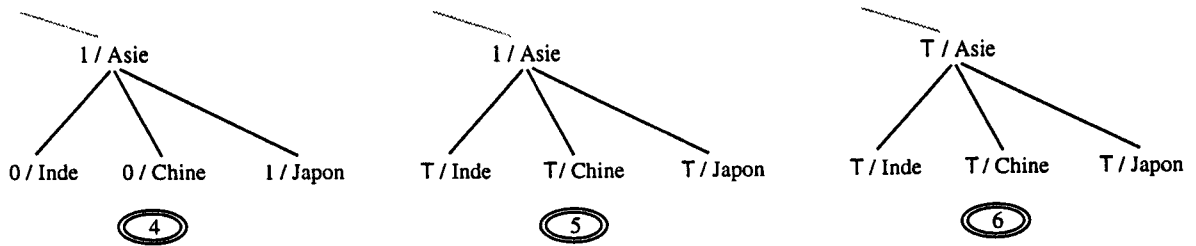


FIG. 4.7 – Instances hiérarchiques bien définies

supérieures ou inférieures de deux éléments.

Proposition 4.3.1 L'application $\vee_{[A]}$ de $[A] \times [A]$ dans V_1^D vérifiant,

$$\forall \mu, \nu \in [A], \quad \mu \vee_{[A]} \nu(h) = \begin{cases} \top & \text{si } (\mu \vee \nu)(h) = 1 \text{ et} \\ & \exists h' \in D, \quad h' > h \text{ et } (\mu \vee \nu)(h') = \top \\ (\mu \vee \nu)(h) & \text{sinon} \end{cases}$$

définit la borne supérieure des éléments μ et ν dans $[A]$ pour un attribut taxonomique.

preuve : voir annexe A.

Proposition 4.3.2 L'application $\wedge_{[A]}$ de $[A] \times [A]$ vérifiant

$$\forall \mu, \nu \in [A], \quad \mu \wedge_{[A]} \nu = \sup \{t \in [A], \quad t \leq \mu \text{ et } t \leq \nu\}$$

définit la borne inférieure des éléments μ et ν dans $[A]$.

preuve :

La borne supérieure et la borne inférieure d'un treillis sont des notions duales. L'unicité de la borne supérieure implique l'unicité de la borne inférieure si cette dernière existe. Or $H = \{t \in [A], \quad t \leq \mu \text{ et } t \leq \nu\}$ est non vide car $\perp_{[A]} \leq \mu$ et $\perp_{[A]} \leq \nu$. Par conséquent la borne inférieure existe et est définie par : $\bigvee_{t \in H} t$.

◇

L'espace des descriptions $[A]$ possède une structure de treillis pour les attributs de type symbolique ou taxonomique. Dans les deux cas, $[A]$ est un treillis de longueur finie car chaque domaine est de cardinal fini. Leurs opérateurs d'infimum et de supremum expriment naturellement les notions de généralisation et de spécialisation de deux individus.

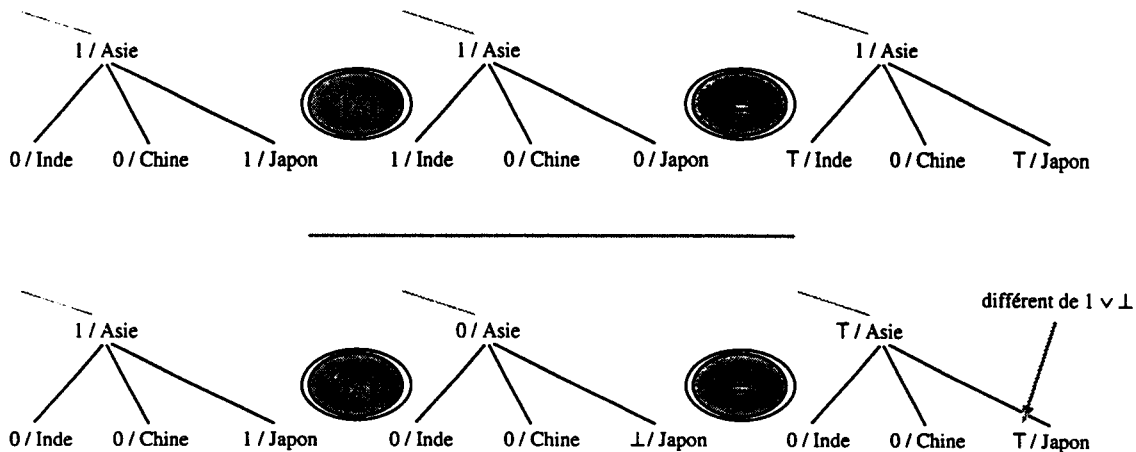


FIG. 4.8 – Exemples de généralisations

Pour les deux types d'attribut, la généralisation de deux individus est calculée simplement à partir de la borne supérieure du treillis de nuance qualifiant les attributs. Dans le cas d'un attribut de type taxonomique, la borne inférieure du treillis V_1 ne suffit pas à définir une procédure de spécialisation. Celle présentée en annexe s'appuie également sur la notion de hiérarchie cohérente, et possède malgré tout une complexité du même ordre que l'opérateur de généralisation.

La modélisation des nuances par des treillis entre dans le cadre de la théorie des sous-ensembles flous [22, 16]. Comme il a été précisé précédemment, les valeurs floues sont des applications du domaine de référence D dans un treillis de nuance V au lieu de $[0, 1]$.

4.4 Les objets structurés

Nous allons montrer dans ce paragraphe comment représenter les objets structurés. Ces derniers sont formés de plusieurs composants qui peuvent être eux-mêmes formés soit d'objets structurés, soit d'attributs de type simple.

4.4.1 Les attributs de type structuré

Un attribut A de type structuré est noté:

$$A = \langle A_1, \dots, A_p \rangle$$

Un tel attribut regroupe un ensemble d'attributs de type simple ou *tuples*. Ainsi, un objet s'écrit:

$$o = (x_1, \dots, x_p)$$

où $x_j \in [A_j]$ pour $j = 1, \dots, p$.

Soit \mathcal{A}_S , l'ensemble des attributs de type simple du modèle et $\mathcal{F} = \{ \prod_i, i \in \mathbb{N} \}$ l'opérateur produit de i ensembles }.

L'ensemble \mathcal{S} de tous les attributs structurés possibles construit à partir de \mathcal{A}_S est le plus petit sous-ensemble vérifiant :

- (B)ase $\forall A \in \mathcal{A}_S, [A] \in \mathcal{S}$
- (I)nduction $\forall (p, \prod_p) \in \mathbb{N} \times \mathcal{F}$, et $\forall [A_1], [A_2], \dots, [A_p] \in \mathcal{S}$ alors,

$$\prod_n ([A_1], \dots, [A_p]) = [A_1] \times \dots \times [A_p] \in \mathcal{S}$$

A chaque attribut $A = \langle A_1, \dots, A_p \rangle$ correspond l'unique espace produit de \mathcal{S} , $[A] = \prod_j [A_j]$ comme ensemble de descriptions possibles.

Définition 4.4.1 *Un attribut de type structuré: $A = \langle A_1, \dots, A_p \rangle$ est caractérisé par son domaine $D = \prod_j [D_j]$, et l'ensemble de ses valeurs nuancées $[A] = \prod_j [A_j]$.*

Extrait du modèle de représentation des Hyalonema, les attributs *Corps* et *Face exhalant* sont définis comme des attributs de type structuré,

Exemple 4.4.1

- *Forme du Corps*:
Domaine: { *subcylindrique, renflée, évasée, aplatie, mal-définie* }
Nuance: V_1
- *Nombre Orifices*:
Domaine: { *unique, quatre, multiples* }
Nuance: V_1
- *Forme Face Exhalant*:
Domaine: { *plane, déprimée, invaginée* }
Nuance: V_1
- *Face Exhalant*:
Structure: { *Forme Face Exhalant, Nombre Orifices* }
- *Corps*:
Structure: { *Forme du Corps, Face Exhalant* }

Comme le montre la figure 4.9, un attribut structuré ou un objet peut être représenté comme une arborescence. Cet arbre est composé de nœuds représentant les objets structurés et de feuilles correspondant aux attributs de type simple.

4.4.2 Structure sur l'ensemble des valeurs

Dans ce paragraphe, nous allons montrer que l'espace des descriptions d'objets possède une structure de treillis, induite par celle des ensembles de descriptions des

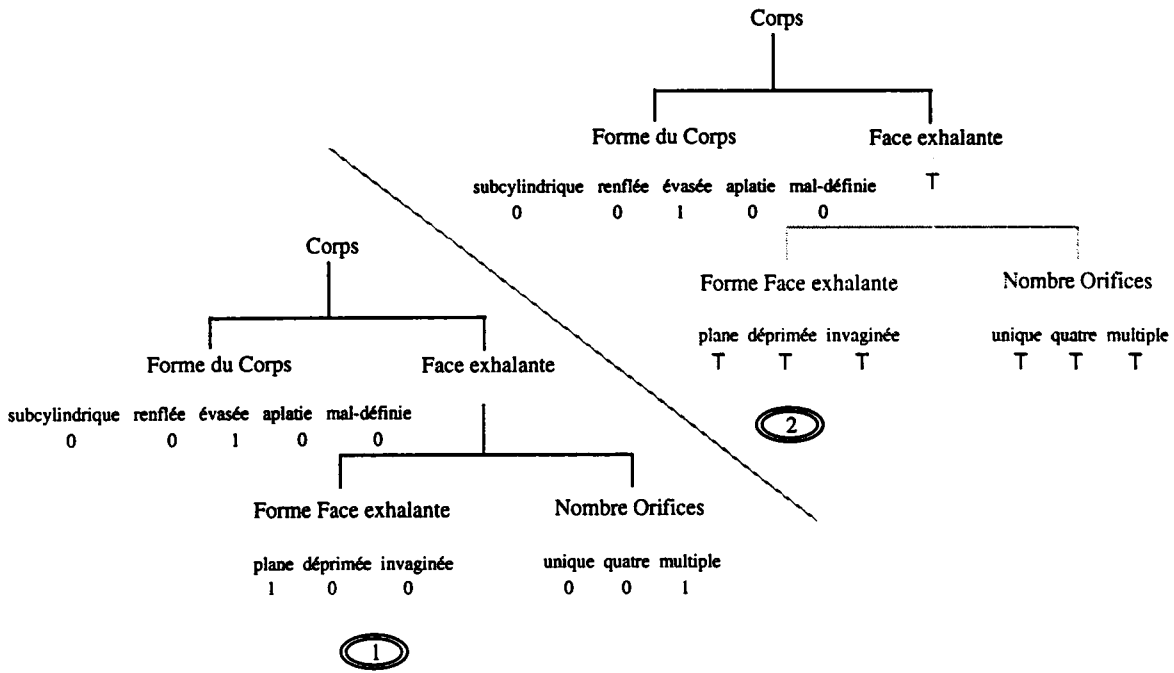


FIG. 4.9 – Instances de l'attribut Corps

attributs de type simple.

Proposition 4.4.1 *L'ensemble des descriptions possibles [A] d'un attribut A possède une structure de treillis de longueur finie.*

preuve :

Soit la proposition P définie pour chaque [A] ∈ S par :

$$P([A]) : [A] \text{ possède une structure de treillis.}$$

- (B)ase $\forall A \in A_S$, par définition des attributs de type simple, [A] possède une structure de treillis,
- (I)nduction Soient $[A_1], [A_2], \dots, [A_p]$ des éléments de S vérifiant la propriété P.
 Comme tout produit fini de treillis de longueurs finies est un treillis de longueur finie, alors $\forall (p, \prod_p) \in \mathbb{N} \times \mathcal{F}, \prod_n (A_1, A_2, \dots, A_p)$ est un treillis de longueur finie.

Par induction, on peut conclure que tout élément de S possède une structure de treillis de longueur finie. Or l'ensemble des descriptions possibles [A] d'un attribut A est un élément de S donc possède une structure de treillis de longueur finie.

◇

La relation d'ordre, les bornes supérieure et inférieure opèrent sur les éléments $x = (x_j)$ et $y = (y_j)$ de $[A]$, de la manière suivante :

- $x < y \iff x_j < y_j \quad \forall j \in \{1, \dots, p\}$
- $x \wedge y = (x_1 \wedge y_1, \dots, x_p \wedge y_p)$
- $x \vee y = (x_1 \vee y_1, \dots, x_p \vee y_p)$

La borne supérieure du treillis $[A]$ est le tuple des plus grands éléments de chaque attribut qui le compose, $\top = (\top, \dots, \top)$. Respectivement, la borne inférieure du treillis $[A]$ est le tuple des plus petits éléments des attributs qui le composent, soit $\perp = (\perp, \dots, \perp)$.

Conséquence, les valeurs inconnue et impossible pour un attribut structuré correspondent à la borne supérieure (\top) et à la borne inférieure (\perp) des descriptions possibles de cet attribut. Par exemple pour le cas 2 de la figure 4.9, la valeur inconnue de l'attribut *face exhalant* se traduit par des valeurs inconnues aux feuilles de ce sous-arbre.

La donnée d'un attribut de type structuré \mathcal{M} , à fortiori de tous les attributs qui le composent, définit un modèle de représentation de données. $[\mathcal{M}]$ représente l'ensemble des descriptions possibles. Par la suite, tous les individus seront décrits sur le même modèle. Le problème d'appariement des valeurs lors de comparaisons ne se pose pas.

4.5 Représentation structurelle des objets

Une description structurelle est une représentation simplifiée de la structure arborescente d'un objet. L'aspect structurel d'une description ne tient compte que de l'*observabilité* de ses différentes propriétés. Ces conditions d'*observabilité* permettent de distinguer trois types d'états à une propriété :

- l'état *I(nconnu)*, indique que les attributs d'un composant A n'ont pas pu être décrits. Cet état généralise toutes les descriptions possibles pour ce composant et correspond à la borne supérieure du treillis $[A]$.
- l'état *A(bsent)*, signifie que le composant n'existe pas ou n'a pas de sens pour l'objet observé. Cet état peut être assimilé à une description vide et correspond à la borne inférieure du treillis des descriptions $[A]$.
- l'état *P(résent)*, souligne que les propriétés du composant ont pu être mesurées.

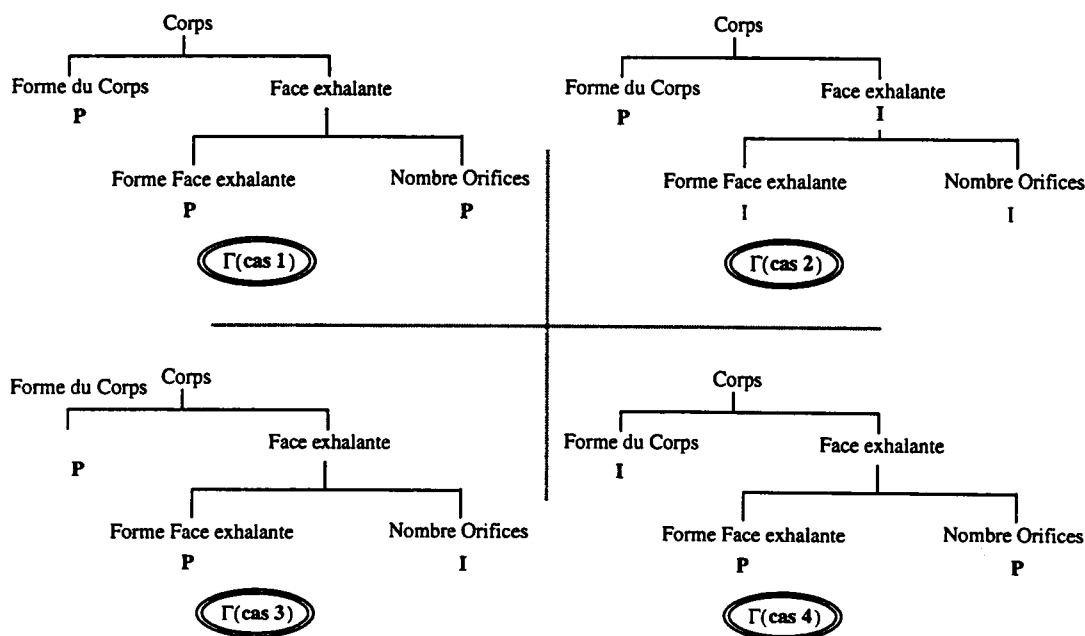


FIG. 4.10 – Exemples de descriptions structurelles

Obtenir une représentation structurelle d'un objet structuré revient à attribuer un de ces trois états à tous les attributs simples de l'objet. Cette transformation est formalisée par la fonction Γ définie ci-dessous.

Définition 4.5.1 La fonction de transformation Γ vérifie :

- Si A est un attribut de type simple alors :

$$\forall x \in [A], \quad \Gamma(x) = \begin{cases} I & \text{si } x = \top \\ A & \text{si } x = \perp \\ P & \text{sinon} \end{cases}$$

- Si $A = \langle A_1, \dots, A_p \rangle$ est un attribut de type structuré alors :

$$\forall x = (x_1, \dots, x_p) \in [A], \quad \Gamma(x) = (\Gamma(x_1), \dots, \Gamma(x_p))$$

L'ensemble des descriptions structurelles $\mathbf{[A]}$ représente l'ensemble des classes d'équivalence de $[A]$, issues de la relation :

$$x \mathcal{R} y \quad \text{ssi} \quad \Gamma(x) = \Gamma(y)$$

d'où $\mathbf{[A]} = [A]/\mathcal{R}$.

Chaque description structurelle représente par définition un groupe de descriptions du modèle.

Les états $\{ A, P, I \}$ peuvent être considérés comme la chaîne de nuance $A \prec P \prec I$ issue de la relation d'ordre définie pour les attributs de type simple, où pour tout x de $[A]$, $\perp < x < \top$.

Dans ce cas, l'espace des structures est un espace d'objets structurés où chaque attribut de type simple est un attribut monovalué nuancé par le treillis des états. Il s'en suit que comme pour les objets, l'espace des descriptions structurelles possède une structure de treillis. Notons que la fonction Γ est un morphisme d'ordre alors qu'elle n'est pas un morphisme de treillis.

Les descriptions structurelles seront utilisées dans le chapitre suivant pour définir une mesure de similarité entre les objets prenant en compte la structure.

Exemple 4.5.1 Les cas suivants sont définis sur l'attribut *corps*. Les cas 1 et 2 sont représentés sur la figure 4.9.

- *cas 1* = ((0 / subcylindrique, 0 / renflée, 1 / évasée, 0 / aplatie, 0 / mal-définie),
 ((1 / plane, 0 / déprimé, 0 / invaginé) , (0 / unique, 0 / quatre, 1 / multiples)))
- *cas 2* = ((0 / subcylindrique, 0 / renflée, 1 / évasée, 0 / aplatie, 0 / mal-définie),
 ((\top / plane, \top / déprimé, \top / invaginé) , (\top / unique, \top / quatre, \top / multiples)))
- *cas 3* = ((0 / subcylindrique, 0 / renflée, 1 / évasée, 0 / aplatie, 0 / mal-définie),
 ((1 / plane, 0 / déprimé, 0 / invaginé) , (\top / unique, \top / quatre, \top / multiples)))
- *cas 4* = ((\top / subcylindrique, \top / renflée, \top / évasée, \top / aplatie, \top / mal-définie),
 ((1 / plane, 0 / déprimé, 0 / invaginé) , (0 / unique, 0 / quatre, 1 / multiples)))

Ces différents cas ont pour descriptions structurelles les formes suivantes, illustrées par la figure 4.10.

$$\begin{array}{ll}
 - \Gamma(\text{cas 1}) = (P, (P, P)) & - \Gamma(\text{cas 3}) = (P, (P, I)) \\
 - \Gamma(\text{cas 2}) = (P, (I, I)) & - \Gamma(\text{cas 4}) = (I, (P, P))
 \end{array}$$

Chapitre 5

Distances et mesures d'agrégation

Sommaire

5.1	Modularité et Distances	50
5.1.1	Les treillis modulaires	50
5.1.2	La distance de <i>Hamming</i>	51
5.1.3	La distance <i>Bi-partite</i>	52
5.1.4	La distance <i>Min-Max</i>	53
5.2	Différentes métriques pour les attributs de type simple . .	54
5.3	Fonctions d'agrégation sur des données structurées	58
5.3.1	Fonction d'agrégation	59
5.3.2	Distance structurelle	60

Dans tous les domaines d'application, où il s'agit d'acquérir de la connaissance, il est nécessaire de savoir comparer des objets entre eux. Les mesures de similarité et les distances sont des outils, permettant de préciser la "ressemblance" ou le "voisinage" entre objets. Ces mesures sont les outils de base de différentes méthodes d'apprentissage telles que le raisonnement par cas.

Dans la pratique, les mesures de similarité sont plus fréquemment utilisées car elles ne vérifient pas l'inégalité triangulaire et sont donc moins contraignantes. Une similarité peut toujours être définie à partir d'une distance.

Définition 5.0.2 L'application s de $T \times T$ dans $[0, 1]$ vérifiant,

- $\forall x, y \in T, s(x, y) = 1$ ssi $x = y$ *séparabilité*
- $\forall x, y \in T, s(x, y) = s(y, x)$ *symétrie*

est une mesure de similarité sur \mathcal{T} .

Ce chapitre est consacré principalement à l'étude de similarité et de distance sur le modèle des objets structurés. Dans la théorie des treillis, les structures dites "modulaires" possèdent des propriétés utiles à la définition de distance.

Après un bref rappel sur les treillis modulaires, nous proposerons des distances sur ce type. Ces distances vont permettre ensuite, d'établir des mesures de similarité concernant les attributs de type simple. Ensuite à partir de ces différentes mesures et fonctions d'agrégation, nous développerons des mesures dites "globales" sur les attributs structurés, qui prennent en compte la structure des objets.

Pour définir nos mesures, nous nous plaçons dans l'hypothèse que la structure et plus précisément que la position d'un attribut par rapport à la racine donne une idée de la contribution de cet attribut pour comparer des objets. Les mesures "globales" que nous proposons, ont la particularité de diminuer l'influence des attributs les plus profonds dans l'arborescence du modèle.

5.1 Modularité et Distances

5.1.1 Les treillis modulaires

Dans un treillis modulaire, tous les chemins reliant deux éléments sont de même longueur. Lorsque le treillis est de longueur finie, il est possible de définir la fonction hauteur h correspondant à une graduation du treillis. La hauteur d'un élément mesure la longueur du chemin séparant l'élément à la borne inférieure du treillis.

Définition 5.1.1 Une graduation ou fonction de rang v d'un treillis \mathcal{T} , est une application de \mathcal{T} dans l'ensemble des entiers relatifs vérifiant :

- $x < y \implies v(x) < v(y)$
- $x \prec y \implies v(y) = v(x) + 1$

Définition 5.1.2 Un treillis \mathcal{T} est modulaire si et seulement si le treillis est gradué et sa fonction de rang v vérifie :

$$\forall x, y \in \mathcal{T}, \quad v(x \wedge y) + v(x \vee y) = v(x) + v(y)$$

Plusieurs définitions équivalentes de treillis modulaires existent, en particulier :

Un treillis est modulaire si et seulement si aucun de ses sous treillis est isomorphe au plus petit sous treillis non modulaire appelé \mathcal{N}_5 (fig. 5.2)

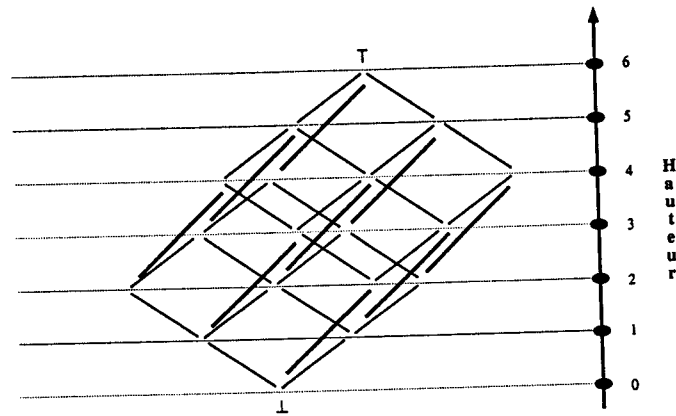


FIG. 5.1 - Exemples de treillis modulaires

Cette définition permet de vérifier rapidement la condition de modularité ou de "non-modularité" d'un treillis.

Proposition 5.1.1 Soit \mathcal{T} un treillis modulaire, et une graduation. La fonction d définie par :

$$d(x, y) = v(x \vee y) - v(x \wedge y) = \mathcal{L}([x \wedge y, x \vee y])$$

est une distance sur \mathcal{T} .

$\mathcal{L}([x, y])$ dénote la longueur de l'intervalle $[x, y]$.

La structure modulaire est la plus "petite" structure permettant d'adapter les distances les plus classiques de la théorie des ensembles et de la théorie des sous-ensembles flous, telle que la distance de Hamming. Pour la suite, les treillis sont supposés modulaires de longueur finie, strictement supérieure à 1.

5.1.2 La distance de Hamming

La fonction d_H dénote la distance usuelle pour un treillis modulaire de longueur finie \mathcal{T} , où :

$$d_H(x, y) = h(x \vee y) - h(x \wedge y) \text{ avec } x, y \in \mathcal{T}$$

Comme \mathcal{T} est de longueur finie, cette distance peut être normalisée à valeurs dans $[0, 1]$ par la hauteur de élément maximal $h(\mathcal{T})$, c'est à dire la longueur de l'intervalle $[\perp, \top]$.

$$D_H(x, y) = (h(x \vee y) - h(x \wedge y)) / h(\mathcal{T}) \text{ avec } x, y \in \mathcal{T}$$

D_H est la généralisation de la distance de Hamming généralisée relative des ensembles ordinaires ou flous [22], où $d(A, B) = |A \cup B| - |A \cap B|$. La preuve de cette distance dans le cas de treillis modulaire se trouve dans la référence [3].

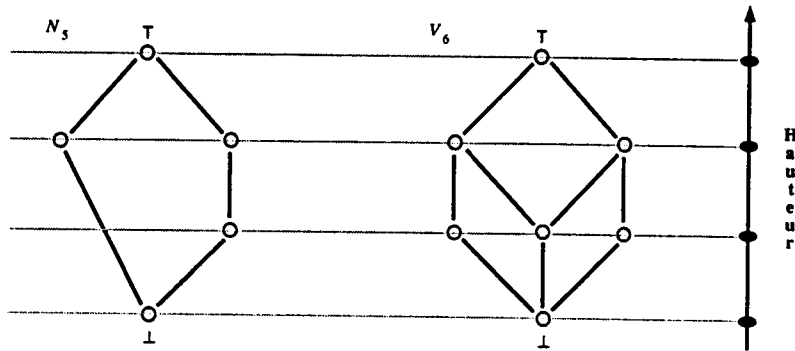


FIG. 5.2 - Treillis non modulaires

5.1.3 La distance *Bi-partite*

Comme pour la distance précédente, la distance bi-partite d_b des treillis modulaires s'identifie à celle définie sur les ensembles ordinaires et flous [21] où $d(A, B) = 1 - |A \cap B| / |A \cup B|$.

Proposition 5.1.2 La fonction d_b de $\mathcal{T} \times \mathcal{T}$ dans $[0, 1]$ définie par :

$$\forall x, y \in \mathcal{T}, \quad d_b(x, y) = \begin{cases} 0 & \text{si } (x, y) = (\perp, \perp) \\ 1 - \frac{h(x \wedge y)}{h(x \vee y)} & \text{sinon} \end{cases}$$

est une distance sur \mathcal{T} .

Montrer que la fonction d_b est à valeur dans \mathbb{R}^+ , et qu'elle vérifie les propriétés de séparabilité et de symétrie d'une distance ne présente aucune difficulté. Le lemme suivant, est utile à la démonstration de l'inégalité triangulaire.

Lemme 1

$$\forall x, y, z \in \mathcal{T}, \quad \begin{cases} d_H(z, y) + h(z \wedge y) \geq h(x \wedge z) \\ d_H(z, x) + h(z \wedge y) \geq h(x \wedge z) \end{cases}$$

preuve: Soient $x, y, z \in \mathcal{T}$ et $A = d(z, y) + h(x \wedge y) - h(x \wedge z)$,

$A = h(z \vee y) - h(z \wedge y) + h(x \wedge y) - h(x \wedge z)$ par définition de d_H ,

h étant une valuation $h(z \wedge y) + h(x \wedge z) = h((x \wedge z) \vee (y \wedge z)) + h(x \wedge y \wedge z)$ donc,

$A = h(z \vee y) - h((x \wedge z) \vee (y \wedge z)) + h(x \wedge y) - h(x \wedge y \wedge z)$,

par isotonie de h ,

$h(z \vee y) \geq h((x \wedge z) \vee (y \wedge z))$ car $z \vee y \geq (x \wedge z) \vee (y \wedge z)$ et,

$h(x \wedge y) \geq h(x \wedge y \wedge z)$ car $x \wedge y \geq x \wedge y \wedge z$,

donc $A \geq 0$.

La propriété $d_H(z, x) + h(z \wedge y) \geq h(x \wedge z)$ se démontre de manière identique.

◇

preuve : de l'inégalité triangulaire. Soit $x, y, z \in T$,

Pour $(x, y) = (\perp, \perp)$, cette inégalité est trivialement vérifiée,

Pour $(x, y) \neq (\perp, \perp)$ alors par définition,

$$d_b(x, y) = \frac{d_H(x, y)}{(d_H(x, y) + h(x \wedge y))}$$

Pour simplifier l'écriture, la distance de Hamming d_H est simplement noté d .

La fonction f de \mathbb{R} dans \mathbb{R} définie par $f(t) = t/(t + h(x \wedge y))$ est une fonction croissante en t car la fonction hauteur est positive c'est à dire $h(x \wedge y) \geq 0$.

d étant une distance, $d(x, y) \leq d(x, z) + d(z, y)$ (inégalité triangulaire) et par croissance de la fonction f on a,

$f(d(x, z) + d(z, y)) \geq f(d(x, y))$ c'est à dire ,

$$(1) \frac{d(x, y)}{d(x, y) + h(x \wedge y)} \leq \frac{d(x, z)}{d(x, z) + d(z, y) + h(x \wedge y)} + \frac{d(z, y)}{d(x, z) + d(z, y) + h(x \wedge y)}$$

d'après le lemme 1, les inégalités suivantes sont vérifiées,

$$\frac{d(x, z)}{d(x, z) + d(z, y) + h(z \wedge y)} \leq \underbrace{\frac{d(x, z)}{d(x, z) + h(x \wedge z)}}_{d_b(x, z)}$$

$$\text{et } \frac{d(z, y)}{d(x, z) + d(z, y) + h(z \wedge y)} \leq \underbrace{\frac{d(z, y)}{d(z, y) + h(z \wedge y)}}_{d_b(z, y)}$$

d'où en remplaçant dans l'inéquation (1), $d_b(x, y) \leq d_b(x, z) + d_b(z, y)$.

Pour tout x, y, z de T , d_b vérifie l'inégalité triangulaire; Il en résulte que d_b est une distance sur T .

◇

5.1.4 La distance *Min-Max*

La distance *Min-Max* mesure la plus grande longueur de x ou y à leur supremum $x \vee y$.

Proposition 5.1.3 La fonction d_m de $T \times T$ dans $[0, 1]$ définie par :

$$\forall x, y \in T, \quad d_m(x, y) = (h(x \vee y) - \min(h(x), h(y)))/h(T)$$

est une distance sur T .

La distance duale mesurant la plus grande longueur entre x ou y et leur infimum $x \wedge y$, c'est à dire $d(x, y) = (\max(h(x), h(y)) - h(x \wedge y))/h(\top)$, est identique à d_m . En effet, h est une valuation et, $h(x \vee y) - \min(h(x), h(y)) = \max(h(x), h(y)) - h(x \wedge y)$. Comme pour la distance précédente, seule l'inégalité triangulaire est démontrée.

preuve : Soit $x, y, z \in \mathcal{T}$,

Posons $A = h(\top) \cdot (d_m(x, z) + d_m(z, y) - d_m(x, y))$, c'est à dire par définition,

$$A = h(x \vee z) - \min(h(x), h(z)) + h(y \vee z) - \min(h(y), h(z)) - h(x \vee y) + \min(h(x), h(y))$$

comme h est une valuation,

$$A = 2h(z) + h(x \wedge y) + \min(h(x), h(y)) - h(x \wedge z) - h(y \wedge z) - \min(h(x), h(z)) - \min(h(y), h(z)),$$

$$\text{et } h(x \wedge z) + h(y \wedge z) = h((x \wedge z) \vee (y \wedge z)) + h(x \wedge y \wedge z),$$

$$\text{d'où } A = 2h(z) + h(x \wedge y) + \min(h(x), h(y)) - h((x \wedge z) \vee (y \wedge z)) - h(x \wedge y \wedge z) - \min(h(x), h(z)) - \min(h(y), h(z)) \text{ or,}$$

- $(x \wedge z) \vee (y \wedge z) \leq z$ et $x \wedge y \wedge z \leq x \wedge y$ et par isotonie $h(z) \geq h((x \wedge z) \vee (y \wedge z))$ et $h(x \wedge y) \geq h(x \wedge y \wedge z)$

- $h(z) + \min(h(x), h(y)) \geq \min(h(x), h(z)) + \min(h(y), h(z))$ car

$$h(z) \geq \min(h(y), h(z)) \text{ ou } \min(h(x), h(z)) \text{ et,}$$

$$\text{si } h(x) \leq h(z) \text{ alors } \min(h(x), h(y)) \geq \min(h(x), h(z))$$

$$\text{ou si } h(y) \leq h(z) \text{ alors } \min(h(x), h(y)) \geq \min(h(y), h(z))$$

Il en résulte que $A \geq 0$ et comme \mathcal{T} est un treillis de longueur supérieure à 1 et $h(\top) > 0$, l'inégalité triangulaire est vérifiée.

En admettant les propriétés de *séparabilité*, de *symétrie*, de *non négativité* qui sont faciles à démontrer, d_m est une distance sur \mathcal{T} .

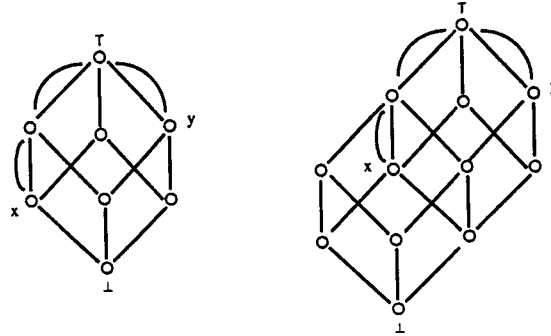
◇

5.2 Différentes métriques pour les attributs de type simple

Dans le modèle des objets structurés, les attributs de type simple $(A, D, V, [A])$ sont constitués d'attributs symboliques et taxonomiques. Dans les deux cas, l'ensemble des descriptions possibles possède une structure de treillis compatible avec la relation d'ordre définie sur les treillis de nuances. Lorsque V est un treillis modulaire, l'ensemble des descriptions possibles d'un attribut symbolique $[A] = V^D$ est modulaire, et dans ce cas l'utilisation des distances définies ci dessus est possible. Par contre dans le cas de taxonomie, $[A]$ n'est pas un sous treillis de V^D et $[A]$ ne vérifie pas nécessairement l'axiome de modularité. Pour définir des mesures *locales* sur les attributs de type simple, il est nécessaire de différencier *symbolique* et *taxonomique*.

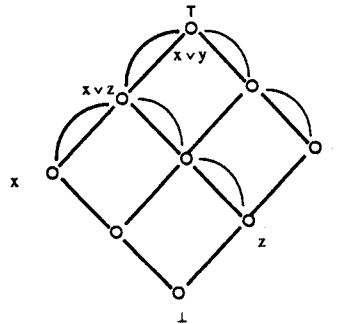
* *Interprétation des Distances sur les Attributs Symboliques*

- La distance de *Hamming* évalue la différence entre deux éléments comme le nombre d'arcs les séparant. D'une certaine manière, cette distance mesure le nombre d'étapes nécessaires pour passer d'un élément à l'autre. Cette valeur correspond également à l'écart entre la borne supérieure et la borne inférieure des éléments comparés.



Sur cette figure la distance entre x et y pour les deux treillis est identique et vaut $d_H(x, y) = 3$. Cependant, lorsque la distance de Hamming est normalisée, les distances ne sont plus identiques dans les deux cas. La distance varie en fonction de la taille du treillis et donc en fonction de la taille du domaine de l'attribut.

- La distance *Min-Max*, évalue la différence entre deux éléments comme l'écart maximum entre chacun des deux éléments et leur borne supérieure. Comme pour la distance de Hamming, la distance Min-Max généralisée, varie en fonction de la taille du domaine de l'attribut.



Sur cet exemple y et z ont une distance identique égale à $d_M(x, y) = \frac{2}{4}$. Appliquée aux valeurs x et y , la distance de Hamming normalisée donne la valeur maximale 1.

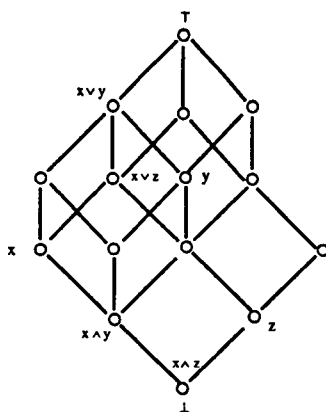
- La distance *Bi-partite*, apporte une information de nature conceptuelle. En effet, comme le montre la proposition ci-dessous, la distance est maximale et vaut 1, dès que les éléments comparés ont comme borne inférieure celle du treillis (\perp). En d'autres termes, deux éléments sont totalement séparés s'ils ne possèdent aucun

élément commun, au sens de la relation d'ordre.

Proposition 5.2.1 $\forall x \in [A], \forall y \in [A], \text{ tel que } (x, y) \neq (\perp, \perp)$

$$x \wedge y = \perp \iff d(x, y) = 1$$

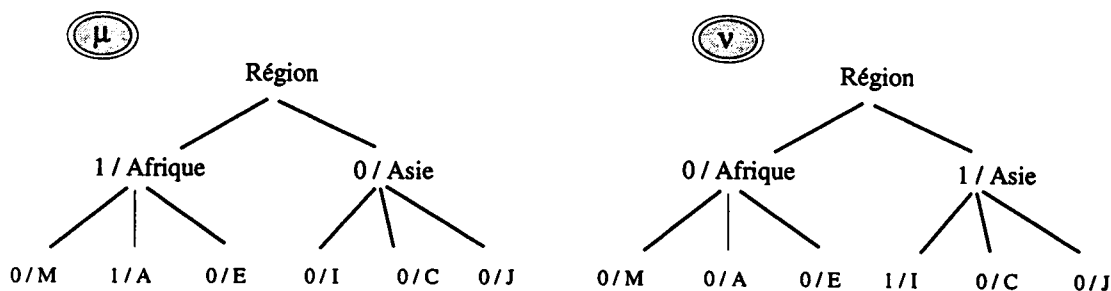
A partir de la définition de la distance, la preuve de cette proposition est immédiate. Sur les exemples de la figure ci-dessous, la distance de x à z est égale à la valeur maximale 1, alors que celle entre x et y est de $\frac{3}{4}$.



Par rapport aux deux autres distances, la normalisation est effectuée en fonction de la hauteur de la généralisation des éléments comparés, et non en fonction de la hauteur maximale du treillis. La conséquence est la stabilité de la distance relativement à la taille des domaines. Contrairement à la distance de Hamming qui mesure d'une certaine manière le nombre d'éléments qui ne sont pas communs, la distance de bi-partite exprime le pourcentage de ces éléments. La similarité induite par la fonction $f(x) = 1 - x$ sur cette distance, est connue pour les descripteurs booléens sous le nom d'indice de *Jaccard*.

* Cas des Attributs Taxonomiques

Les taxonomies sont nuancées dans notre modèle par le treillis de croyance V_1 (fig. 4.1). V_1 est un treillis modulaire et peut donc disposer d'une métrique noté δ . Les distances de *Hamming*, *Min-Max* ou *Bi-partite* donnent toutes des résultats différents sur V_1 .



La distance entre deux valeurs hiérarchiques nuancées peut être définie comme la moyenne des distances évaluées pour chaque valeur nuancée du domaine.

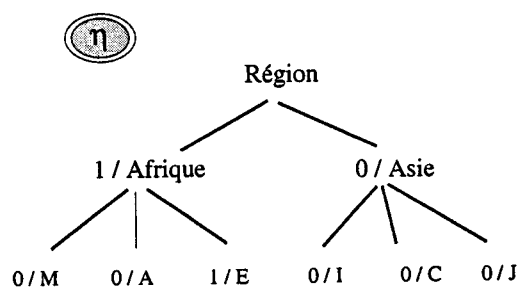
$$\forall \mu, \nu \in [A], \quad d(\mu, \nu) = \frac{1}{|D|} \sum_{h \in D} \delta(\mu(h), \nu(h))$$

Ainsi pour l'exemple ci-dessus les distances obtenues sont,

- $d(\mu, \nu) = \frac{4}{9}$ où δ représente la distance *Hamming*, ou *Bi-partite*,
- $d(\mu, \nu) = \frac{2}{9}$ où δ représente la distance *Min-Max*

L'utilisation de la distance *Min-Max* n'est pas appropriée pour les hiérarchies nuancées. En effet, il est souhaitable qu'une valeur qualifiée par la nuance *vraie* soit totalement différente d'une valeur qualifiée par la nuance *fausse*. Les distances de *Hamming* et *Bi-partite* ne diffèrent que lorsque la nuance \perp est utilisée.

Remarque : L'intervention des points non terminaux de la hiérarchie dans l'évaluation de la distance, telles que les régions *Asie* et *Afrique*, permet de prendre en compte la structure hiérarchique.



En effet, comme la région *Asie* est commune pour μ et η , elle n'intervient pas dans le calcul de leur différence ($d(\mu, \eta) = \frac{2}{9}$). Par contre entre les cas ν et η , la différence de valeurs pour les régions *Asie* et *Afrique* accentue leurs différences ($d(\nu, \eta) = \frac{4}{9}$).

L'utilisation de la distance euclidienne ou des distances généralisées de Minkowski, permettent d'accentuer les différences maximales ou de diminuer l'influence de la largeur du domaine

$$\forall \mu, \nu \in [A], \quad d(\mu, \nu) = \frac{\sqrt{\sum_{h \in D} \delta^2(\mu(h), \nu(h))}}{\sqrt{|D|}}$$

Dans ce cas, la différence entre μ et η est $d(\mu, \eta) = \frac{\sqrt{2}}{3} \simeq 0.47$ au lieu de 0.22.

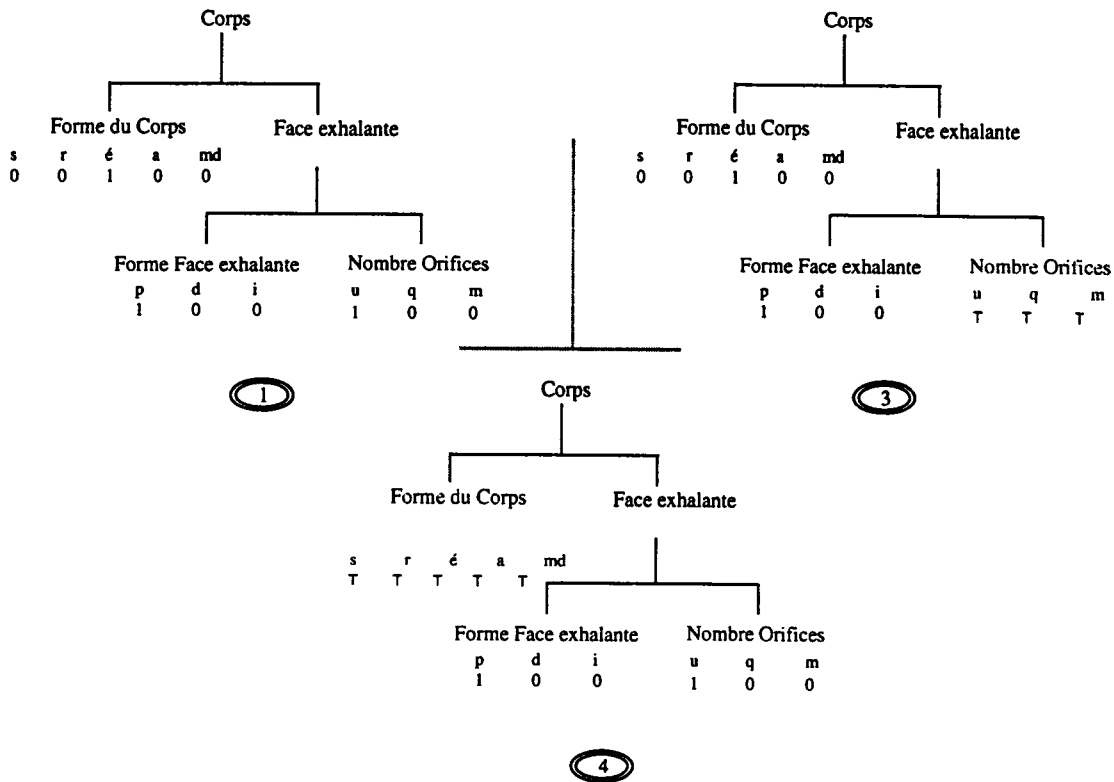


FIG. 5.3 - Les différents cas

5.3 Fonctions d'agrégation sur des données structurées

Le problème essentiel à résoudre dans la conception d'une distance pour des données composites, est la prise en compte des différentes similitudes évaluées sur chaque composant. Classiquement ces mesures dites *globales* sont élaborées à partir de fonctions d'agrégation sur \mathbb{R}^p telles que la moyenne pondérée, les distances euclidiennes, etc ...

Ces distances ne sont pas adaptées au modèle des objets structurés car les relations de dépendance entre attributs n'interviennent pas dans le calcul des similarités. Sauf avis contraire de l'expert, chaque composant d'un attribut structuré intervient de la même façon dans le calcul de la similarité. Mais chaque composant peut être constitué de composants. Ces derniers ne sont pas au même niveau de description que leur parent et leur importance dans le calcul de la similarité doit diminuer. Pour l'exemple 4.5.1 du chapitre précédent représenté par la figure 5.3, comme la différence entre le *cas 1* et le *cas 3* ne porte que sur un sous-attribut de l'attribut *Corps*, alors que celle entre le *cas 1* et le *cas 4* porte sur un attribut direct du *Corps*, la distance entre le *cas 1* et le *cas 3* devrait être moins prononcée que celle entre le *cas 1* et le *cas 4*.

Deux méthodes sont utilisées pour évaluer la similitude d'objet. La première consiste à prendre en compte simultanément la structure et les différentes similarités locales. La deuxième consiste à évaluer d'une part la similarité structurelle et d'autre part de l'agréger avec une similarité moyenne des valeurs.

5.3.1 Fonction d'agrégation

Proposition 5.3.1 *L'application D_S définie par :*

- Si A est un attribut de type simple,

$$\forall (x, y) \in [A] \times [A], \quad D_S(x, y) = \delta_A(x, y)$$

où δ_A est une distance sur l'attribut de type simple A .

- Si $A = \langle A_1, \dots, A_p \rangle$ est un attribut de type tuple,

$$\forall x = (x_1, \dots, x_p) \in [A], \quad \forall y = (y_1, \dots, y_p) \in [A],$$

$$D_S(x, y) = \frac{\sum_i \rho(A_i) \cdot D_S(x_i, y_i)}{\sum_i \rho(A_i)}$$

est une distance sur $[A]$. D_S est la distance pondérée par la structure.

preuve : L'application D_S est définie implicitement de manière inductive sur l'ensemble des structures possibles \mathcal{S} (conf. 7.1).

Soit la proposition P définie pour chaque treillis $[A]$ de \mathcal{S} par :

$P([A])$: L'application D_S de $[A] \times [A]$ dans $[0, 1]$ définie ci-dessus, est une distance.

- (B)ase Pour chaque attribut A de A_S , l'application D_S est la distance δ_A définie pour l'attribut de type simple A ,
- (I)nduction Soient $[A_1], \dots, [A_p]$ des éléments de \mathcal{S} vérifiant la propriété P .
Comme toute combinaison linéaire de distance est une distance,

$$\frac{\sum_i \rho(A_i) \cdot D_S(x_i, y_i)}{\sum_i \rho(A_i)} \quad \text{est une distance}$$

Il est démontré par induction que l'application D_S est une distance pour tout élément de \mathcal{S} . En particulier pour l'ensemble des descriptions possibles $[A] \in \mathcal{S}$ de l'attribut A .

◇

Illustrée par la figure ci-dessus, la distance D_S correspond à une moyenne pondérée des distances évaluées sur chaque composant de type simple, où les nouvelles pondérations

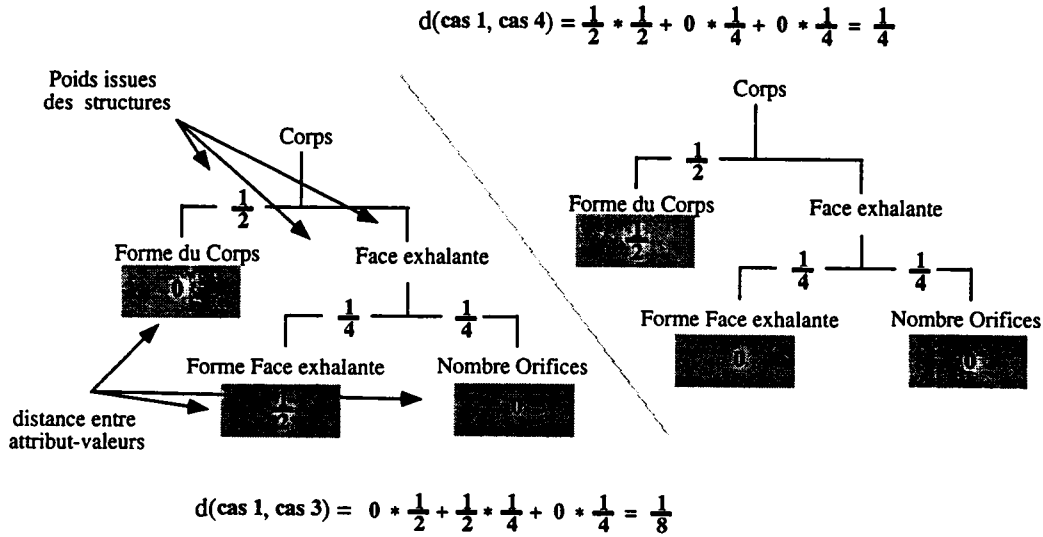


FIG. 5.4 - Répartition des poids en fonction de la hauteur

sont calculées de manière récursive en fonction de la structure, où :

Le poids de l'attribut racine vaut 1 et ,

Si $A = \langle A_1, \dots, A_p \rangle, \forall k \in \{ 1, \dots, p \} \quad \rho(A_k) = \frac{\rho(A)}{p}$.

La structure a pour effet de diminuer l'influence des attributs les plus *profonds* dans l'arbre de description.

Ce qui se traduit sur l'exemple 4.5.1 par $D_S(\text{cas1, cas3}) = \frac{1}{8} < D_S(\text{cas1, cas3}) = \frac{1}{4}$ avec comme mesure pour les attributs de type simple, la distance *Bi-partite*.

5.3.2 Distance structurelle

La distance dite "structurelle" a pour objectif de comparer uniquement les objets au niveau de leur structure. L'ensemble des descriptions structurelles $\mathbf{[A]}$ dispose donc d'une métrique où la part de la structure dans la distance est calculée sur le même principe que celle des objets, et correspond à la proposition suivante :

Proposition 5.3.2 L'application ∇ définie sur $\mathbf{[A]}$ par :

- Si A est un attribut de type simple,

$$\forall (x, y) \in \mathbf{[A]} \times \mathbf{[A]}, \quad \nabla(x, y) = \delta(x, y)$$

δ est la distance entre les éléments $\{A, I, P\}$.

- Si $A = \langle A_1, \dots, A_p \rangle$ est un attribut de type tuple,

$$\forall x = (x_i) \in \prod_i |A_i| \text{ et } \forall y = (y_i) \in \prod_i |A_i|,$$

$$\nabla(x, y) = \frac{\sum_i \rho(A_i) \cdot \nabla(x_i, y_i)}{\sum_i \rho(A_i)}$$

est une distance.

Les feuilles d'une description structurée, correspondant aux attributs de type simple, sont des éléments de $\{A, I, P\}$. Pour compléter la distance ∇ , la table d'indices de proximité suivante permet de définir la fonction de comparaison δ sur les valeurs $\{A, I, P\}$.

δ	I	P	A
I	0	1/2	1
P	1/2	0	1
A	1	1	0

En énumérant tous les cas, on montre que δ sur $\{A, P, I\}$ vérifie les propriétés d'une distance. Les différentes valeurs d'indices de proximité ont été établies en fonction des particularités de la distance *Bi-partite*, c'est à dire :

- Par définition, $\delta(A, P) = \delta(\Gamma(\perp), \Gamma(x))$.

D'après la propriété 5.2.1, si $x \in [A] \setminus \{\top, \perp\}$ alors $d(\perp, x) = 1$. Par analogie avec les descriptions réelles, la différence entre les valeurs A et P est fixée à 1.

- De la même manière, $\delta(P, I) = \delta(\Gamma(*), \Gamma(x))$.

Cependant la différence entre la borne supérieure et une description quelconque varie dans $]0, 1[$ quelque soit la distance utilisée. La différence entre P et I est fixée arbitrairement à $\frac{1}{2}$, estimant la distance moyenne entre \top et une valeur quelconque x , soit $\delta(P, I) = \frac{1}{2}$

Une mesure globale sur les objets structurés résulte de l'agrégation de la distance calculée pour la structure et de celle calculée sur les valeurs considérées comme des données vectorielles. Différentes fonctions permettent d'agréger ces deux valeurs. Cependant les fonctions recherchées doivent définir une mesure globale à valeur dans $[0, 1]$, comme pour toutes les mesures présentées dans ce chapitre. Cette particularité est nécessaire pour définir, par la suite, des fonctions de reconnaissance. De plus la prise en compte de la structure ne doit pas occulter les différences entre les valeurs des propriétés des objets.

La fonction d'agrégation choisie f est définie par :

$$\forall u, v \in [0, 1], \quad f(u, v) = 1 + a uv - u - a v = (1 - u) (1 - av), \quad \text{où } 0 \leq a \leq 1$$

On note D une distance à valeur dans $[0, 1]$, qui sera dans notre cas la moyenne pondérée éventuellement des différences évaluées sur les attributs de type simple des objets.

Proposition 5.3.3 *L'application S de $[A] \times [A]$ dans $[0, 1]$ vérifiant,*

$$S(x, y) = f(D(x, y), \nabla(\Gamma(x), \Gamma(y)))$$

est une mesure de similarité. On appellera S similarité structurelle.

preuve : Soit $x, y \in [A] \times [A]$,

- $S(x, y)$ est à valeur dans $[0, 1]$ comme produit de valeurs de $[0, 1]$.
- $S(x, y) = 1 \iff (1 - D(x, y))(1 - a\nabla(\Gamma(x), \Gamma(y))) = 1$ par définition de f ,
Comme $(1 - a\nabla(\Gamma(x), \Gamma(y))) \in [0, 1]$ et $(1 - D(x, y)) \in [0, 1]$,
 $S(x, y) = 1 \iff \nabla(\Gamma(x), \Gamma(y)) = 0$ et $D(x, y) = 0$,
Comme D est une distance sur $[A] \times [A]$ $D(x, y) = 0 \iff x = y$,
à fortiori, si $x = y$ alors $\Gamma(x) = \Gamma(y)$ et $\nabla(\Gamma(x), \Gamma(y)) = 0$.

Par conséquent, $S(x, y) = 1 \iff x = y$ et S vérifie la propriété de séparabilité.

- $S(x, y)$ est une fonction symétrique car D et ∇ sont des fonctions symétriques

La fonction S est une mesure de similarité dans $[0, 1]$.

◇

Le paramètre a de la fonction f permet d'influencer le rôle de la similarité structurelle dans l'évaluation de la mesure globale et donne un sens à cette similarité. En effet, comme $a \leq 1$ nous avons montré que la similarité entre deux éléments x et y est maximale si et seulement si ces éléments sont égaux au niveau des valeurs, c'est à dire $x = y$.

Nous verrons dans la suite que pour définir des fonctions de reconnaissance cohérentes il est nécessaire de considérer a strictement inférieur à 1.

Dans l'exemple qui suit, les similarités seront calculées d'après les cas de l'exemple 4.5.1 avec la valeur $a = \frac{1}{2}$ et en fonction de la distance *Bi-partite* pour D .

- *Similarité vectorielle entre cas,*

$$D(\text{cas 1}, \text{cas 3}) = \frac{1}{3} \cdot (0 + 0 + \frac{1}{2}) = \frac{1}{6}$$

$$D(\text{cas 1}, \text{cas 4}) = \frac{1}{3} \cdot (\frac{1}{2} + 0 + 0) = \frac{1}{6}$$

- *Similarité structurelle entre cas,*

$$\nabla(\Gamma(\text{cas 1}), \Gamma(\text{cas 3})) = \nabla((P, (P, P)), (P, (P, I))) = \frac{1}{2} \cdot 0 + \frac{1}{2} \left(\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \frac{1}{2} \right) = \frac{1}{8}$$

$$\nabla(\Gamma(\text{cas 1}), \Gamma(\text{cas 4})) = \nabla((P, (P, P)), (I, (P, P))) = \frac{1}{2} \cdot 0 + \frac{1}{2} \left(\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 0 \right) = \frac{1}{4}$$

- *Similarité globale entre cas,*

$$d(\text{cas 1}, \text{cas 3}) = \left(1 - a \frac{1}{8}\right) \left(1 - \frac{1}{6}\right) = \frac{25}{32} \simeq 0.78$$

$$d(\text{cas 1}, \text{cas 4}) = \left(1 - a \frac{1}{4}\right) \left(1 - \frac{1}{6}\right) = \frac{5}{8} = 0.4$$

Comme il était présenté en introduction, l'influence de la structure s'exprime par une similarité plus faible entre le cas 1 et le cas 3 par rapport aux cas 1 et cas 4.

L'espace des descriptions possibles du modèle des objets structurés peut être muni de différentes métriques et similarités. Les mesures développées dans ce chapitre permettent à l'utilisateur de choisir la manière de comparer les objets structurés en fonctions de ses attentes. L'expérimentation de la partie application montre l'influence du choix de la similarité dans la qualité des résultats.

Chapitre 6

La représentation des concepts

Sommaire

6.1	La caractérisation des observations	65
6.1.1	“L'appartenance” à une description	65
6.1.2	Fonctions caractéristiques floues	67
6.2	Généralités sur les concepts imprécis	70

Pour apprendre à partir d'un ensemble d'observations, il était nécessaire de disposer d'abord d'un modèle de représentation de données imprécises et structurées et de métriques permettant de les comparer. Ce fut l'objet des chapitres précédents. Dans ce chapitre, nous définirons de manière précise la notion de *concept* imprécis.

6.1 La caractérisation des observations

Soit \mathcal{O} un ensemble d'observations décrites sur un modèle de représentation de données \mathcal{M} . L'ensemble des descriptions $[M]$ (voir chap. 4) représente l'ensemble des observations possibles décrites à partir des attributs du modèle \mathcal{M} . Cependant toutes les descriptions ne correspondent pas à des observations du monde réel. Etant donnée une description $d \in [M]$, comment mesurer le recouvrement d'une observation par d ?

6.1.1 “L'appartenance” à une description

Dans différents systèmes d'apprentissage tel que [40], une observation o est dite *appartenir* à une description d , lorsque o possède toutes les caractéristiques ou encore des caractéristiques plus spécifiques que celle de d . On dit encore que o *satisfait* d .

Exemple 6.1.1 Soient les attributs couleur de domaine { bleu, rouge, vert } et taille de domaine { petit, moyen, fort }.

L'observation $o = [x_1 = \text{bleu}] [x_2 = \text{petit}]$ appartient à la description $d = [\text{couleur} = \text{bleu} \vee \text{rouge}] [\text{taille} = \text{petit} \vee \text{moyen}]$, alors que $o' = [x_1 = \text{vert}] [x_2 = \text{petit}]$ ne satisfait pas la caractéristique **couleur** de d .

Ce qui revient à dire que, l'appartenance d'une observation à une description d est mesurée à partir d'une fonction caractéristique ou de reconnaissance χ_d définie sur l'ensemble des descriptions, où :

$$\chi_d : [\mathcal{M}] \longrightarrow \{0, 1\}$$

$$u \longmapsto \begin{cases} 1 & \text{si } u \text{ satisfait toutes les propriétés de } d \\ 0 & \text{sinon} \end{cases}$$

L'ensemble des observations caractérisé par la description d est l'image par χ_d de \mathcal{O} .

Lorsque $[M]$ est muni d'une relation d'ordre \leq : "est plus spécifique que", une observation o est recouverte par une description d signifie que o est moins général que d . Dans les approches comme le raisonnement par cas, l'espace $[M]$ est muni d'une métrique δ . Le recouvrement d'une observation par une description est déterminée à partir de cette métrique, au sens où : l'observation o est reconnue par la description d si o est proche de d . Deux éléments sont dits proches lorsque leur similarité dépasse un certain seuil fixé a priori.

Exprimée de cette manière, la représentativité d'une description est très limitatif. En effet, une observation o appartient ou non à la représentation d'une description. Or dans certaines situations par exemple pour les problèmes de diagnostic, une observation peut être proche d'une description ou de plusieurs descriptions, sans satisfaire réellement toutes les propriétés de ces descriptions. Lorsqu'une observation satisfait "plus ou moins" les propriétés d'une description, il serait intéressant d'exprimer son appartenance *graduelle* à la représentation de la description.

En résumé, l'approche classique revient à définir une fonction de reconnaissance χ qui associe à chaque description d une fonction caractéristique χ_d à valeur dans $\{0, 1\}$, c'est à dire :

$$\chi : [\mathcal{M}] \longrightarrow \{0, 1\}^{[\mathcal{M}]}$$

$$d \longmapsto \chi_d$$

Pour pouvoir définir une appartenance graduelle, les fonctions caractéristiques devront être floues. Dans ce qui suit, la fonction χ associera à chaque description d une fonction caractéristique χ_d à valeur dans $[0, 1]$, c'est à dire :

$$\chi : [\mathcal{M}] \longrightarrow [0, 1]^{[\mathcal{M}]}$$

$$d \longmapsto \chi_d$$

Ces fonctions vont nous permettre de définir des concepts et des classes imprécis.

6.1.2 Fonctions caractéristiques floues

Lorsque la fonction caractéristique est à valeur dans $[0, 1]$, l'ensemble des observations recouvert par une description d est un sous-ensemble flou défini sur l'ensemble des observations \mathcal{O} .

Cette fonction caractéristique devra mesurer le recouvrement d'une observation par une description d comme le degré avec lequel l'observation satisfait les propriétés de d . Par analogie avec les fonctions classiques, une observation est *totale*ment recouverte par une description si elle satisfait toutes les propriétés de la description. Par opposition, une observation n'est *pas du tout* recouverte par une description d si d et o n'ont aucune propriété commune. Par conséquent, une fonction caractéristique floue doit vérifier les contraintes suivantes :

Définition 6.1.1 Soit $[\mathcal{M}]$ l'ensemble des descriptions possibles pour le modèle \mathcal{M} . La fonction χ_d de $[\mathcal{M}]$ dans $[0, 1]$ est une fonction caractéristique cohérente pour une description d du modèle \mathcal{M} si,

$$\begin{aligned} \forall u \in [\mathcal{M}], \\ \chi_d(u) = 1 &\iff u \text{ satisfait totalement } d \text{ (cad. } u \leq d) \\ \chi_d(u) = 0 &\iff u \text{ et } d \text{ sont totalement incompatibles (cad. } u \wedge d = \perp) \end{aligned}$$

Proposition 6.1.1 L'utilisation de la distance bi-partite sur les attributs de type simple permet de définir des fonctions de reconnaissance $\chi = 1 - D_S(u, u \wedge d)$ ou encore $\chi = \chi_d(u)$ qui sont cohérentes.

Cette proposition est démontrée à partir des propositions 6.1.2 et 6.1.3. Lorsque la distance de Hamming ou Min-Max est utilisée, la deuxième condition de cohérence n'est pas vérifiée. En effet, u peut être totalement incompatible avec d tout en appartenant un peu à la description d , ce qui pose un problème de sémantique.

Pour le modèle des objets structurés, les fonctions caractéristiques sont construites à partir de la donnée de mesures de similarité (notées s), de la manière suivante,

$$\forall u \in [M], \quad \chi_d(u) = s(u \wedge d, u)$$

Une observation o appartient *totale*ment à une description d , lorsque o est moins générale que d , c'est à dire $o \leq d$. Par analogie, une observation o ($o \neq \perp$) et une description d sont *totale*ment incompatibles si $o \wedge d = \perp$.

Proposition 6.1.2 Pour toute mesure de similarité s , la fonction χ_d vérifie :

$$\forall u, d \in [M], \quad \chi_d(u) = s(u \wedge d, u) = 1 \iff u \leq d$$

preuve : Soit $u \in [\mathcal{M}]$

Comme χ_d est définie à partir d'une similarité s alors,

$$\chi_d(u) = 1 \iff s(u \wedge d, u) = 1$$

or la propriété de séparabilité assure que $u \wedge d = u$.

Par définition de la borne inférieure,

$$u \wedge d \leq d \iff u \leq d$$

◊

Lorsque δ est une distance normalisée, c'est à dire à valeur dans $[0, 1]$, alors $1 - \delta(u)$ définit une mesure de similarité. Ainsi, lorsque s est définie à partir de la distance D_S , cette contrainte est vérifiée.

Soient A_S l'ensemble fini des attributs de type simple $\{A_1, A_2, \dots, A_p\}$ et δ la distance associée à un attribut simple A de A_S . Lorsque la fonction de reconnaissance χ est définie par la distance D_S ou la similarité S , la proposition suivante est vérifiée :

Proposition 6.1.3 *Pour tout élément u, d de $[\mathcal{M}]$ tel que $u \neq \perp$,*

si $\chi_d(u) = 0$ et $u \wedge d = \perp$ alors $\delta(u_i, d_i \wedge u_i) = 1$

où d_i et u_i sont les composantes de d et u sur l'attribut A_i .

preuve : Soient $u = (u_i), d = (d_i)$ des descriptions de l'espace des descriptions $[\mathcal{M}]$ et D une distance de $[\mathcal{M}]$ définie par :

$$D(x, y) = 1 - \sum_{j \in \{1, \dots, p\}} w_j \delta(x_j, y_j)$$

où les poids w_j associés à chaque attribut de type simple A_j sont strictement positifs et vérifient $\sum_{j \in \{1, \dots, p\}} w_j = 1$.

Pour une fonction χ définie à partir de la distance D , si $\chi_d(u) = 0$ et $u \wedge d = \perp$ alors

$$\chi_d(u) = 1 - D(u, u \wedge d) = 0$$

comme $u \wedge d = \perp$ c'est à dire pour tout j , $u_j \wedge d_j = \perp$ alors

$$1 - \sum_{j \in \{1, \dots, p\}} w_j \delta(u_j, \perp) = 0$$

or $\sum_{j \in \{1, \dots, p\}} w_j = 1$ donc,

$$\sum_{j \in \{1, \dots, p\}} w_j (1 - \delta(u_j, \perp)) = 0$$

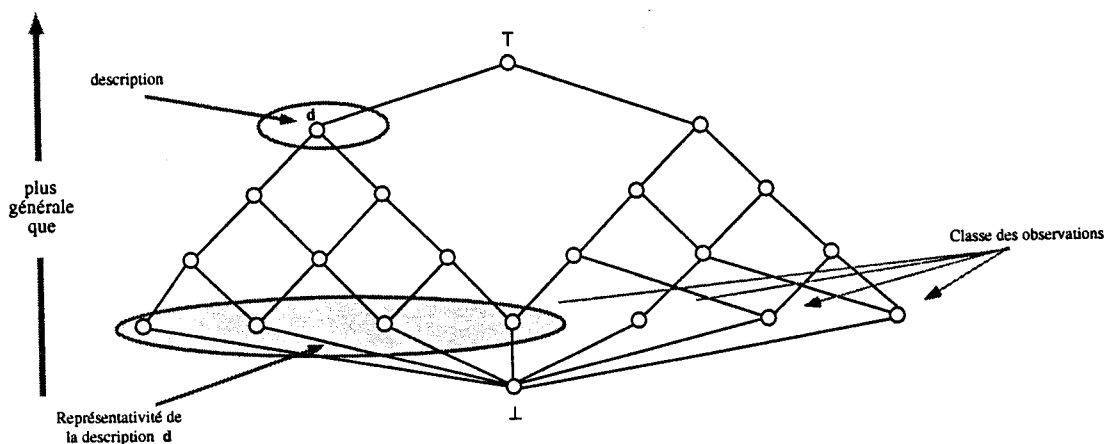


FIG. 6.1 - Représentativité d'une description en fonction de la relation d'ordre

de plus comme les poids sont strictement positifs, on a pour tout u_j , $1 - \delta(u_j, \perp) = 0$.

- Or lorsque la fonction χ est définie par D_S , on sait que D_S est une distance pondérée, normalisée par rapport aux attributs simples et par conséquent si $\chi_d(u) = 0$ et $u \wedge d = \perp$ alors d'après la preuve précédente pour tout j on a $\delta(u_j, \perp) = 1$.

- Lorsque χ est définie à partir de la similarité S , c'est à dire $\chi_d(u) = S(e \wedge u, u) = (1 - D(e \wedge u, u)) (1 - a \nabla(\Gamma(e \wedge u), \Gamma(u)))$ alors comme $0 \leq a < 1$, $\chi_e(u) = 0 \iff 1 - D(e \wedge u, u) = 0$. Or D est la distance moyenne pondérée sur les attributs de type simple, par conséquent on peut lui appliquer les résultats de la preuve précédente c'est à dire :

si $\chi_d(u) = 0$ et $u \wedge d = \perp$ alors pour tout j on a $\delta(u_j, \perp) = 1$.

◇

Cette proposition est importante, car elle montre qu'une fonction caractéristique définie à partir des distances et similarités du chapitre précédent, est cohérente que si la distance utilisée pour chaque attribut simple A vérifie pour tout $u \in [A]$ tel que $u \neq \perp$, la propriété $\delta(u, \perp) = 1$. Or parmi les distances Min-Max, Bi-partite et de Hamming, seule la distance Bi-partite vérifie cette propriété (prop. 5.2.1).

Exemple 6.1.2 - On considère comme modèle de représentation de données, l'attribut *corps* de l'exemple 4.4.1 sur lequel sera décrit les observations.

- Les observations a, b, c et la description e sont décrites sur ce modèle de la manière suivante,

- $a = ((0 / \text{subcylindrique}, 0 / \text{renflée}, 1 / \text{évasée}, 0 / \text{aplatie}, 0 / \text{mal-définie}), ((1 / \text{plane}, 0 / \text{déprimé}, 0 / \text{invaginé}), (0 / \text{unique}, 0 / \text{quatre}, 1 / \text{multiples})))$
- $b = ((0 / \text{subcylindrique}, 0 / \text{renflée}, 0 / \text{évasée}, \top / \text{aplatie}, 1 / \text{mal-définie}), ((0 / \text{plane}, 1 / \text{déprimé}, \perp / \text{invaginé}), (0 / \text{unique}, 1 / \text{quatre}, 0 / \text{multiples})))$

- $c = ((1 / \text{subcylindrique}, 0 / \text{renflée}, 0 / \text{évasée}, 0 / \text{aplatie}, 0 / \text{mal-définie}), ((1 / \text{plane}, 0 / \text{déprimé}, 0 / \text{invaginé}), (\top / \text{unique}, \top / \text{quatre}, \top / \text{multiples})))$
- $e = ((0 / \text{subcylindrique}, 0 / \text{renflée}, 1 / \text{évasée}, 0 / \text{aplatie}, 0 / \text{mal-définie}), ((1 / \text{plane}, 0 / \text{déprimé}, 0 / \text{invaginé}), (\top / \text{unique}, \top / \text{quatre}, \top / \text{multiples})))$
- L'ensemble $\mathcal{O} = \{ a, b, c \}$ représente les observations du contexte.

- Soit la fonction χ définie à partir de la distance D_S présentée au chapitre précédent.

Le concept (\mathcal{O}, χ_e) dont la fonction caractéristique est définie sur le modèle par $\chi_e(u) = 1 - D_S(u \wedge e, u)$, a pour extension floue :

$$\{ 1 / a, 0.25 / b, 0.8 / c \}$$

6.2 Généralités sur les concepts imprécis

Dans le langage courant, on parle de concept pour une représentation mentale abstraite ou générale d'un ensemble d'objets. Un concept est une représentation générale regroupant des observations possédant une ou plusieurs propriétés communes. Dans ce cas, un concept peut être défini soit par l'ensemble des individus : les exemples du concept, soit par une description générale permettant de caractériser les exemples de ce concept. On parle respectivement de concept défini en extension et en intention. La notion de concept en intention suppose implicitement l'existence d'un langage de représentation pour les concepts et celle de fonctions spécifiant l'appartenance ou non d'un individu au concept.

Définition 6.2.1 *Etant donné un ensemble d'observations \mathcal{O} , un ensemble de descriptions $[M]$ et une fonction $\chi \in [0, 1]^{[M]}$.*

On appelle concept imprécis, le couple (\mathcal{O}, χ_d) où d représente une description de $[M]$,

- χ_d désigne l'intention du concept C et,
- $\chi_d(\mathcal{O}) = \{ (o, \chi_d(o)), o \in \mathcal{O} \}$ représente l'extension de ce concept.

L'extension d'un concept défini en intention est obtenue simplement par l'application de la fonction de reconnaissance à l'ensemble des observations. Par contre, l'intention d'un concept défini en extension résulte en général d'un processus d'apprentissage. Il est clair que d'un point de vue utilisateur, une définition en intention est plus représentative pour un concept qu'une définition en extension.

L'ensemble des concepts ne se limite pas aux concepts formés par les couples (\mathcal{O}, χ_d) . Toute union finie de concepts est également un concept.

Définition 6.2.2 *Soient $C = (\mathcal{O}, \chi)$ et $C' = (\mathcal{O}, \chi')$ deux concepts définis sur le contexte $(\mathcal{O}, [M])$. Le couple $(\mathcal{O}, \chi_{C \cup C'})$ désigne le concept $C \cup C'$ où,*

$$\forall u \in [M], \quad \chi_{C \cup C'}(u) = \max(\chi(u), \chi'(u))$$

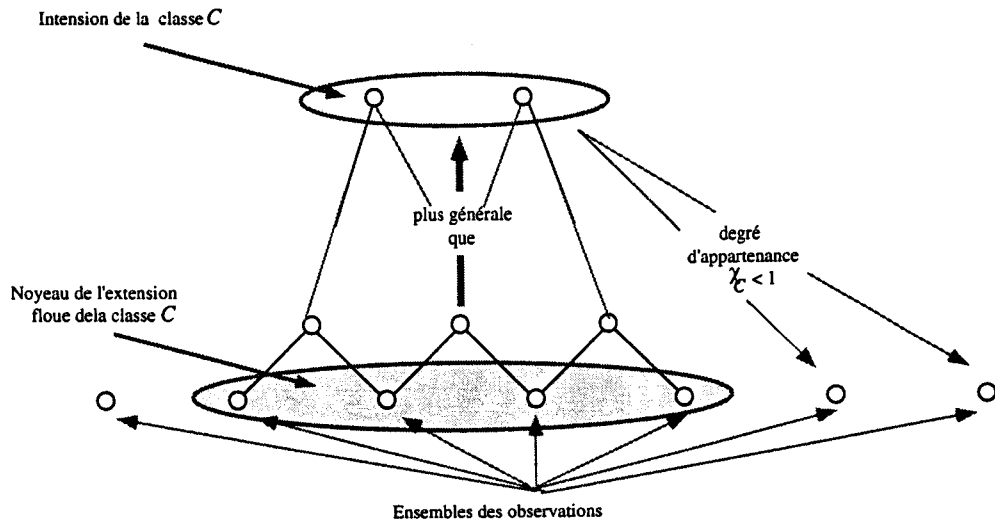


FIG. 6.2 – Intention et extension d'une classe imprécise

Lorsque C et C' sont respectivement issues des descriptions d et d' , on dit que le sous ensemble $\{d, d'\}$ caractérise le concept $C \cup C'$. L'extension du concept $C \cup C'$ correspond à l'union floue des extensions de C et C' où l'union est définie par le maximum. En théorie, il est possible de définir l'union de deux concepts à partir d'une conorme quelconque.

L'intention et l'extension d'un concept représentées dans ce chapitre, expriment l'imprécision de la représentation d'un ensemble d'observations par une description, par opposition au modèle des objets structurés, qui permet de décrire une observation de manière imprécise. De plus, nous avons vu que dans le cas des objets structurés, les fonctions de reconnaissance floues sont définies à partir de la distance bi-partite pour les attributs de type simple.

Chapitre 7

Qualité de la discrimination

Sommaire

7.1	Introduction	73
7.2	Définition de la complétude et de la consistance	74
7.2.1	Complétude et consistance	76
7.2.2	Les propositions floues quantifiées	77
7.3	Evaluation des propositions floues de types "les A sont B"	78
7.3.1	Implications floues et degré d'inclusion	79
7.3.2	Utilisation de mesures de similarité	80
7.3.3	Utilisation des α -coupes	82
7.4	Application sur un exemple "jouet"	84
7.5	Conclusion	87

7.1 Introduction

Une démarche usuelle pour l'acquisition de connaissances à partir d'une classe d'observations consiste à générer des concepts caractérisant la classe en explorant l'espace des descriptions possibles. Issu du modèle de représentation de données, cet espace de descriptions est généralement vaste. Aussi, pour caractériser une classe d'observations, il est essentiel de savoir explorer cet espace de manière efficace et de savoir mesurer la qualité de représentation de la classe.

Dans la littérature, les critères utilisés pour mesurer la qualité de discrimination d'un ensemble de concepts sont relatifs aux notions de complétude et de consistance.

- La fonction caractéristique d'un concept est dite *complète* si elle recouvre tous les exemples.

- La fonction caractéristique d'un concept est dite *consistante* si elle ne recouvre aucun contre-exemple.

Cependant les données d'applications réelles sont souvent bruitées et diminuent la pertinence de ces critères. L'utilisation de critères moins stricts permet de corriger partiellement la présence de données mal classées. De tels critères peuvent s'exprimer de la manière suivante :

- Une description de concept est complète si elle recouvre *la plupart* des exemples.
- Une description de concept est consistante si elle recouvre *peu de* contre-exemples.

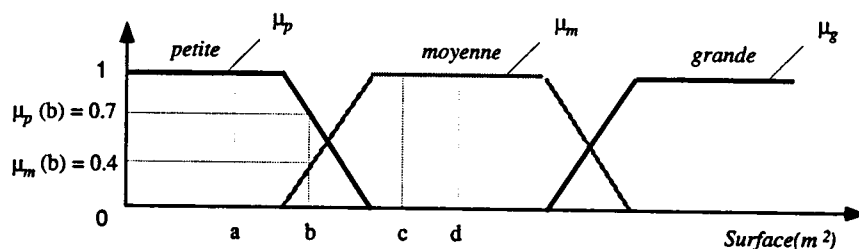
Dans ce chapitre, nous proposons des critères mesurant la qualité de recouvrement d'une fonction de reconnaissance d'un concept relativement à un ensemble d'exemples et de contre-exemples. La complétude et la consistance des descriptions de classes seront évaluées en fonction du degré de vérité de propositions floues quantifiées.

7.2 Définition de la complétude et de la consistance

Nous supposons donné un contexte $(\mathcal{O}, \mathcal{M})$ où \mathcal{O} représente l'ensemble des observations et \mathcal{M} le modèle de représentation de données. Chaque observation de \mathcal{O} est un élément de l'espace de description $[M]$. Les observations sont réparties en deux sous-ensembles, le sous-ensemble des exemples \mathcal{E} et celui des contre-exemples $\mathcal{C}_{\mathcal{E}}$. $(\mathcal{E}, \mathcal{C}_{\mathcal{E}})$ définit le concept à caractériser par des fonctions discriminantes.

Dans l'approche classique, une observation est un exemple ou un contre-exemple. Dans l'approche floue, l'appartenance d'une observation aux exemples ou aux contre-exemples, peut éventuellement être partielle. La gradualité de l'appartenance permet de décrire des classes aux contours imprécis. Les exemples et les contre-exemples forment ainsi une partition de l'ensemble des observations.

Exemple 7.2.1 On considère les observations $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$ ayant respectivement les valeurs $\{a, b, c, d\}$ pour l'attribut surface (fig. 3.2).



Sur cet exemple, lorsque l'on cherche à caractériser des observations de surface moyenne, \mathcal{O} est partitionné en ensembles,

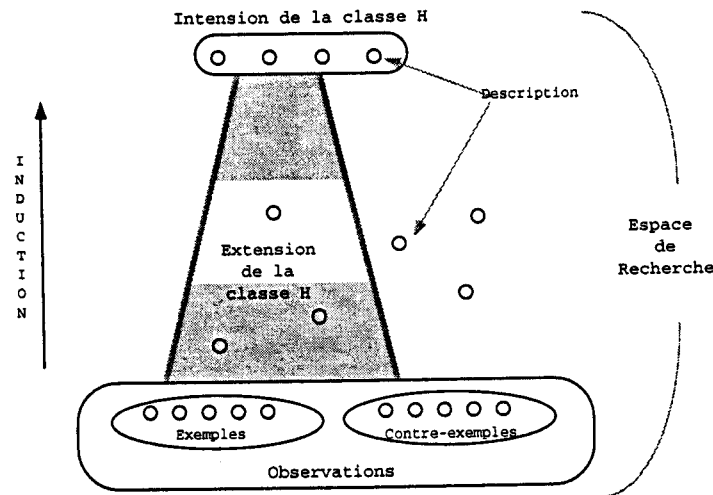


FIG. 7.1 - La classe dans son contexte

- des exemples

$$\mathcal{E} = \{(o_1, \mu_m(a) = 0), (o_2, \mu_m(b) = 0.4), (o_3, \mu_m(c) = 1), (o_4, \mu_m(d) = 1)\} \text{ et,}$$

- des contre-exemples

$$C_{\mathcal{E}} = \{(o_1, \mu_{pg}(a) = \max(\mu_p(a), \mu_g(a)) = 1), (o_2, \mu_{pg}(b) = 0.4), (o_3, \mu_{pg}(c) = 0), (o_4, \mu_{pg}(d) = 0)\}.$$

On ne décide pas à priori que l'observation o_2 est un exemple ou un contre-exemple.

La partition définie pour l'attribut surface est une partition floue maximale. Différentes formes de partition ont été formalisées par Bouchon-Meunier dans [5]. Pour une famille finie $\{E_1, \dots, E_n\}$ de sous-ensembles flous définis sur \mathcal{O} , voici quelques formes de partitions :

- Une partition floue normale vérifie :

$$\forall o \in \mathcal{O}, \sum_{i=1}^n \mu_{E_i}(o) = 1$$

- Une partition floue maximale vérifie :

$$\forall i \in \{1, \dots, n\}, h(E_i) = 1 \text{ et } h(E_i \cap E_k) < 1 \text{ pour } k \neq i$$

($h(E)$ est la hauteur du sous-ensemble flou E)

$$\forall o \in \mathcal{O}, \max_{i=\{1, \dots, n\}} \mu_{E_i}(o) = 1$$

où l'intersection est définie par la norme *min*.

- Une partition floue naturelle vérifie :

Pour tout élément o de \mathcal{O} , il existe une unique classe E_i telle que $\max_{i=\{1, \dots, n\}} \mu_{E_i}(o) = \mu_{E_i}(o)$

Les partitions définies ci-dessus assurent que chaque observation appartient *totale*ment à au plus, une classe. Ainsi comme dans l'approche classique, un exemple ne peut pas être un contre-exemple.

7.2.1 Complétude et consistance

La qualité de représentation d'un concept $C = (\mathcal{E}, \mathcal{C}_{\mathcal{E}})$ pour un contexte $(\mathcal{O}, \mathcal{M})$ par une classe \mathcal{H} est indépendante du processus d'induction utilisé. Illustrée par la figure 7.1, cette qualité revient à mesurer le degré de recouvrement des exemples et des contre-exemples.

Le recouvrement d'une classe \mathcal{H} définie comme un concept à l'aide d'une fonction de reconnaissance, est calculé à partir de son extension (chap. 6).

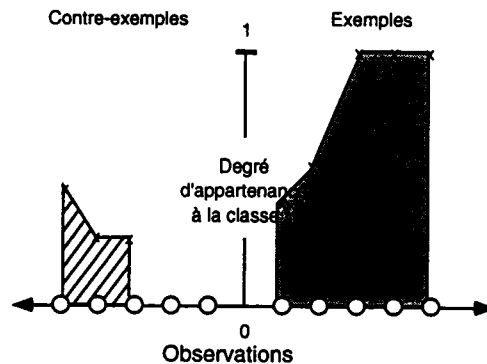


FIG. 7.2 - Recouvrement de la classe H

Les notions de complétude et de consistance d'une fonction de reconnaissance d'un concept (fig. 7.2) s'expriment par les propositions suivantes :

- *La plupart des exemples sont dans la classe H. (complétude)*
- *Peu de contre-exemples sont dans la classe H. (consistance)*

Notons que sur la figure 7.2, la classe des exemples et celle des contre-exemples sont données comme des ensembles ordinaires et sont représentées par leurs éléments les plus typiques. D'autre part, l'extension de la classe \mathcal{H} est floue et l'appartenance d'une observation à la classe varie entre 0 et 1. Comment prendre en compte cette imprécision dans les définitions de complétude et de consistance ?

7.2.2 Les propositions floues quantifiées

Les différentes propositions utilisées pour exprimer les notions de complétude et de consistance correspondent à des propositions floues quantifiées, dont la forme est :

Q des A sont dans B

- où
- Q est un quantificateur flou tel que : "tout", "la plupart", "la moitié", "peu de", "aucun"
 - A peut respectivement prendre la valeur \mathcal{E} ou $\mathcal{C}_{\mathcal{E}}$
 - B correspond à la classe d'approximation \mathcal{H} .

L'utilisation des quantificateurs flous permet de décrire des situations entre le quantificateur universel "tout" et le quantificateur existentiel "au moins un". Ces quantificateurs sont représentés comme des sous-ensembles flous sur l'ensemble des réels et plus généralement expriment un pourcentage (fig. 7.3).

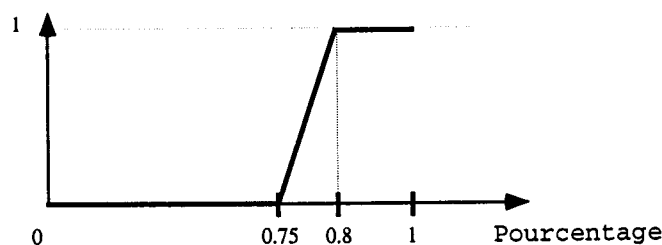


FIG. 7.3 – Une représentation du quantificateur "la plupart"

Un quantificateur n'est utilisé que pour donner un sens à l'évaluation du degré de vérité ou de possibilité de la proposition *les A sont B*.

Les propositions floues données précédemment permettent de traiter de manière plus naturelle l'imprécision des classes. Il existe d'autres types de propositions permettant d'assouplir la représentation précédente. L'utilisation de modificateurs linguistiques apporte un sens plus précis à la représentation des classes. Ces modificateurs sont des opérateurs de transformation de sous-ensembles flous dont les effets peuvent être d'élargir ou de réduire le noyau ou le support du sous-ensemble (fig 7.4). Bouchon-Meunier décrit dans [6] différentes familles de modificateurs,

- les modificateurs de renforcement tels que "très", dont le rôle est de *diminuer* l'imprécision. Le modificateur "très" peut être défini par l'application $f_t(x)$ de $[0, 1]$ dans $[0, 1]$ qui à tout x associe $f_T(x) = x^2$.
- les modificateurs d'affaiblissement ont pour effet *d'augmenter* l'imprécision. Par exemple, l'application f_{pm} qui à tout x de $[0, 1]$ associe la valeur $x^{\frac{1}{2}}$ définit le modificateur "plus ou moins".

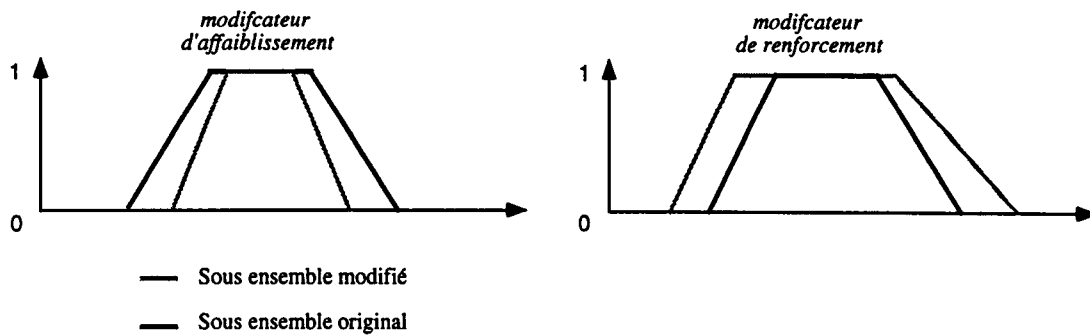


FIG. 7.4 – Exemples de modificateurs linguistiques

A l'aide de ces modificateurs, les propositions liées aux notions de complétude et de consistance s'affinent, soit :

- "la plupart des exemples sont plus ou moins dans la classe H "
- "peu de contre-exemples sont plus ou moins dans la classe H "

Dans notre cas, l'utilisation de modificateurs d'affaiblissement va permettre d'élargir le recouvrement. En effet, une classe est satisfaisante même si elle représente *presque* un concept $C = (\mathcal{E}, C_{\mathcal{E}})$. Mais parallèlement ces modificateurs facilitent la compréhension et l'interprétation des résultats.

7.3 Evaluation des propositions floues de types "les A sont B "

Dans la logique classique, les propositions de type "les A sont B ", où A et B sont des sous-ensembles, sont logiquement vraies si tous les éléments de A sont des éléments de B .

Dans la réalité, ces propositions sont rarement satisfaites dans leur totalité, dans le sens où les A sont B sauf pour certains éléments. L'insuffisance de la logique à formaliser simplement cette imprécision, a favorisé le développement de nouvelles logiques dites "non classiques" telles que :

- la logique des défauts, qui définit une quantification intermédiaire aux propositions quantifiées *tous les x sont* et *il existe un x tel que*, dont le sens est : *sauf exceptions, les x sont*.
- les logiques de l'incertain, telles que la théorie des possibilités qui caractérisent un événement par sa possibilité, mais aussi par celle de son contraire. Contrairement à la théorie des probabilités, la connaissance d'un événement ne permet pas de déterminer la possibilité de l'événement contraire.

- la logique floue, qui autorise l'utilisation de prédicats vagues ou encore de descriptions aux frontières mal définies.

...

Étant donné que nos concepts ont une extension floue, parmi ces différentes approches, nous nous sommes intéressés à la logique floue et plus précisément aux propositions floues quantifiées.

Différentes approches peuvent être utilisées pour évaluer des propositions floues quantifiées, dont la logique possibiliste [32]. Cependant dans notre cas, comme les données sont des instances de treillis où la négation n'existe pas forcément, il est difficile de concevoir convenablement des mesures de possibilité et de certitude à partir de l'appartenance d'une observation à un concept.

Une autre approche consiste à évaluer le degré de vérité de la proposition. Dans [20], Kacprzyk et Iwansky calculent ce degré de vérité en fonction de la proportion d'éléments de A recouverts par B et de quantificateurs flous (fig. 7.5).

Dans cette section, après un rappel sur les degrés d'inclusion basés sur les implications floues, nous allons développer deux méthodes permettant d'évaluer des propositions floues de type "les A sont B", l'une définie à partir de mesures de similarités, l'autre à partir des α -coupes d'un sous-ensemble flou.

7.3.1 Implications floues et degré d'inclusion

Pour un ensemble fini X , le degré d'inclusion noté $\mathcal{N}_{\underline{\sim}}$, d'un sous-ensemble flou A dans un sous-ensemble flou B définis sur X , peut être évalué par :

$$\mathcal{N}_{\underline{\sim}}(B|A) = \inf_{u \in X} \{ \mu_A(u) \xrightarrow{\sim} \mu_B(u) \}$$

où μ dénote la fonction caractéristique des sous-ensembles flous et $\xrightarrow{\sim}$ désigne une implication floue.

La plupart des implications floues¹ sont des généralisations de l'implication de la logique classique dite usuelle. Ainsi lorsque a et b sont des éléments de $\{0,1\}$ l'implication $a \xrightarrow{\sim} b$ et l'implication usuelle coïncident. Le tableau ci-dessous représente des implications floues classiques où les sous-ensembles ont le même ensemble de référence.

1. Les implications de Larsen et Mamdani définies respectivement par $\mu_A(u) \cdot \mu_B(u)$ et $\min(\mu_A(u), \mu_B(u))$ ne généralisent pas l'implication usuelle

Valeur de vérité $\mu_A(u) \xrightarrow{\sim} \mu_B(u)$	Nom
$\begin{cases} 1 & \text{si } \mu_A(u) \leq \mu_B(u) \\ \mu_B(u) & \text{sinon} \end{cases}$	Brouwer-Godël
$\min(1, 1 - \mu_A(u) + \mu_C(u))$	Lukasievick
$\max(1 - \mu_A(u), \mu_B(u))$	Kleene-Dienes
$\begin{cases} 1 & \text{si } \mu_A(u) \leq \mu_B(u) \\ 1 - \mu_B(u) & \text{sinon} \end{cases}$	Réciproque de Gödel

Selon l'implication utilisée, la fonction \mathcal{N}_{\supseteq} ne décrit pas les mêmes formes d'inclusion. Par exemple, pour l'implication de *Kleene-Dienes*, $\mathcal{N}_{\supseteq}(B|A) = 1$ signifie que $\text{Noyau}(B) \supseteq \text{Support}(A)$ alors que pour les autres implications, le degré d'inclusion est maximal dès que $B \supseteq A$.

Un autre exemple, $\mathcal{N}_{\supseteq}(B|A) = 0$ signifie pour la réciproque de Gödel que $\text{Noyau}(B) \not\supseteq \text{Noyau}(A)$ alors que pour l'implication de Lukasiewicz, elle indique que $\text{Support}(B) \not\supseteq \text{Noyau}(A)$.

Cependant, ce degré d'inclusion s'avère mal adapté pour mesurer le recouvrement par un concept \mathcal{H} des exemples ou des contre-exemples. En effet, la valeur $\mathcal{N}_{\supseteq}(\mathcal{H}|\mathcal{E})$ va définir le plus faible degré d'implication d'un exemple pour le concept \mathcal{H} . Ainsi, ce degré d'inclusion est trop strict surtout en présence de bruits dans les données. Nous nous intéressons à des fonctions donnant un aperçu plus général ou global de l'inclusion de deux sous-ensembles flous.

7.3.2 Utilisation de mesures de similarité

Par analogie avec la logique classique, le degré de vérité d'une proposition de type "les A sont B " doit être maximale lorsque $A \subset B$. Il paraît naturel de mesurer le degré d'inclusion de A dans B , pour évaluer la validité de ce type de proposition. Ce degré d'inclusion, noté $V(A \subset B)$, peut être défini comme l'adéquation entre A et $A \cap B$. En effet, B contient A si et seulement si A et $A \cap B$ sont semblables, c'est à dire que $A \cap B = A$. Nous pouvons alors nous placer d'un point de vue ensembliste en mesurant

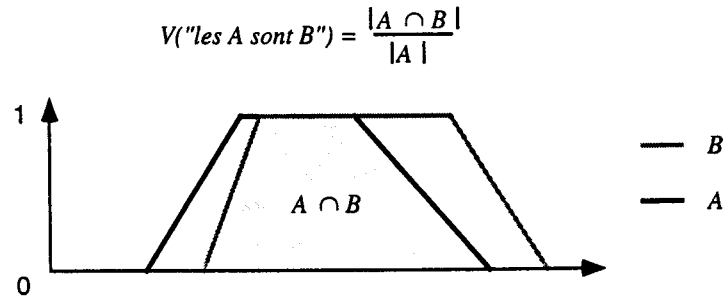


FIG. 7.5 - Inclusion de A dans B

cette adéquation comme la similarité des sous-ensembles flous A et $A \cap B$. Le choix d'une mesure de similarité se fera en fonction du problème à traiter.

Lorsque la mesure de similarité est définie à partir de la distance bi-partite, la fonction degré de vérité V que nous définissons est :

$$V(A \subset B) = 1 - d_b(A, A \cap B) = \sum_{x \in \mathcal{O}} \frac{\mu_{A \cap B}(x)}{\mu_A(x)} \quad (1)$$

où μ désigne la fonction caractéristique des sous-ensembles flous et l'intersection \cap est définie par la norme *min*. En considérant A comme l'ensemble des exemples et B comme la classe apprise H , la fonction V est une généralisation dans les sous-ensembles flous de l'indice de Jaccard et est utilisé comme degré de vérité par Kacprzyk et Iwansky dans [20].

Les fonctions de vérité sont bien définies lorsqu'elles sont compatibles avec la transitivité de l'inclusion, c'est à dire que si $B \subset C$ alors $V(A \subset B) < V(A \subset C)$.

En particulier, les fonctions construites à partir des distances de *Hamming généralisée relative* ou non, ou à partir de la distance bi-partite, vérifient cette compatibilité.

Pour des propositions floues, avec modificateurs linguistiques m , il s'agit de mesurer le degré d'inclusion de A dans $m(B)$, où $m(B)$ est le sous-ensemble B modifié. Il s'agit alors de déterminer $V(A \subset m(B))$. Dans le cas précédent,

$$V(A \subset m(B)) = \sum_{x \in \mathcal{O}} \frac{\mu_{A \cap m(B)}(x)}{\mu_A(x)}$$

Ainsi pour un concept $C = (\mathcal{E}, \mathcal{C}_\mathcal{E})$, le degré de vérité des propositions "la plupart des exemples sont plus ou moins dans la classe H " et "peu de contre-exemples sont

1. $\sum_{x \in \mathcal{O}} \frac{\mu_{A \cap B}(x)}{\mu_A(x)}$ est définie uniquement pour des ensembles A non vides. Dans le cas contraire, par définition de la distance bi-partite on a $d_b(\emptyset, \emptyset) = 0$ et $V(\emptyset \subset B) = 1 - d_b(\emptyset, \emptyset) = 1$, c'est à dire que l'ensemble vide est totalement inclus dans n'importe quel sous-ensemble.

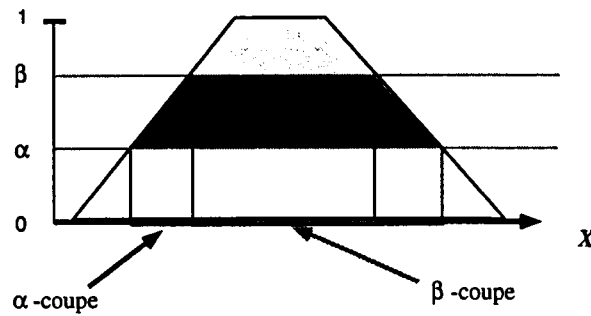


FIG. 7.6 – Exemples de coupes

plus ou moins dans la classe H ” se formule respectivement par $\mu_{Q_1}(V(\mathcal{E} \subset m(H)))$ et $\mu_{Q_2}(V(\mathcal{C}\mathcal{E} \subset m(H)))$, où Q_1, Q_2 sont les quantificateurs *la plupart* et *peu de* et m est un modificateur.

7.3.3 Utilisation des α -coupes

Dans [27], le recouvrement des exemples par la classe H est calculé en fonction des meilleurs exemples. L'intérêt est de diminuer l'influence des éléments mal classés. L'approche utilisée ignorait les éléments dont l'appartenance était inférieure à un seuil t . Les inconvénients de cette approche sont :

- le choix de la valeur α ,
- le biais induit par l'élimination des valeurs les moins significatives.

L'idée pour mesurer l'inclusion de A dans B en diminuant l'influence des éléments peu significatifs de A , est de comparer les sous-ensembles A et $A \cap B$ par coupes. Une α -coupe A_α d'un sous-ensemble flou A de référence X , est l'ensemble des éléments de X appartenant à A avec un degré supérieur à α , ou encore $A_\alpha = \{x \in X / \mu_A(x) \geq \alpha\}$ (fig. 7.6). Le recouvrement des éléments de A par B au niveau α noté r_{ACB}^α , sera mesuré à l'aide de l'indice de Jaccard, où :

$$r_{ACB}^\alpha = \begin{cases} \frac{|A_\alpha \cap B_\alpha|}{|A_\alpha|} & \text{pour } |A_\alpha| \neq 0 \\ 0 & \text{sinon} \end{cases}$$

Pour un ensemble fini et non vide S de valeurs de $[0, 1]$ toutes distinctes, le degré d'inclusion par coupe de A dans B est calculé en fonction des recouvrements r_{ACB}^α où $\alpha \in S$, de la manière suivante :

$$V[S](A \subset B) = \frac{\sum_{\alpha \in S} \alpha r_{ACB}^\alpha}{\sum_{\alpha \in S} \alpha}$$

Par exemple, en prenant $S = \{0, 0.2, 1\}$ avec $r_{ACB}^{0.2} = 1$ et $r_{ACB}^1 = 0.1$, nous obtenons $V[S](A \subset B) = \frac{0.2 \times 1 + 1 \times 0.1}{1.2} = 0.175$. L'objectif étant de diminuer l'influence des éléments peu significatifs de A , l'influence de chaque α -coupe est régulée par le niveau de coupe α . Lorsque $S = \{0\}$, on pose par définition $V[S](A \subset B) = 1$, pour tous les sous-ensembles flous A et B .

Proposition 7.3.1 $V[S](A \subset B) = 1 \iff \forall \alpha \in S, A_\alpha \subset B_\alpha$

preuve :

$$\begin{aligned} V[S](A \subset B) = 1 &\iff \sum_{\alpha \in S} \alpha(1 - r_{ACB}^\alpha) = 0 \\ &\iff \forall \alpha \in S, r_{ACB}^\alpha = 1 \quad (\text{car } \forall \alpha \in S, r_{ACB}^\alpha \leq 1) \\ &\iff \forall \alpha \in S, |A_\alpha \cap B_\alpha| = |A_\alpha| \\ &\iff \forall \alpha \in S, A_\alpha \subset B_\alpha \end{aligned}$$

◇

Remarque :

- Le degré d'inclusion par coupe dépend de la partition S . Pour que cette valeur soit significative, il faut que la partition soit régulière. En d'autres termes, si S est de cardinal $n + 1$ alors $S = \{\frac{i}{n}, 0 \leq i \leq n\}$. Dans l'exemple ci dessus, il est évident que la partition est trop dispersée pour pouvoir donner de bons résultats.

- Si nous avons $A \subset B \implies V[S](A \subset B) = 1$, sa réciproque est fautive, il suffit pour cela de prendre $S = \{0\}$. Le degré d'inclusion par coupe donne une estimation du degré d'inclusion réel. En prenant $S = \{\mu_A(x), x \in X\}$, la réciproque devient valide quelque soit le sous-ensemble B .

La fonction V est-elle compatible avec la transitivité de l'inclusion?

Comme pour les fonctions définies à partir de similarité, cette fonction a un sens si elle est compatible avec la transitivité de l'inclusion dont l'expression est : $A \subset B$ et $B \subset C$ alors $A \subset C$. En d'autres termes, tout sur-ensemble de B recouvre mieux A que B . Nous allons donc montrer la proposition suivante :

Proposition 7.3.2 Si $B \subset C$ alors $V[S](A \subset B) \leq V[S](A \subset C)$

preuve : Soit A un sous-ensemble flou d'un ensemble de référence fini X .

- Si A est vide alors $\forall \alpha \in S, |A_\alpha| = 0$ et par définition quelque soit le sous-ensemble flou B de X , $r_{ACB}^\alpha = 0$, c'est à dire $V[S](A \subset B) = 0$.

Nous pouvons donc conclure que si $B \subset C$ alors $V[S](A \subset B) \leq V[S](A \subset C)$.

- Si A est non vide et $B \subset C$ alors $A \cap B \subset A \cap C$ et,

$$A \cap B \subset A \cap C \quad \Rightarrow \quad A_\alpha \cap B_\alpha \subset A_\alpha \cap C_\alpha \quad \text{propriétés des } \alpha\text{-coupes}$$

$$A_\alpha \cap B_\alpha \subset A_\alpha \cap C_\alpha \quad \Rightarrow \quad \frac{|A_\alpha \cap B_\alpha|}{|A_\alpha|} \leq \frac{|A_\alpha \cap C_\alpha|}{|A_\alpha|} \quad \text{par définition de } |\cdot|$$

Par conséquent, $\forall \alpha \in S$, $r_{ACB}^\alpha \leq r_{ACC}^\alpha$ et $\alpha r_{ACB}^\alpha \leq \alpha r_{ACC}^\alpha$. Le passage à la somme sur les α laisse invariante l'inégalité donc $V[S](A \subset B) \leq V[S](A \subset C)$.

◇

L'évaluation des propositions floues quantifiées se différencie de la méthode précédente dans l'utilisation du quantificateur numérique Q . Ce dernier ne s'applique plus de manière globale sur la valeur V mais de manière simultanée sur chaque niveau de coupe α . Le degré d'inclusion obtenu s'exprime alors par :

$$V_Q[S](A \subset B) = \frac{\sum_{\alpha \in S} \mu_Q(\alpha r_{ACB}^\alpha)}{\sum_{\alpha \in S} \alpha}$$

Pour les propositions floues modifiées, le modificateur m s'utilise de la même manière que dans le cas des similarités. Il est important de noter que la fonction V_Q , n'exprime pas de degré d'inclusion mais correspond à une proposition quantifiée. Comme pour les similarités, le comportement de cette fonction dépend fortement du quantificateur ,

- pour le quantificateur *la plupart*, $B \subset C \Rightarrow V(A \subset B) \leq V(A \subset C)$
- pour le quantificateur *peu de*, $B \subset C \Rightarrow V(A \subset B) \geq V(A \subset C)$

Pour un concept $C = (\mathcal{E}, \mathcal{C}_\mathcal{E})$ à caractériser; où Q_1, Q_2 sont les quantificateurs *la plupart* et *peu de* et m est le modificateur *plus ou moins*; la complétude et la consistance de la classe H s'écrivent $V_{Q_1}[S](\mathcal{E} \subset m(H))$ et $V_{Q_2}[S](\mathcal{C}_\mathcal{E} \subset m(H))$.

7.4 Application sur un exemple "jouet"

Soient le contexte $(\mathcal{O}, \mathcal{M})$ et le concept $C = (\mathcal{E}, \mathcal{C}_\mathcal{E})$ définis par :

- $\mathcal{O} = \{o_1, o_2, \dots, o_{15}\}$
- \mathcal{M} un modèle de représentation de données.

- \mathcal{E} et $C_{\mathcal{E}}$ les sous-ensembles définis dans la table ci-dessous.

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}	o_{11}	o_{12}	o_{13}	o_{14}	o_{15}
\mathcal{E}	0	0.2	0.6	0	1	1	1	1	1	1	0.8	0.4	0.2	0.2	0
$C_{\mathcal{E}}$	1	1	0.4	0	0	0	0	0	0	0	0.2	0	0.8	1	1
H_1	0	0.2	0.6	0	1	1	1	1	0.5	0.5	0	0.4	0.2	0.2	0
H_2	0	0	0.6	0	1	1	1	1	1	0.2	0.8	0	0	0	0
$\mathcal{E} \cap H_1$	0	0.2	0.6	0	1	1	1	1	0.5	0.5	0	0.4	0.2	0.2	0
$\mathcal{E} \cap H_2$	0	0	0.6	0	1	1	1	1	1	0.2	0.8	0	0	0	0
$C_{\mathcal{E}} \cap H_1$	0	0.2	0.4	0	0	0	0	0	0	0	0	0.2	0.2	0	0
$C_{\mathcal{E}} \cap H_2$	0	0	0.4	0	0	0	0	0	0	0.2	0	0	0	0	0

Les classes et leurs intersections

Les classes H_1 et H_2 représentent les classes apprises et dont les extensions sont données dans la table ci-dessus. Notre objectif est de comparer la qualité de la représentation du concept C par les classes H_1 et H_2 . Il s'agit donc de mesurer la complétude et la consistance de ces classes à l'aide des degrés de vérité définis précédemment.

Nous utilisons une définition courante des quantificateurs flous "la plupart" et "peu de", notés respectivement Q_1 et Q_2 , où :

$$\mu_{Q_1}(t) = \begin{cases} 1 & , & 0.8 \leq t \\ 2t - 0.6 & , & 0.3 \leq t < 0.8 \\ 0 & , & t < 0.3 \end{cases}$$

et,

$$\mu_{Q_2}(t) = \begin{cases} 1 & , & t \leq 0.2 \\ -2t + 1.4 & , & 0.2 < t \leq 0.7 \\ 0 & , & 0.7 < t \end{cases}$$

- Cas où la fonction de vérité est définie à partir de l'indice de Jaccard généralisée (conf. 7.3.2).

-	$\mu_{Q_1}(V(\mathcal{E} \subset -))$	$\mu_{Q_2}(V(C_{\mathcal{E}} \subset -))$
H_1	1	1
H_2	1	1

Qualité de recouvrement

Les classes H_1 et H_2 sont satisfaisantes car complètes et consistantes. Cependant, le critère utilisé ne permet pas de différencier les classes H_1 et H_2 , même si la classe H_2 contient les exemples les plus représentatifs. Cet inconvénient n'a pas pour unique origine l'utilisation quantificateurs car, dans le cas de la consistance, $V(\mathcal{E} \subset -) = \frac{6.6}{8.4} =$

0,79 pour H_1 ou H_2 . Le degré d'inclusion défini à partir de l'indice de Jaccard accorde trop d'importance aux éléments peu significatifs.

- Cas où la fonction de vérité est définie à partir des α -coupes.

Soient $r_1(\alpha) = \frac{|\mathcal{E}_\alpha \cap H_{1\alpha}|}{|\mathcal{E}_\alpha|}$ et $r_2(\alpha) = \frac{|C_{\mathcal{E}_\alpha} \cap H_{1\alpha}|}{|C_{\mathcal{E}_\alpha}|}$ pour $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$

α -coupes	$r_1(\alpha)$	$\mu_{Q_1}(r_1(\alpha))$	$r_2(\alpha)$	$\mu_{Q_2}(r_2(\alpha))$
$\alpha = 1$	0.67	0.73	0	1
$\alpha = 0.8$	0.57	0.54	0	1
$\alpha = 0.6$	0.63	0.65	0	1
$\alpha = 0.4$	0.89	1	0.17	1
$\alpha = 0.2$	0.92	1	0.57	0.54
$\alpha = 0$	1	1	1	0

$V_{Q_1}[S](\mathcal{E} \subset H_1) = 0.72$ $V_{Q_2}[S](C_{\mathcal{E}} \subset H_1) = 0.97$

Répartition des coupes

Par construction, la fonction de vérité basée sur les coupes, diminue l'influence des exemples moins significatifs. Or, comme le montre le tableau ci-dessus, la classe H_1 ne recouvre pas bien les meilleurs exemples. Contrairement à la mesure précédente, le degré de vérité de la proposition "la plupart des exemples sont H_1 " n'est pas maximal. Mesurer l'adéquation de deux sous-ensembles flous en évaluant la ressemblance des coupes, donne un meilleur aperçu de la répartition des éléments.

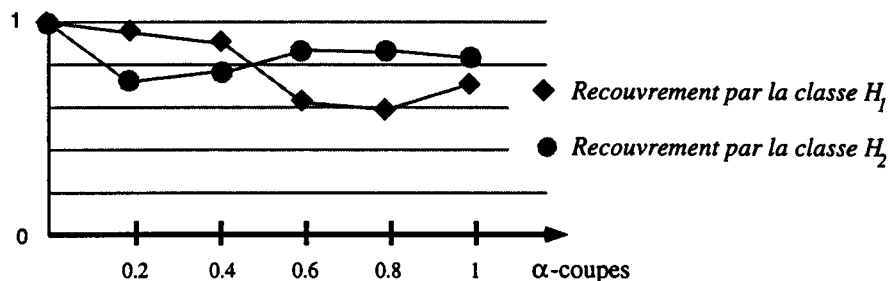


FIG. 7.7 - Exemples de recouvrements par coupes

Illustré par la figure 7.7, La classe H_2 recouvre convenablement la classe des exemples pour chaque coupe et donne des résultats plus significatifs pour les exemples les plus typiques. Pour cette classe, la valeur de la complétude est $V_{Q_1}[S](\mathcal{E} \subset H_2) = 0.98$, signifiant presque que la plupart des exemples sont dans H_2 .

7.5 Conclusion

Dans le cadre de la discrimination, nous nous sommes intéressés à mesurer la qualité de discrimination d'une classe définie en intention. Cette qualité décrite en termes de complétude et de consistance, est mesurée comme des degrés de vérité de propositions floues quantifiées, et est calculée comme le degré d'inclusion d'un ensemble A dans un ensemble B .

Pour évaluer ce degré d'inclusion, deux critères ont été présentés. La première est basée sur une approche classique consistant à mesurer, à partir de fonctions de similarité, la ressemblance entre A et $A \cap B$. Dans la deuxième, il s'agit de mesurer et de quantifier la ressemblance de chaque α -coupe des ensembles A et $A \cap B$. L'intérêt est de diminuer l'importance des coupes les moins significatives.

Cependant, cette fonction dépend des seuils utilisés pour définir les coupes. Ces derniers feront partie des paramètres de notre système.

Chapitre 8

Méthode de discrimination

Sommaire

8.1	L'espace de recherche : Le treillis $\mathcal{T}_\alpha[A]$	90
8.1.1	Définitions et propriétés	90
8.1.2	Le treillis des descriptions discriminantes	92
8.1.3	Méthode de construction du treillis $\mathcal{T}_\alpha[A]$	95
8.1.4	Exemple	98
8.2	Caractérisation de la classe des exemples	100
8.2.1	Critère de qualité d'un concept	100
8.2.2	Méthode de parcours du treillis \mathcal{T}_α	101
8.2.3	Exemple	103
8.3	Complexité des algorithmes	104
8.4	Optimisation de la discrimination	106
8.5	Méthode de recherche des attributs discriminants	108
8.5.1	Pouvoir discriminant d'un ensemble d'attributs	109
8.5.2	Algorithme de sélection des attributs discriminants	111

Dans ce chapitre, nous définirons l'espace de recherche des descriptions α -discriminantes et un procédé pour le générer. L'espace de recherche est engendré de manière incrémentale par rapport aux meilleurs exemples du concept. Cependant, l'algorithme de construction est indépendant de l'ordre de traitement des exemples. Les propriétés du domaine vont permettre de construire efficacement cet espace de recherche.

Ensuite, à partir d'un critère mesurant la qualité de discrimination, basé sur les notions de complétude et de consistance (chap. 7), nous proposons deux méthodes de recherche de *concept*. Ces méthodes ont pour objectif de déterminer à partir de l'espace de recherche un ensemble de description recouvrant les exemples et reconnaissant *peu* de contre-exemples.

8.1 L'espace de recherche : Le treillis $\mathcal{T}_\alpha[A]$

La définition de l'espace de recherche s'appuie sur la notion de descriptions α -discriminantes. Pour l'approche symbolique, une description ne convient pas dans la construction d'une fonction discriminante, dès lors qu'elle reconnaît un contre-exemple, [30]. Cependant, la présence de bruit dans les données nécessite une contrainte moins forte. Il est, par exemple, habituel de considérer les descriptions dites α -discriminantes [24, 34] dont voici une définition :

Une description est α -discriminante si sa fonction caractéristique reconnaît moins de α contre-exemples.

Dans notre approche, les fonctions de reconnaissance χ associées à une description sont floues et une observation peut appartenir *un peu, plus ou moins, presque ...* à une description discriminante.

Dans les paragraphes ci-dessous, à partir de la notion de description discriminante définie dans un cadre flou, nous allons montrer que l'espace de recherche peut être muni d'une structure de treillis.

8.1.1 Définitions et propriétés

Les définitions et propriétés de ce paragraphe sont établies relativement à un contexte d'apprentissage $(\mathcal{O}, \mathcal{M}, \mathcal{E}, \mathcal{C}_\mathcal{E})$. $\mathcal{O}, \mathcal{E}, \mathcal{C}_\mathcal{E}$ représentent respectivement les ensembles des observations, des exemples et des contre-exemples. Tous ces individus sont décrits sur le modèle de représentation de données $\mathcal{M} = \{ (A_j, D_j, V_j, T_j), j \in \{1, \dots, p\} \}$

Définition 8.1.1

- a) *Une observation o est valide pour une description d par rapport à un modificateur linguistique m si et seulement si $\mu_m(\chi_d(o)) = 1$. Dans ce cas, o est dit m -valide pour d .*
- b) *Une description est dite α -discriminante si elle reconnaît moins de α contre-exemples.*

Nous montrerons que la fonction de reconnaissance χ définie à partir de la distance bi-partite (chap. 5), est compatible avec la relation d'ordre sur $[M]$ dont sont issues les propriétés suivantes.

Rappels :

- Soient $u = (u_j), e = (e_j)$ des éléments de $[M]$, w_j des poids associés à l'attribut A_j vérifiant $\sum_j w_j = 1$ et d_b la distance bi-partite utilisée pour les attributs

simples. La fonction de reconnaissance, basée sur la distance pondérée par sa structure D_S , associée à la description e est :

$$\chi_e(u) = 1 - D_S(e \wedge u, u) = 1 - \sum_{j \in \{1, \dots, p\}} w_j d_b(e_j \wedge u_j, u_j)$$

- Soient $u = (u_j), e = (e_j)$ des éléments de $[\mathcal{M}]$. La fonction de reconnaissance, basée sur la similarité structurelle S , associée à la description e est :

$$\chi_e(u) = S(e \wedge u, u) = f(D(e \wedge u, u), \nabla(\Gamma(e \wedge u), \Gamma(u)))$$

où - $f(u, v) = (1 - u)(1 - av)$ avec $0 \leq a < 1$,

$$- D(x, y) = 1 - \sum_{j \in \{1, \dots, p\}} w_j d_b(x_j, y_j),$$

$$- \nabla(\Gamma(x), \Gamma(y)) = 1 - \sum_{j \in \{1, \dots, p\}} w_j \delta(\Gamma(x_j), \Gamma(y_j))$$

w_j, w'_j sont des poids associés à chaque attribut de type simple, dont la somme vaut 1.

Proposition 8.1.1 Lorsque la fonction de reconnaissance χ est définie à partir de la distance pondérée par la structure D_S ou de la similarité structurelle S alors : $\forall e, f \in [\mathcal{M}]$,

$$e \leq f \implies \forall o \in [\mathcal{M}], \quad \chi_e(o) \leq \chi_f(o)$$

preuve :

a) Cas où la fonction χ dépend de D_S

Soient $e, f, o \in [\mathcal{M}]$ avec $e \leq f$.

$$e \leq f \implies e \wedge o \leq f \wedge o$$

par isotonie de la fonction
 $g(x) = x \wedge o$

$$\implies \forall j \in \{1, \dots, p\}, \quad (e \wedge o)_j \leq (f \wedge o)_j$$

par définition du modèle \mathcal{M}

$$\implies \forall j \in \{1, \dots, p\}, \quad h((e \wedge o)_j) \leq h((f \wedge o)_j)$$

par isotonie de la fonction
hauteur

Comme χ est définie à partir de la distance bi-partite alors pour chaque attribut A_j du modèle \mathcal{M} :

$$\frac{h((e \wedge o)_j)}{h(o_j)} \leq \frac{h((f \wedge o)_j)}{h(o_j)}$$

(Lorsque $h(o_j) = 0$ nous avons par définition $\chi_{e_j}(o) = 1$.)

donc,

$$d_b(f_j \wedge o_j, o_j) \leq d_b(e_j \wedge o_j, o_j)$$

c'est à dire,

$$D_S(f \wedge o, o) = \sum_{j \in \{1, \dots, p\}} w_j d_b(f_j \wedge o_j, o_j) \leq D_S(e \wedge o, e) = \sum_{j \in \{1, \dots, p\}} w_j d_b(e_j \wedge o_j, o_j)$$

c'est à dire que

$$\chi_e(o) = 1 - D_S(e \wedge o, e) \leq \chi_f(o) = 1 - D_S(f \wedge o, o)$$

b) Cas où la fonction χ dépend de ∇

La distance moyenne D et la distance structurelle ∇ sont de la même forme que D_S , où δ pour la distance ∇ est la distance bi-partite appliquée à la chaîne $A \prec P \prec I$. Comme la fonction Γ est un morphisme d'ordre, en appliquant les résultats de la preuve précédente à D et à ∇ , on obtient pour $e \leq f$ donc $\Gamma(e) \leq \Gamma(f)$:

$$\begin{aligned} - D(f \wedge o, o) &\leq D(e \wedge o, o) \\ - \nabla(\Gamma(f \wedge o), \Gamma(o)) &\leq \nabla(\Gamma(e \wedge o), \Gamma(o)) \end{aligned}$$

Or la fonction f est décroissante pour la composante en u et en v , il en résulte que $S(e \wedge o, o) \leq S(f \wedge o, o)$.

◇

Proposition 8.1.2 *Toutes les descriptions plus générales qu'une description non α -discriminante ne sont pas α -discriminantes.*

$\forall e, f \in [\mathcal{M}]$ tel que $e \leq f$, non (e α -discriminant) \implies non (f α -discriminant)

preuve :

Soient $e, f \in [\mathcal{M}]$ avec $e \leq f$.

Si e n'est pas α -discriminant alors il existe au moins $\alpha + 1$ contre-exemples $ce_1, ce_2, \dots, ce_{\alpha+1}$ tels que $\forall i \in \{1, \dots, n\}$, $\chi_e(ce_i) = 1$.

Or nous avons montré que pour χ définie à partir de d ou S , si $e \leq f$ alors $\chi_e(ce_i) \leq \chi_f(ce_i)$. Par conséquent, f reconnaît au moins $\alpha + 1$ contre-exemples et n'est pas α -discriminant.

◇

Remarque : Pour que ces propriétés soient conservées lors de l'utilisation de modificateurs linguistiques, il est nécessaire que ceux-ci aient une fonction caractéristique croissante. Il est facile de voir que dans le cas contraire l'inégalité de la propriété a) est fausse.

8.1.2 Le treillis des descriptions discriminantes

Soit un ensemble d'observations A dans le contexte $(\mathcal{O}, \mathcal{M}, \mathcal{E}, \mathcal{C}_{\mathcal{E}})$.

$\mathcal{T}_{\alpha}[A]$ est le treillis des descriptions α -discriminantes de $[\mathcal{M}]$ engendré par les éléments

de A et complété par le plus petit élément \perp et le plus grand élément \top du treillis des descriptions $[\mathcal{M}]$, c'est à dire :

$$\mathcal{T}_\alpha[A] = \perp \oplus \{d = \bigvee_{s \in S} s \mid S \subset A \text{ et } d \text{ est } \alpha\text{-discriminant}\} \oplus \top$$

où \oplus désigne la somme linéaire des treillis.

Nous montrerons que $\mathcal{T}_\alpha[A]$ possède une structure de treillis de longueur finie.

Lemme : Pour toutes les descriptions discriminantes e et f de \mathcal{T}_α , on a

$$e \vee f \text{ } \alpha\text{-discriminante} \iff e \vee f \in \mathcal{T}_\alpha$$

\vee est l'opérateur de supremum défini pour $[\mathcal{M}]$.

preuve : Soient e et f des descriptions α -discriminantes de $\mathcal{T}_\alpha[A]$.

- Par définition de $\mathcal{T}_\alpha[A]$, il existe deux sous-ensembles S_e et S_f de A tels que $e = \bigvee_{s \in S_e} s$ et $f = \bigvee_{s \in S_f} s$.

- Par commutativité et associativité de l'opérateur de borne supérieure \vee dans $[\mathcal{M}]$, nous avons $e \vee f = \bigvee_{s \in S_f \cup S_e} s$.

Donc $e \vee f$ est α -discriminante alors $e \vee f$ est un élément de $\mathcal{T}_\alpha[A]$.

◇

Proposition 8.1.3 Pour toutes les descriptions discriminantes e et f de \mathcal{T}_α , \vee_α défini par :

$$\forall e, f \in \mathcal{T}_\alpha[A], \quad e \vee_\alpha f = \begin{cases} e \vee f & \text{si } e \vee f \text{ est } \alpha\text{-discriminant} \\ \top & \text{sinon} \end{cases}$$

est l'opérateur de borne supérieure de $\mathcal{T}_\alpha[A]$.

preuve :

Soient e et f des descriptions α -discriminantes de $\mathcal{T}_\alpha[A]$.

Dans $[\mathcal{M}]$, l'unique borne supérieure de e et f est $e \vee f$.

Si cette description est α -discriminante, elle est la plus petite description plus générale que e et f pour la relation d'ordre et d'après le lemme précédent appartient à $\mathcal{T}_\alpha[A]$.

Sinon toutes les descriptions plus générales ne sont pas α -discriminantes et le supremum

de $e \vee f$ est par construction \top .

◇

Proposition 8.1.4 *Tout couple d'éléments de $\mathcal{T}_\alpha[A]$ possède une borne inférieure. Pour e et f éléments de $\mathcal{T}_\alpha[A]$, $e \wedge_\alpha f$ est le plus grand élément de $\mathcal{T}_\alpha[A]$ inférieur à e et f .*

preuve :

Soient e et f des descriptions α -discriminantes de $\mathcal{T}_\alpha[A]$.

La contraposé de la propriété 8.1.1 assure que s'il existe deux descriptions a et b de $\mathcal{T}_\alpha[A]$, α -discriminantes et moins générales que e et f , alors $a \vee b$ est une description α -discriminante et d'après le lemme $a \vee b \in \mathcal{T}_\alpha[A]$.

Dans le cas, où l'ensemble des minorants de e et f est non vide, la remarque précédente assure que e et f possèdent un plus grand minorant. En effet, si a et b sont deux plus grands minorants de f et e , on en déduit par la propriété précédente que $a \vee b$ est une description discriminante, vérifiant :

$a \vee b \leq a$ et $a \vee b \leq b$ par définition de la généralisation et $a \vee b$ est minorant de e et f plus grand que a ou b . Cette propriété est possible que si $a \vee b = a$ et $a \vee b = b$, c'est à dire $a = b$. Ainsi, dans le cas où l'ensemble des plus grands minorants de e et f est non vide, il existe un plus grand minorant e et f .

Comme \perp est le plus petit élément du treillis, nous pouvons conclure qu'il existe au moins une borne inférieure à deux éléments e et f de $\mathcal{T}_\alpha[A]$.

◇

Proposition 8.1.5 *$(\mathcal{T}_\alpha[A], \leq, \wedge_\alpha, \vee_\alpha, \top, \perp)$ possède une structure de treillis de longueur finie.*

preuve :

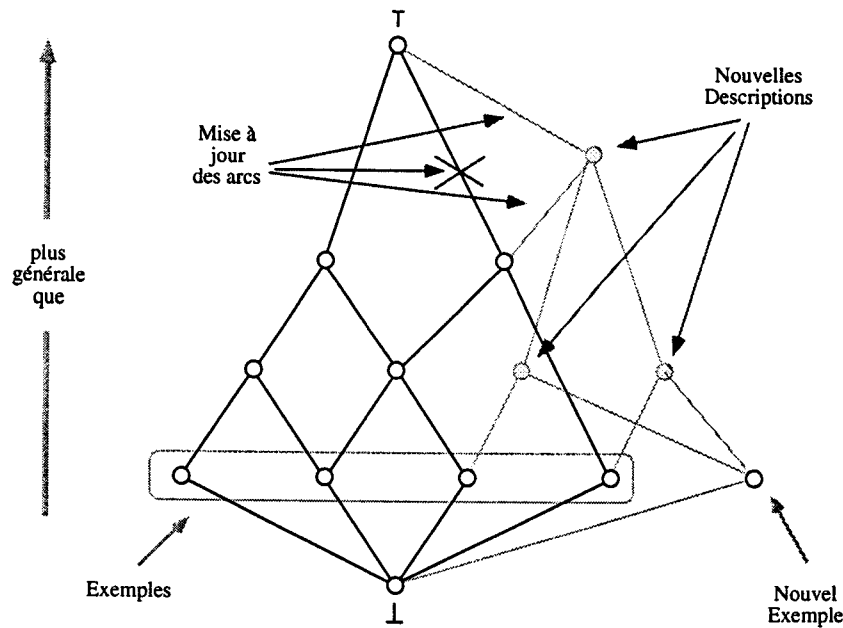
Comme $\mathcal{T}_\alpha[A]$ est un sous-ensemble du treillis $[\mathcal{M}]$ qui est de longueur finie, alors $\mathcal{T}_\alpha[A]$ est également de longueur finie et possède \perp et \top comme minorant et majorant universels.

De plus, nous avons montré ci-dessus que tout couple de $\mathcal{T}_\alpha[A]$ possède une borne supérieure et une borne inférieure donc $\mathcal{T}_\alpha[A]$ est muni d'une structure de treillis.

◇

Remarque :

- $\mathcal{T}_\alpha[A]$ n'est pas un sous-treillis de $[\mathcal{M}]$. En effet le supremum dans $\mathcal{T}_\alpha[A]$ de deux

FIG. 8.1 - Insertion d'un nouvel exemple dans $\mathcal{T}_\alpha[A]$

descriptions α -discriminantes peut être par construction, différent de celui dans $[M]$, c'est à dire: $e \vee_\alpha f = e \vee f$ ou $= *$.

- $\mathcal{T}_\alpha[A]$ ne contient pas toutes les descriptions α -discriminantes, mais uniquement celles qui peuvent être construites à partir des observations de A . En d'autres termes, ce sont les descriptions α -discriminantes les plus proches des données.

8.1.3 Méthode de construction du treillis $\mathcal{T}_\alpha[A]$

Illustrée par la figure 8.1, la méthode de construction du treillis $\mathcal{T}_\alpha[A]$ consiste pour chaque observation o , à générer le treillis des descriptions α -discriminantes issues de $A \cup \{o\}$ à partir de celui issu de A . La proposition suivante définit les propriétés élémentaires sur lesquelles se base notre méthode.

Proposition 8.1.6 Pour un treillis quelconque \mathcal{T} , $\forall x, y, o \in \mathcal{T}$, tel que $x \leq y$

1. Si $x \vee o \leq y$ alors,
 $\forall z \in \mathcal{T}, y \leq z \implies z \vee o = z$.
2. Si $x \vee o > y$ alors,
 $x \leq y \implies x \vee o = y \vee o$.

Preuve :

1. Soit $z \in \mathcal{T}$,

$$\begin{aligned}
 x \vee o \leq y &\implies o \leq y && \text{par définition de } \vee \\
 o \leq y \text{ et } y \leq z &\implies o \leq z && \text{par transitivité de } \leq \\
 &\implies o \vee z \leq z && \text{par isotonie de } _ \vee z \\
 &\implies o \vee z = z && \text{car par définition de } \vee \quad z \leq z \vee o
 \end{aligned}$$

2. Soit $z \in \mathcal{T}$,

$$\begin{aligned}
 x \vee o > y \text{ et } y \leq x &\implies x \vee o \geq y \vee o \text{ et } \geq y \vee o \geq x \vee o && \text{par isotonie de } _ \vee o \\
 &\implies x \vee o = y \vee o && \text{par antisymétrie} \\
 &&& \text{de la relation d'ordre}
 \end{aligned}$$

◇

Algorithme de construction de \mathcal{T}_α

La fonction de généralisation, appelée **générer** décrite ci-dessous, construit de manière récursive le treillis des descriptions α -discriminantes à partir d'un nœud n et d'une observation o . Le problème pour chaque nœud n est d'établir la mise à jour des liens, pour chaque nouvelle généralisation $g = o \vee n$. En procédant ainsi, la méthode s'appuie sur la structure du treillis de l'étape précédente d'une part pour orienter les généralisations à effectuer et d'autre part pour insérer à la bonne place les nouveaux nœuds. Cette fonction débute pour le nœud associé à la borne inférieure.

Fonction g n rer (observation o , n ud n)

1. Si le n ud n est d j  trait , alors retourner le n ud g .
 2. G n ralisation entre o et n ud n .
Si le n ud $o \vee n$ existe d j  alors l'affecter   g .
Sinon cr er un n ud $g = o \vee n$.
 3. Si g est identique   n alors retourner le n ud n .
 4. Si g est α -discriminant alors pour tout successeur s du n ud n faire
 - a - Si $g < s$ alors - ins rer s comme successeur de g
- Supprimer s comme successeur de n
La proposition 1 assure qu'il est inutile de continuer la proc dure g n rer sur le successeur s . Le n ud g s'intercale entre s et n .
 - b - Si $g > s$ alors effectuer la fonction **g n rer** avec comme argument l'observation o et le n ud s .
La proposition 2 affirme que g est un successeur de s , par cons quent aucune mise   jour doit  tre effectu e entre g et le n ud courant n .
 - c - Si g et s ne sont pas comparables **g n rer** avec comme argument l'observation o et le n ud s et **ins rer** la valeur retourn e de la fonction comme successeur de g .
 5. Si aucun successeur de n est inf rieur   g , **ins rer** g comme successeur de n .
 6. Si g ne poss de aucun successeur, ajouter le n ud associ    \top comme successeur de g .
 7. Retourner le n ud g .
-

L' tape d'insertion est importante car elle doit assurer la coh rence dans la liste des successeurs. En effet, elle doit supprimer de la liste des n uds, ceux qui sont sup rieurs au n ud ins r .

Proc dure ins rer (description d , liste de n uds $succ$)

1. y est un  l ment de la liste $succ$
2. Si la description d reconna t y alors supprimer y de $succ$ et **ins rer** y comme successeur de x .
3. Si la description y reconna t d alors **ins rer** d dans la liste des successeurs de y .

4. Répéter pour tout y de $succ$.
5. Si aucun élément de $succ$ est inférieur ou égal à x , ajouter x à $succ$.

Lors de l'implantation, les tests de comparaison sont optimisés en considérant la hauteur des éléments dans le treillis des descriptions $[M]$. Par exemple, deux éléments sont égaux si et seulement si leurs hauteurs le sont également. Cette remarque permet de diminuer considérablement la complexité, car les tests de comparaison entre deux descriptions sont fonction du nombre d'attributs et du cardinal de chaque domaine de l'attribut.

Il faut noter que lorsque la classe \mathcal{E} des exemples est précise, l'objectif est de construire $\mathcal{T}_\alpha[\mathcal{E}]$. Cependant, lorsque cette classe est floue, générer le treillis à partir de tous les éléments qui appartiennent au moins un peu aux exemples est inutile car ces éléments sont peu représentatifs de la classe. Nous nous limitons dans ce cas aux éléments à une coupe de niveau ε de \mathcal{E} et nous générons le treillis $\mathcal{T}_\alpha[\mathcal{E}_\varepsilon]$.

8.1.4 Exemple

On considère l'ensemble $A = \{a, b, c, d, e, f\}$.

Le contexte d'apprentissage est $(\mathcal{O}, \mathcal{P}(A), \mathcal{E}, \mathcal{C}_\mathcal{E})$, où :

- $\mathcal{O} = \{ \{a, b\}, \{d\}, \{b, c\}, \{a, c\} \}$,
- $\mathcal{E} = \{ \{a, b\}, \{d\}, \{b, c\} \}$,
- $\mathcal{C}_\mathcal{E} = \{ \{a, c\} \}$.

$(\mathcal{P}(A), \leq, \cap, \cup, \emptyset, A)$ est le treillis des descriptions. Nous allons construire sur cette espace des descriptions, l'espace des descriptions discriminantes $\mathcal{T}_\alpha[\mathcal{E}]$.

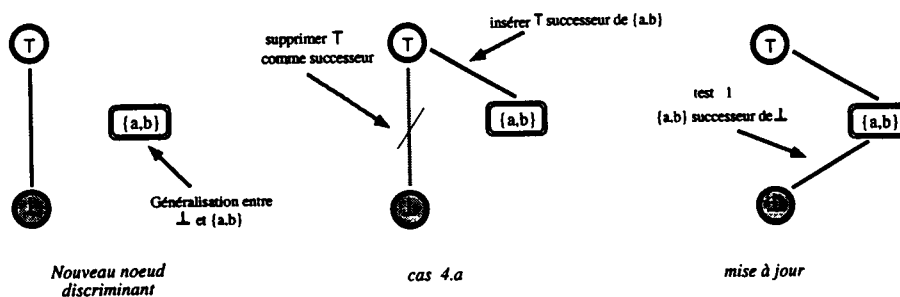


FIG. 8.2 - Insertion de l'exemple $\{a, b\}$

Comme le montre la figure 8.2, le treillis est initialement constitué des valeurs \perp et \top . La fonction d'appartenance utilisée est définie par la relation d'inclusion. Une observation est reconnue par un concept si sa description est contenue dans celle du concept.

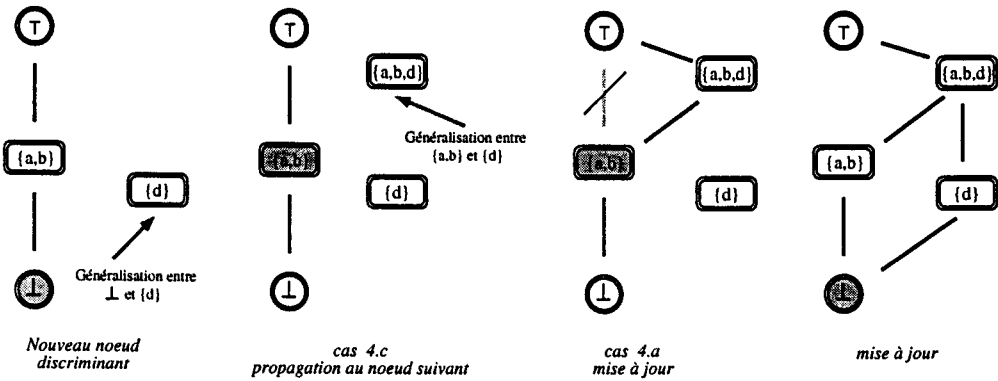


FIG. 8.3 – Insertion de l'exemple $\{d\}$

Pour cet exemple, une description discriminante ne doit recouvrir aucun contre-exemple. Chaque nœud grisé sur la figure est un nœud actif appliquant sa méthode *générer*.

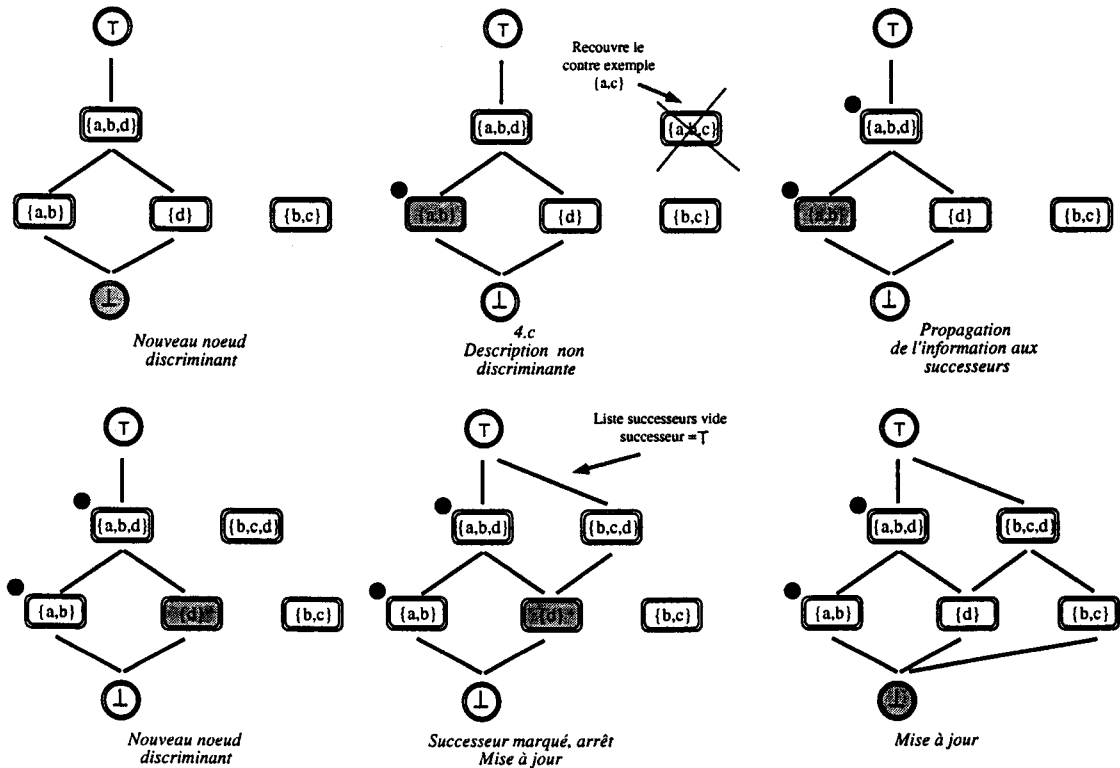


FIG. 8.4 – Insertion de l'exemple $\{b,c\}$

Un nœud est marqué, lorsqu'il ne peut pas produire de descriptions α -discriminantes. Par conséquent, d'après la propriété 8.1.1, tous ses successeurs ne sont pas α -discriminants et peuvent être marqués. Les arcs sont mis à jour progressivement pendant la généralisation.

8.2 Caractérisation de la classe des exemples

Par construction, chaque description du treillis \mathcal{T}_α discrimine au plus α contre-exemples et ne reconnaît qu'une partie des exemples. Le problème est d'obtenir, à partir des descriptions α -discriminantes, un concept \mathcal{H} représentant convenablement la classe des exemples et discriminant au mieux les contre-exemples. Dans le chapitre précédent, nous avons proposé différents critères pour mesurer la complétude et la consistance d'un concept. La qualité de discrimination d'un concept sera définie à partir de ces critères.

8.2.1 Critère de qualité d'un concept

Dans un contexte $(\mathcal{O}, \mathcal{M}, \mathcal{E}, \mathcal{C}_\mathcal{E})$, les indices mesurant la complétude et la consistance d'un concept \mathcal{H} sont respectivement notés $V_{Q_1}(\mathcal{E} \subset \mathcal{H})$ et $V_{Q_2}(\mathcal{C}_\mathcal{E} \subset \mathcal{H})$, où Q_1 et Q_2 sont des quantificateurs linguistiques et ont été définis dans le chapitre 7. Pour que ces mesures aient un sens, il faut que Q_1 et Q_2 soient des applications respectivement croissante et décroissante. Ces applications expriment des pourcentages de recouvrement par rapport aux exemples et aux contre-exemples. Les indices de complétude et de consistance étant des critères indépendants, nous pouvons donner comme indice mesurant la qualité globale d'un concept par la moyenne de ces deux mesures :

$$G(\mathcal{H}) = (V_{Q_1}(\mathcal{E} \subset \mathcal{H}) + V_{Q_2}(\mathcal{C}_\mathcal{E} \subset \mathcal{H}))/2$$

Etant donné les propriétés des fonctions Q_1 , Q_2 et V , la valeur de la fonction G augmente pour un concept \mathcal{H}' reconnaissant plus d'exemples que \mathcal{H} et discriminant au plus autant de contre-exemples, ou pour un concept \mathcal{H}' reconnaissant moins de contre-exemple que \mathcal{H} mais au moins autant d'exemples.

La méthode de construction du concept \mathcal{H} consiste à compléter \mathcal{H} par ajouts successifs de descriptions discriminantes qui augmentent la qualité de sa représentation. Le concept \mathcal{H} défini par l'ensemble des observations \mathcal{O} , peut maximiser le critère de qualité G mais dans ce cas il s'agit d'un concept trivial.

Aussi, lors de la construction du concept \mathcal{H} , le nombre de descriptions le composant doit être minimisé. Par construction du treillis \mathcal{T}_α , toutes ses descriptions¹ reconnaissent moins de α contre-exemples et les plus générales sont celles qui reconnaissent le plus d'exemples. De plus, toutes descriptions inférieures à celles représentant \mathcal{H} sont redondantes. Comme le montre la figure 8.5, le choix d'une description plus générale permet

1. Exception faite pour la borne supérieure du treillis \mathcal{T}_α .

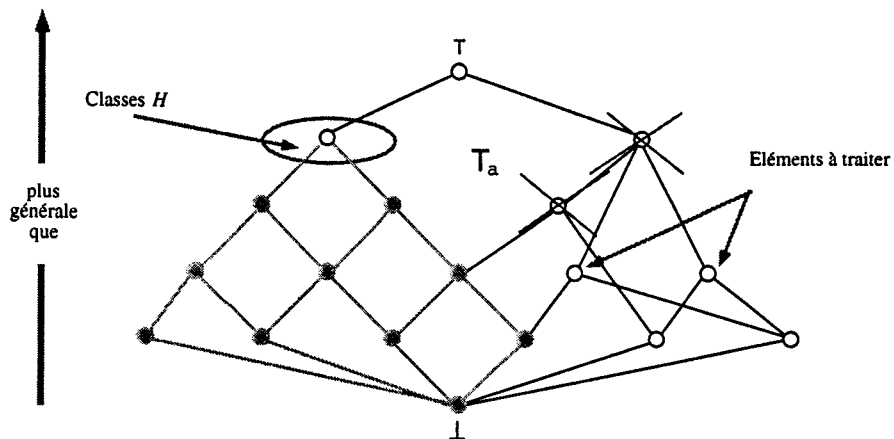


FIG. 8.5 – Recherche de concepts

de réduire considérablement le nombre de représentants possibles.

Dans le paragraphe suivant nous présenterons une méthode de parcours descendante du treillis \mathcal{T}_α pour construire le concept \mathcal{H} . Contrairement aux premières approches utilisées [26], [27], la structure d'ordre du treillis permet d'optimiser l'étape de recouvrement.

8.2.2 Méthode de parcours du treillis \mathcal{T}_α

Le principe de cette méthode est relativement simple. L'ensemble des plus grands éléments de \mathcal{T}_α distincts de la borne supérieure recouvre tous les exemples. Cependant cet ensemble n'est pas forcément consistant. Il s'agit dans une première étape de construire à partir de ces plus grands éléments, un concept \mathcal{H} satisfaisant le critère G . Dans une deuxième étape, nous réitérons ce processus sur les fils des plus grands éléments ne faisant pas partie du concept.

Dans le pire des cas, ce processus continue tant que la borne inférieure n'est pas atteinte et s'arrête dès que le critère G est maximal. Comme les indices de complétude et consistance varient dans l'intervalle $[0, 1]$, la valeur maximale du critère G est 1. La fonction de caractérisation qui résulte de ce principe est nommée *caractériser*.

fonction caractériser

1. Initialiser Pge avec les fils de la borne supérieure et le concept \mathcal{H} comme ensemble vide.
2. Initialiser la description d à vide

3. Pour tout x de la liste Pge mettre $d = x$ si $(G(\mathcal{H} \cup \{x\}) = G(\mathcal{H} \cup \{d\})$ et $hauteur(x) > hauteur(d)$) ou si $G(\mathcal{H} \cup \{x\}) > G(\mathcal{H} \cup \{d\})$.
 4. Si $d \neq \emptyset$ alors ajouter d à \mathcal{H}
 marquer les fils de d
 supprimer d de Pge
 5. Répéter 4 tant que $d \neq \emptyset$ et $G(\mathcal{H}) \neq 1$
 6. Construire la nouvelle liste des plus grands éléments $Pge = \text{ListePge}(Pge)$.
 7. Répéter 2 tant que $Pge \neq \emptyset$ et $G(\mathcal{H}) \neq 1$
 8. Retourner le concept \mathcal{H} .
-

Cette fonction retourne le concept \mathcal{H} obtenu pour le critère G et fait appel à la fonction *ListePge* qui construit à chaque étape la liste des plus grands éléments.

fonction ListePge(Liste)

1. Initialiser L comme ensemble vide et x comme un élément de *Liste*.
 2. Pour tout fils y de x non marqué faire
 - Supprimer de L tous les éléments de L supérieurs à y . - Ajouter y à L si aucun élément de L est supérieur à y .
 3. Supprimer x de *Liste*
 4. Répéter 2 tant que *Liste* n'est pas vide.
-

8.2.3 Exemple

Nous reprenons en partie l'exemple de construction de treillis avec comme modèle de représentation de données l'ensemble de toutes les parties de A , où $A = \{ a, b, c, d, e, f \}$. Pour que la caractérisation soit convenablement illustrée, le contexte est modifié de la manière suivante :

- $\mathcal{E} = \{ \{a, b\}, \{c, d\}, \{a, d\}, \{f, c\}, \{e, f\} \}$
- $C_{\mathcal{E}} = \{ \{a, f\}, \{d, f\}, \{a, e\}, \{b, d\} \}$

De plus, une description de concept est discriminante si elle recouvre ou contient au plus un contre-exemple (ie. $\alpha = 1$). Le treillis $\mathcal{T}_{\alpha}[\mathcal{E}]$ illustré par la figure 8.6.

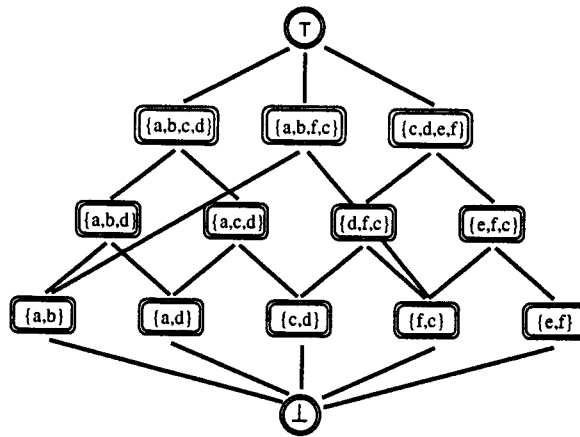


FIG. 8.6 – Initialisation de la caractérisation

Les quantificateurs utilisés pour le critère G sont *la plupart* pour la complétude et *peu de* pour la consistance (chap. 7.4).

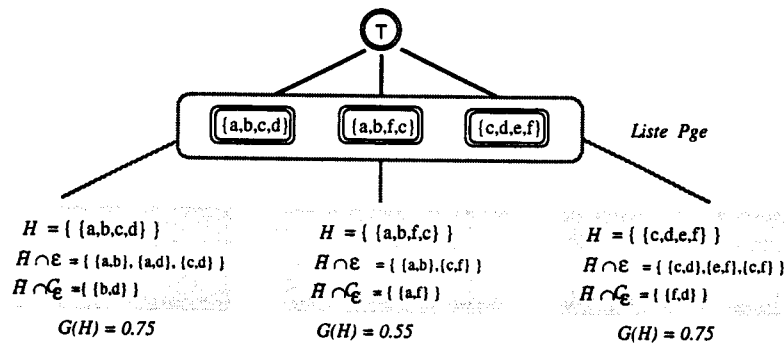


FIG. 8.7 – Choix de la 1^{ère} description de concept

La qualité du concept \mathcal{H} est identique pour les descriptions $\{a, b, c, d\}$ et $\{c, d, e, f\}$, et ces descriptions ont la même hauteur (ie. 3) dans le treillis \mathcal{T}_{α} , le choix s'effectue en fonction de l'ordre d'apparition.

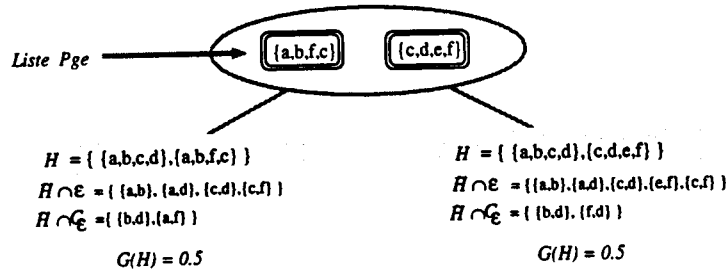


FIG. 8.8 - Liste Pge à renouveler

Les éléments restant de la liste Pge, ne permettent pas d'obtenir une meilleure caractérisation du contexte pour le critère G. Il s'agit de déterminer à partir de ces éléments la nouvelle liste Pge.

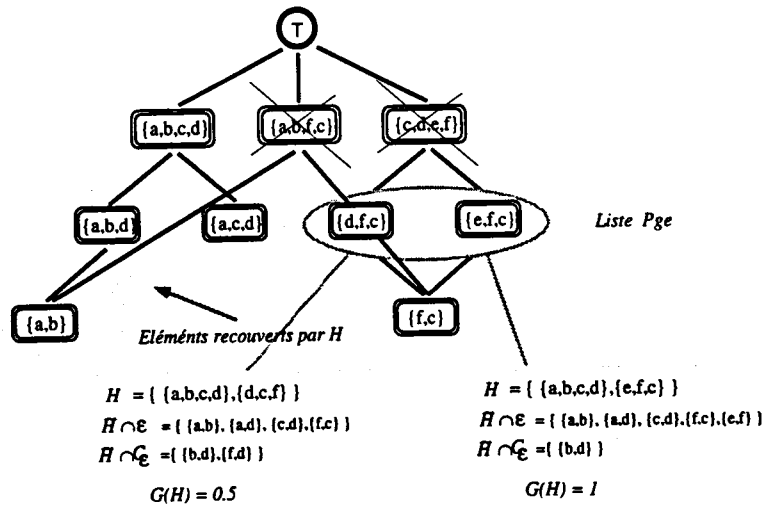


FIG. 8.9 - Valeur maximale atteinte, fin

Sur la figure 8.9, le critère G est maximal pour le concept $\mathcal{H} = \{ \{a, b, c, d\}, \{ \{c, d, e, f\} \} \}$ et l'algorithme s'arrête.

8.3 Complexité des algorithmes

Les opérations de base sur lesquelles s'appuient les algorithmes de ce chapitre sont la généralisation, la spécialisation, la fonction hauteur et les fonctions de reconnaissance. $(\mathcal{O}, \mathcal{P}(A), \mathcal{E}, \mathcal{C}_E)$ est le contexte d'apprentissage et A_S l'ensemble des attributs de type simple dont le cardinal est m .

- Si les attributs sont symboliques ou taxonomiques (conf. 4.3.3), la généralisation ou la spécialisation de deux descriptions e et f d'un attribut A revient à évaluer pour tout $d \in D_A$, $(\mu_e \vee \mu_f)(d)$ ou $(\mu_e \wedge \mu_f)(d)$. Le calcul du supremum ou d'infimum de

deux nuances est considérée comme une opération élémentaire. Pour un attribut de type simple A , la complexité de ces opérations est de $o(|D_A|)$ opérations élémentaires.

Dans le cas d'attributs structurés, généraliser (resp. spécialiser) deux descriptions consiste à généraliser (spécialiser) chaque valeur définie sur les attributs de type simple. Pour des valeurs décrites sur l'attribut structuré représentant le modèle, la complexité d'une généralisation est de $o(\sum_{i=1}^m |D_{A_i}|)$.

- Dans le cas d'un attribut de type simple, la hauteur d'une valeur μ est la somme des hauteurs de chaque valeur d du domaine, c'est à dire $h(\mu(d))$. Par conséquent la complexité de $h(\mu)$ est de $o(|D_A|)$ opérations élémentaires.

- La fonction de reconnaissance évaluant l'appartenance d'une observation o à une description de concept d , consiste à évaluer $1 - d(\mu_o, \mu_d \wedge \mu_o)$ ou $s(\mu_o, \mu_d \wedge \mu_o)$. Or pour toutes les distances ou similarités citées dans le chapitre 5, cette évaluation consiste à calculer une spécialisation et la hauteur des valeurs de chaque attribut de type simple. Par conséquent, la reconnaissance d'une observation par une description a une complexité de $o(\sum_{i=1}^m |D_{A_i}|)$.

- Dans le pire des cas, la fonction *discriminer* évalue le degré de reconnaissance de tous les "meilleurs" contre-exemples, ou encore les contre-exemples m -valides pour $\mathcal{C}_{\mathcal{E}}$. Par conséquent, la complexité de la fonction *discriminer* a pour ordre de grandeur $|m(\mathcal{C}_{\mathcal{E}})| \cdot o(\sum_{i=1}^m |D_{A_i}|)$.

- Lors de l'ajout d'une nouvelle observation o dans le treillis des descriptions discriminantes, chaque nœud du treillis effectue au plus une fois la méthode *génère*. Pour la construction du treillis $\mathcal{T}_{\alpha}[A \cup \{o\}]$, le nombre d'appels à la méthode *génère* est $|\mathcal{T}_{\alpha}[A]|$.

Or chaque fois que la méthode *génère* est appelée, la fonction *discriminer* est appelée également. La mise à jour du treillis de descriptions discriminantes pour l'ajout d'une observation a une complexité de l'ordre de $|\mathcal{T}_{\alpha}[A]| \cdot |m(\mathcal{C}_{\mathcal{E}})| \cdot o(\sum_{i=1}^m |D_{A_i}|)$.

Dans le pire des cas, le treillis $\mathcal{T}_{\alpha}[A]$ possède $\min(|[M]|, 2^{|A|})$ où $[M]$ est le nombre maximum de descriptions possibles pour le modèle \mathcal{M} et $2^{|A|}$ est le nombre d'éléments du treillis des parties issues de A .

- La fonction caractériser parcourt le treillis $\mathcal{T}_{\alpha}[A]$ en largeur. Même si la fonction de reconnaissance sur l'ensemble des observation \mathcal{O} est évaluée une seule pour chaque

description, le nombre de calculs du critère G dans le pire des cas est de l'ordre de $|\mathcal{T}_\alpha[A]|^2$.

Par conséquent, l'ordre de cette fonction est $\max(|\mathcal{T}_\alpha[A]|^2, |\mathcal{O}| \cdot o(\sum_{i=1}^m |D_{A_i}|) \cdot (|\mathcal{T}_\alpha[A]|))$.

8.4 Optimisation de la discrimination

La méthode précédente génère de manière exhaustive l'ensemble des descriptions α -discriminantes, pour ensuite en extraire une caractérisation des exemples. Dans cette partie, il s'agit d'une certaine manière, de caractériser les exemples "le plus rapidement possible", en évitant de générer toutes les descriptions discriminantes.

L'idée est de "fusionner" la procédure de généralisation et la procédure de caractérisation afin d'obtenir une méthode de parcours du treillis des descriptions discriminantes, sans avoir à les produire toutes. Le principe est le suivant :

- Pour chaque observation o ajoutée à un treillis \mathcal{T} , la méthode **générer** produit le treillis contenant \mathcal{T} et toutes les généralisations d'éléments de \mathcal{T} avec o (cad. $d \vee o$).
- La caractérisation de ce treillis détermine un ensemble de descriptions de \mathcal{T} discriminant les exemples et les contre-exemples. Ensuite, seules les descriptions moins générales que les descriptions de la caractérisation sont conservées. Cet ensemble auquel on ajoute la borne supérieure possède une structure de treillis.
- A partir de ce treillis, le procédé peut recommencer pour l'insertion d'une nouvelle observation.

fonction `optdisc`

1. Initialiser

\mathcal{T} avec la borne inférieure du treillis et la borne supérieure du treillis des descriptions,
 \mathcal{H} comme ensemble vide,
 o comme une observation de \mathcal{E} ,
 V à la valeur 0.

2. Appliquer la fonction **générer** à o et \mathcal{T}

3. Extraire le concept \mathcal{H} en effectuant la fonction **caractériser** au treillis \mathcal{T} . Affecter V de la qualité de discrimination $G(\mathcal{H})$ du concept \mathcal{H} .

4. Supprimer de \mathcal{T} toutes les descriptions non reconnues par les éléments de \mathcal{H} .

5. Répéter 2 pour toutes les observations de \mathcal{E} tant que $V \neq 1$.

6. Retourner le concept \mathcal{H} .

A chaque nouvelle insertion d'une observation, le nombre de descriptions discriminantes est inférieure au treillis des descriptions discriminantes issues de toutes les observations insérées. Cette restriction permet de réduire le coût de chaque étape.

La procédure *optdisc* est illustrée avec l'exemple "jouet" utilisé pour la caractérisation.

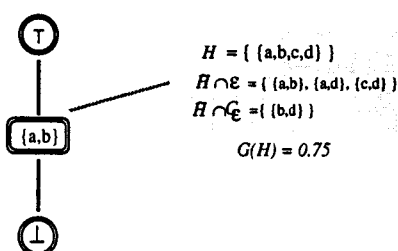


FIG. 8.10 – Initialisation

Pour les trois premières observations, le treillis des descriptions α -discriminantes est complet. Les deux procédures de discrimination donnent des résultats identiques.

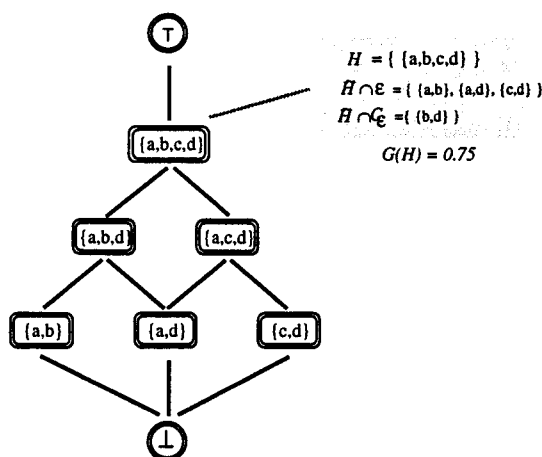


FIG. 8.11 – Insertion des premiers éléments

Comme le montre la figure ci-contre, lors de l'ajout de l'observation $\{f, c\}$, le treillis des descriptions discriminantes n'est plus complet. La procédure de caractérisation va permettre de restreindre l'ensemble des descriptions.

Les descriptions $\{a, b, f, c\}$ et $\{d, f, c\}$, n'appartenant pas à la caractérisation $\mathcal{H} = \{ \{a, b, f, c\}, \{d, f, c\} \}$, sont supprimées du treillis. Comme le concept \mathcal{H} maximise

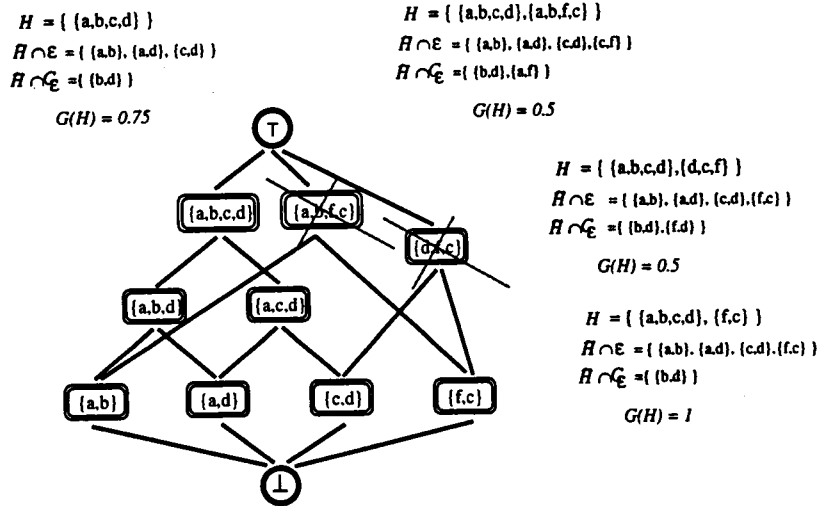


FIG. 8.12 - Exemple de réduction du treillis

le critère de qualité G , l'algorithme s'arrête. L'ajout de l'observation $\{ e, f \}$ est devenu inutile.

8.5 Méthode de recherche des attributs discriminants

Les études en Analyse de Données ont montré que le choix des attributs les plus discriminants pour un concept $C = (\mathcal{E}, \mathcal{C}_E)$ permet d'améliorer la qualité et la fiabilité des résultats [8]. Lorsque le nombre d'attribut est important, tel que pour les Hyalonema (fig. 8.13), limiter la discrimination sur les attributs les plus représentatifs permet de diminuer la complexité de l'algorithme de recherche de concept.

Cependant, la recherche exhaustive de variables les plus discriminantes est coûteuse en temps de calcul car elle nécessite de tester toutes les combinaisons d'attributs. Différentes méthodes de sélection performantes ont été développées telle que celle de Furnival et Wilson [8]. Les méthodes n'ont pas la prétention de trouver la "meilleure" combinaison d'attributs, mais un ensemble d'attributs, appelé également modèle, discriminant convenablement les données, en fonction d'un critère de sélection.

Sélection d'attributs avec remise en cause

Cette méthode procède pas à pas par ajout ou suppression d'attributs en fonction de deux critères. Plus précisément, à chaque étape l'attribut contribuant le plus au pouvoir de discrimination du modèle, lui est ajouté. Parallèlement, afin d'être maintenu dans l'ensemble des attributs sélectionnés, l'attribut du modèle satisfaisant le moins au pouvoir de discrimination, doit satisfaire un critère de présence. La méthode s'arrête

dès qu'un attribut ne peut plus être sélectionné ou supprimé.

Cette méthode est utilisée en Analyse des Données pour des variables continues. Le pouvoir de discrimination est mesuré usuellement par le Lambda de Wilks qui doit être minimisé.

Méthode de Furnival et Wilson

La méthode Furnival et Wilson permet d'obtenir l'ensemble des attributs discriminants le mieux deux groupes d'individus, en sélectionnant à chaque étape l'attribut qui maximise le coefficient de détermination de la régression R^2 [36]. Dans le cas de deux groupes d'individus, ce coefficient correspond à la valeur propre principale de la matrice $T^{-1}B$, où T représente la matrice de covariance totale du nuage de points et B est la matrice covariance inter-classe. Cette méthode s'avère peu efficace lorsque le nombre d'attributs est supérieur à 40.

Contribution de la structure dans la sélection des attributs

Les différentes méthodes de sélection nécessitent de rechercher pas à pas une combinaison d'attributs avec un pouvoir discriminant convenable. Dans la plupart des cas, il s'agit à chaque étape de déterminer l'attribut à ajouter à ceux déjà sélectionnés. Dans les méthodes précédentes, le pouvoir discriminant d'un ensemble d'attributs est mesuré à partir de critères statistiques valables pour des attributs numériques.

Nous allons présenter dans ce paragraphe, une méthode de sélection d'attributs les plus discriminants adapté à notre système d'apprentissage tirant profit de la structure arborescente des attributs hiérarchiques.

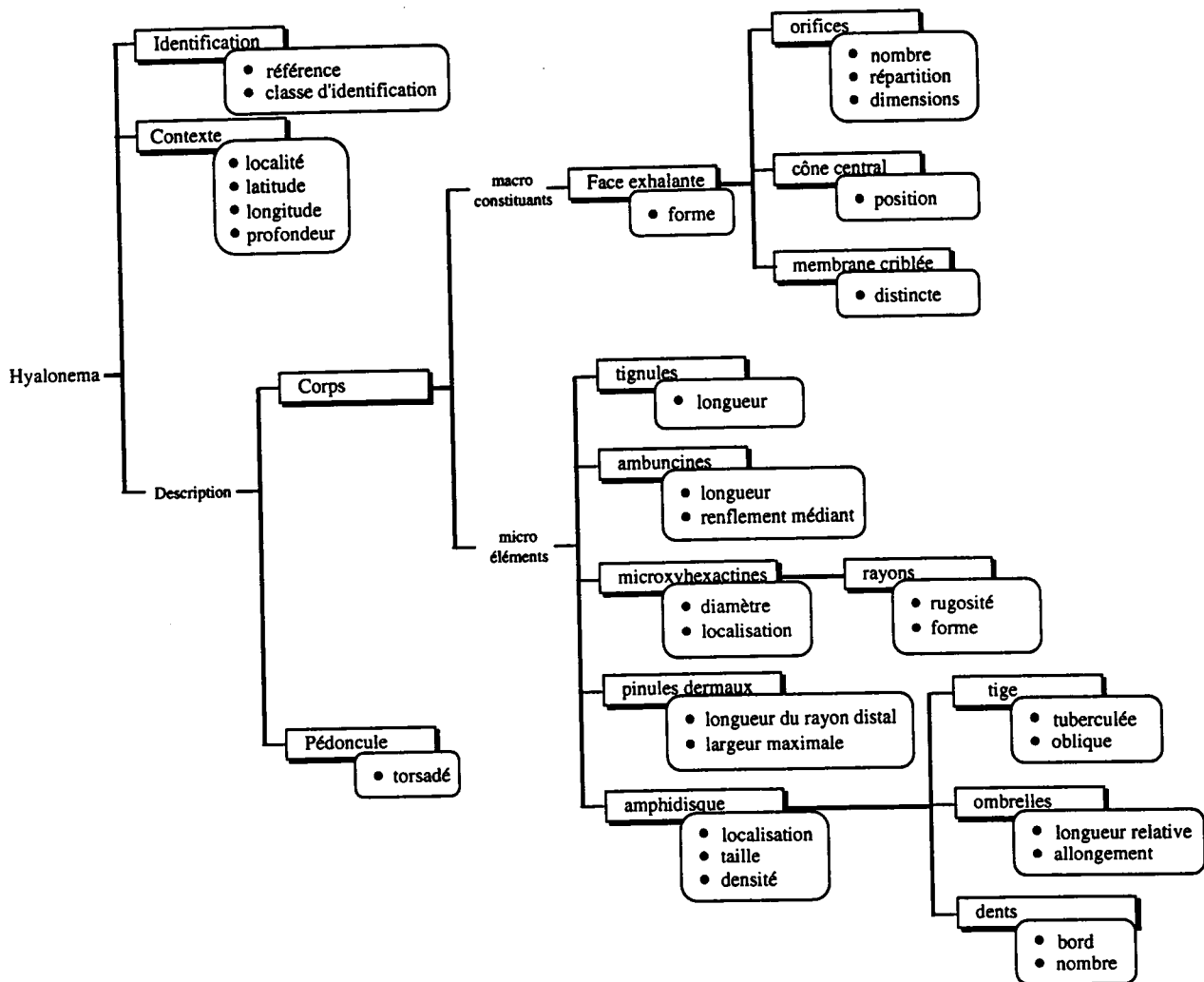
8.5.1 Pouvoir discriminant d'un ensemble d'attributs

On considère le concept $C = (\mathcal{E}, C_{\mathcal{E}})$ défini par l'ensemble des exemples \mathcal{E} et des contre-exemples $C_{\mathcal{E}}$. On définit :

- *Indice de séparabilité entre deux observations décrites sur p attributs A_1, \dots, A_p*

L'indice de séparabilité I_S évalue la différence maximale existante entre les composantes des observations comparées. Pour deux observations e, f décrites sur les attributs $\{A_1, \dots, A_p\}$, cet indice désigne le pouvoir discriminant de l'ensemble $\{A_1, \dots, A_p\}$ pour e et f .

$$I_S(e, f) = \max_{i \in \{1, \dots, p\}} d(e_{\Gamma(A_i)}, f_{\Gamma(A_i)})$$

FIG. 8.13 – Le genre *Hyalonema*

où $\Gamma(A_i)$ correspond à l'indice de l'attribut A_i dans le modèle global \mathcal{M} de représentation de données.

La distance d peut être la distance bi-partite, la distance de Hamming ou en encore la distance Min-max. Ces distances sont toutes normalisées à valeur dans $[0, 1]$. Ainsi $I_S(e, f) = 0$ signifie que e et f ne sont pas différenciables sur les p attributs alors que $I_S(e, f) = 1$ signifie qu'il existe au moins un attribut pour lequel les descriptions e et f sont totalement distinctes.

- *Indice de cohérence d'un exemple e sur p attributs A_1, \dots, A_p*

L'indice de cohérence I_C mesure avec quel degré un exemple se différencie des contre-exemples, lorsque sa description est restreinte aux attributs A_1, \dots, A_p . Cet indice est

calculé comme l'indice de séparabilité minimale entre e et les contre-exemples.

$$I_C(e) = \min_{c_e \in C_{\mathcal{E}}} I_S(e, c_e)$$

$I_C(e) = 0$ signifie qu'il existe au moins un contre-exemple qui soit identique à e pour la restriction à A_1, \dots, A_p .

- *Pouvoir discriminant d'un ensemble d'attributs pour le concept C*

Le pouvoir discriminant P_W des p attributs A_1, \dots, A_p correspond à la moyenne des indices de cohérence évalués pour chaque exemple.

$$P_W(\{A_1, \dots, A_p\}) = \frac{1}{\mathcal{E}} \sum_{e \in \mathcal{E}} I_C(e)$$

Plus $P_W(\{A_1, \dots, A_p\})$ est élevé, plus les exemples se distinguent des contre-exemples pour les p attributs A_1, \dots, A_p . Ainsi lors de la sélection des attributs discriminants, il s'agira de maximiser le critère P_W . A noter que ce critère n'est pas symétrique par rapport aux exemples et aux contre-exemples.

8.5.2 Algorithme de sélection des attributs discriminants

Notre processus de sélection se différencie des méthodes vues précédemment par la prise en compte des structures hiérarchiques des attributs. Cette structure permet d'établir un ordre de sélection des attributs de type simple.

A un niveau donné, chaque attribut de type structuré détermine par ses composants une partition de l'ensemble des attributs de type simple A_S . En effet, les feuilles d'un composant forment une partie de A_S . Par exemple sur la figure 8.13, l'attribut amphidisque détermine la partition suivante :

$$\{ \{ \text{localisation} \}, \{ \text{taille} \}, \{ \text{densité} \}, \{ \{ \text{tuberculée} \}, \{ \text{oblique} \} \}, \\ \{ \{ \text{longueur relative} \}, \{ \text{allongement} \} \}, \{ \{ \text{bord} \}, \{ \text{nombre} \} \} \}$$

Pour chaque niveau, on recherche les attributs les plus discriminants pour chaque composant (c'est à dire à un niveau supérieur) pour ensuite sélectionner parmi ceux qui sont retenus la meilleure sélection du niveau.

L'algorithme décrit ci-dessous, effectue la sélection des attributs en profondeur puis en largeur le long de la hiérarchie. Cet algorithme utilise l'objet *structure*, représentant les attributs de type structuré, composé des champs suivants :

- les *feuilles* et les *nœuds*, qui correspondent respectivement à la liste de ses attributs de type simple et de ses attributs de type structuré,

- le *modèle* qui désigne la liste des attributs de type simple les plus discriminants pour la sous-hiérarchie issue de l'objet.

Pour un objet *structure* A , l'algorithme se compose de trois étapes.

- 1 - Rechercher le *modèle* constitué des attributs les plus discriminants pour tous les *nœuds* de l'attribut structuré A .
- 2 - Choisir parmi les branches de l'attribut A , c'est à dire les *feuilles* et les *modèles* issues des *nœuds*, celle qui maximise le critère P_W et l'ajouter au *modèle* de A .
- 3 - Recommencer l'étape - 2 - tant que le pouvoir discriminant P_W augmente d'au moins ε .

La fonction d'évaluation du pouvoir discriminant d'un *modèle*, effectuée pour chaque attribut du modèle $|\mathcal{E}| \cdot |\mathcal{C}_{\mathcal{E}}|$ calculs de distance. Ce qui fait pour chaque attribut de domaine D une complexité de l'ordre de $o(|\mathcal{E}| \cdot |\mathcal{C}_{\mathcal{E}}| \cdot |\mathcal{D}|)$ opérations élémentaires. Ainsi, comme pour les méthodes de l'Analyse des données, l'ajout d'un attribut a une complexité de l'ordre du carré du nombre d'observations.

Lorsque ce nombre est trop grand, le calcul du pouvoir discriminant devra se faire sur un échantillon de n observations représentatifs du concept $C = (\mathcal{E}, \mathcal{C}_{\mathcal{E}})$. L'avantage par rapport aux méthodes précédentes est de se limiter à chaque étape aux attributs composant le *nœud* traité. Cette manière de procéder se justifie dans l'hypothèse où, la structure donnée par l'expert pour les attributs est significative.

L'algorithme et ses effets seront illustrés dans l'application sur les éponges. La variable ε définit un seuil minimal de contribution pour chaque nouvel attribut par rapport au modèle. Dans certains cas, l'existence de ce seuil permet d'obtenir un modèle moins large où la contribution de chaque attribut est plus significative.

Chapitre 9

Le système DFI

Sommaire

9.1	Présentation du système	113
9.1.1	Le traitement des données et des connaissances	114
9.1.2	Sélection des attributs les plus discriminants	115
9.1.3	La discrimination	115
9.2	Les paramètres d'entrée	116

Le système DFI est un système de discrimination conceptuelle. Ce système présente l'intérêt de prendre en compte en entrée des données imprécises et structurées et de découvrir des concepts décrits de manière imprécise. Ce système a été développé dans le langage orienté objets (C++).

Nous présentons dans ce chapitre le système DFI en rappelant les différents paramètres qui sont utilisés lors des expérimentations.

9.1 Présentation du système

Le système DFI est composé principalement de 3 modules :

- le module de traitement des données,
- le module de recherche d'un modèle constitué d'attributs discriminants,
- le module de discrimination.

Ces modules seront présentés dans les paragraphes suivants.

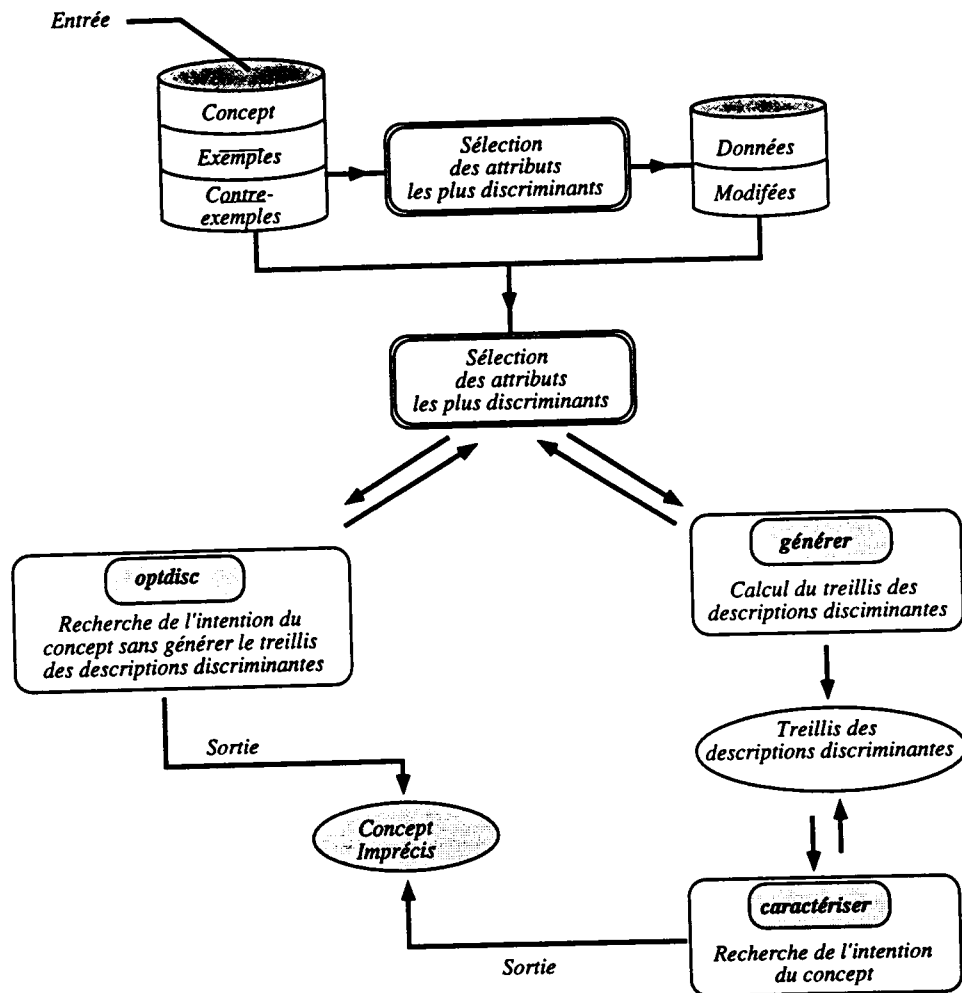


FIG. 9.1 – Le système DFI

9.1.1 Le traitement des données et des connaissances

Ce module se divise en deux parties. La première consiste à saisir le modèle et dans la deuxième les données seront prises en compte.

Saisie du modèle descriptif des données

Les attributs simples et structurés sont décrits par l'utilisateur dans un fichier où sont spécifiés :

- les nuances définies par leurs relations de couverture.

nuance (Croyances,4) {⊥, 1, 0, T} {0110, 0001, 0001, 0000}

Les relations de borne supérieure et de borne inférieure sont ensuite représentées sous forme de tableau pour le système.

- les domaines des attributs sont décrits par leurs valeurs,
 - domaine (Longueur tignules,3) { courts[0-6] , moyens[6-12] , long[12-20] }*
 - domaine (Forme corps,5) { subcylindrique , renflée , évasée , aplatie , mal-définie }*
 - domaine (Pédoncule,1) { torsadé }*
- les attributs sont représentés comme un couple (domaine,nuance).
 - attribut (longueur tignules (Longueur tignules,Croyances));*
 - attribut (forme corps (Forme corps,Croyances));*
 - attribut (pédoncule (Pédoncule,Croyances));*

La structure des attributs est spécifiée séparément et permet d'attribuer les poids des attributs intervenant dans l'évaluation des similarités.

Enregistrement des données relatives au concept

Les données sont décrites par l'utilisateur dans un autre fichier comme une liste de couples (valeur,nuance) décrites sur tous les attributs de type simple. Par exemple, voici un extrait de cas pour les éponges du genre *Hyalonema* :

```
cas1 =  pédoncule::(torsade:1);
        forme corps::(subcylindrique:0) (renflée:1) (évasée:0) (aplatie:0) (mal-
        définie:0);
        longueur tignules::(courts[0-6]:T) (moyens[6-12]:T) (long[12-20]:T);
```

Les classes d'exemples et de contre-exemples sont ensuite précisées en extension par la liste des observations qui les composent.

9.1.2 Sélection des attributs les plus discriminants

Dans ce module, les attributs les plus discriminants sont sélectionnés pour le concept $C = (\mathcal{E}, C_{\mathcal{E}})$ selon l'algorithme du paragraphe 8.5 et déterminent un nouveau modèle des observations. Ensuite, le poids de chaque attribut retenu est évalué pour ce modèle. Comme on l'a précisé précédemment, ces poids interviennent dans le calcul des mesures de similarité.

9.1.3 La discrimination

Le module de discrimination implante les méthodes présentées dans le chapitre précédent. Les données en entrée de ce module sont décrites sur le modèle restreint aux attributs les plus discriminants.

En sortie, on obtient la classe \mathcal{H} décrite en intention, "approximant au mieux" le concept $C = (\mathcal{E}, C_{\mathcal{E}})$. Cette classe est caractérisée par sa qualité de discrimination $G(\mathcal{H})$ dont on rappelle la définition dans le paragraphe suivant. Le système fournit en option

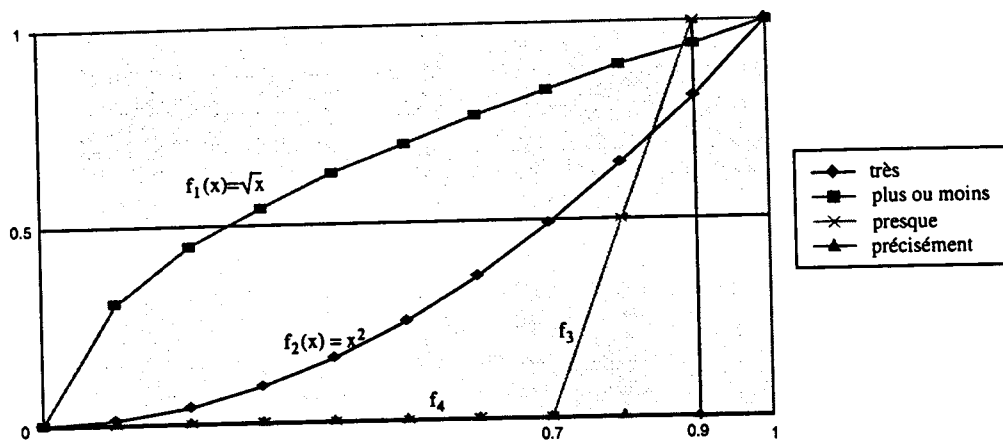


FIG. 9.2 - Exemples de modificateurs

une description de la classe \mathcal{H} par son extension floue sur l'ensemble des observations.

Par exemple, la description 7 du concept \mathcal{H}_4 décrite dans l'application, concerne les individus dont :

- la membrane criblée est inapplicable et ,
- la face exhalante possède 4 orifices et,
- le diamètre de la microoxy-exhactines1 est petit ou moyen et,
- l'amphidisque1 a une petite taille.

9.2 Les paramètres d'entrée

Plusieurs paramètres sont nécessaires pour la mise en œuvre du système DFI tels que les indices similarités, les modificateurs, les quantificateurs qui interviennent dans la déterminations de :

- la qualité de recouvrement
- le calcul du treillis de descriptions discriminantes
- l'évaluation des fonction de reconnaissance.

• Qualité de recouvrement $G(\mathcal{H})$ d'un concept $C = (\mathcal{E}, C_{\mathcal{E}})$ par le concept \mathcal{H} (. 8.2)
 Les paramètres intervenant dans le calcul de la fonction G sont le degré de recouvrement V , le modificateur m et les quantificateurs linguistiques Q_1 et Q_2 , dont voici une description : - Le degré de recouvrement V mesuré soit à l'aide du recouvrement par coupe, soit par la similarité issue de la distance-bipartite (. 7.3).

Le recouvrement par coupe augmente l'influence des éléments appartenant bien aux classes dans l'évaluation de la complétude et de la consistance, alors que la similarité est plus sensible au contour imprécis des différentes classes.

- Le modificateur d'affaiblissement m peut prendre comme fonction-valeur "*plus ou moins*", "*presque*" ou encore la fonction identité.

Comme le montre la figure 9.2, l'utilisation d'un modificateur d'affaiblissement a pour conséquence d'élargir le recouvrement de la classe \mathcal{H} . Le modificateur "*plus ou moins*" augmente le recouvrement pour toutes les coupes de la classe alors que le modificateur "*presque*" ignore les coupes les plus faibles.

- Les quantificateurs linguistiques Q_1 et Q_2 sont éventuellement les couples de valeur "*la plupart*" et "*peu de*" ou encore "*tout à fait*" et "*aucun*".

Les quantificateurs "*la plupart*" et "*peu de*" permettent de diminuer l'influence d'erreurs éventuellement présentes dans les données. L'utilisation de ces quantificateurs augmente la robustesse du système.

Lorsque les classes d'exemples et de contre-exemples sont précises, il est préférable de définir la fonction G à partir du recouvrement par coupe, car ce recouvrement diminue l'importance des contre-exemples à faible appartenance. Par ailleurs, le modificateur "*presque*" aura pour effet de *préciser* les contours de la classe \mathcal{H} et facilite la comparaison avec des ensembles d'exemples et de contre-exemples.

- Le treillis des descriptions α -discriminantes

Le seuil α et le modificateur de validité sont utilisés pour définir une description α -discriminantes.

- Le modificateur m définit la propriété d'être m -valide pour une description α -discriminante et prend comme fonction-valeur "*plus ou moins*", "*presque*", "*tout à fait*", "*très*".

Rappelons que les modificateurs utilisés ici doivent être strictement croissants sur $[0, 1]$ afin de conserver la propriété 8.1.1.

- Le seuil α est une valeur entière qui détermine le nombre de contre-exemples qu'une description peut reconnaître.

La valeur de ce paramètre sera fonction du contexte $(\mathcal{O}, \mathcal{M})$ et surtout du concept C .

Lorsque α est élevé, la qualité de discrimination des fonctions de reconnaissance diminue, mais en contrepartie le nombre de descriptions discriminantes disponibles augmente. De telles descriptions permettent en présence de bruit de caractériser les exemples par des concepts plus concis, c'est à dire composés de peu de descriptions.

- Les fonctions de reconnaissance χ des concepts sont définies à partir soit de la distance D_S , soit de la similarité S .

Chapitre 10

Applications

Sommaire

10.1 Application sur les Hyalonema	119
10.1.1 La représentation des éponges	119
10.1.2 Sélection des attributs discriminants	121
10.1.3 Caractérisation de la classe des <i>Cyliconema</i>	122
10.2 Application sur les Votes	126
10.2.1 Les votes	126
10.2.2 Du numérique au symbolique	126
10.2.3 Caractérisation des classes	129

10.1 Application sur les Hyalonema

10.1.1 La représentation des éponges

Le modèle de la figure 8.13 issu des travaux de Conruyt [10], décrit des éponges marines appartenant aux genre *Hyalonema*. Après élimination des attributs les moins représentatifs, le modèle de représentation de données utilisé dans notre expérimentation est décrit sur la figure 10.1.

L'attribut structuré *description Hyalonema* caractérise les observations du modèle. Cet attribut est composé de 13 attributs structurés et 30 attributs de type simple. Le domaine de valeurs possibles de ces derniers sont grisés sur la figure 10.1.

Pour tous les domaines, les valeurs sont qualifiées par le treillis de croyance V_1 (fig. 4.1) présenté au chapitre 4.

Les éponges du genre *Hyalonema* sont classées en 9 sous-genres comme le montre la figure 10.2. Dans notre expérimentation, le concept $C = \{\mathcal{E}, C_{\mathcal{E}}\}$ à caractériser est

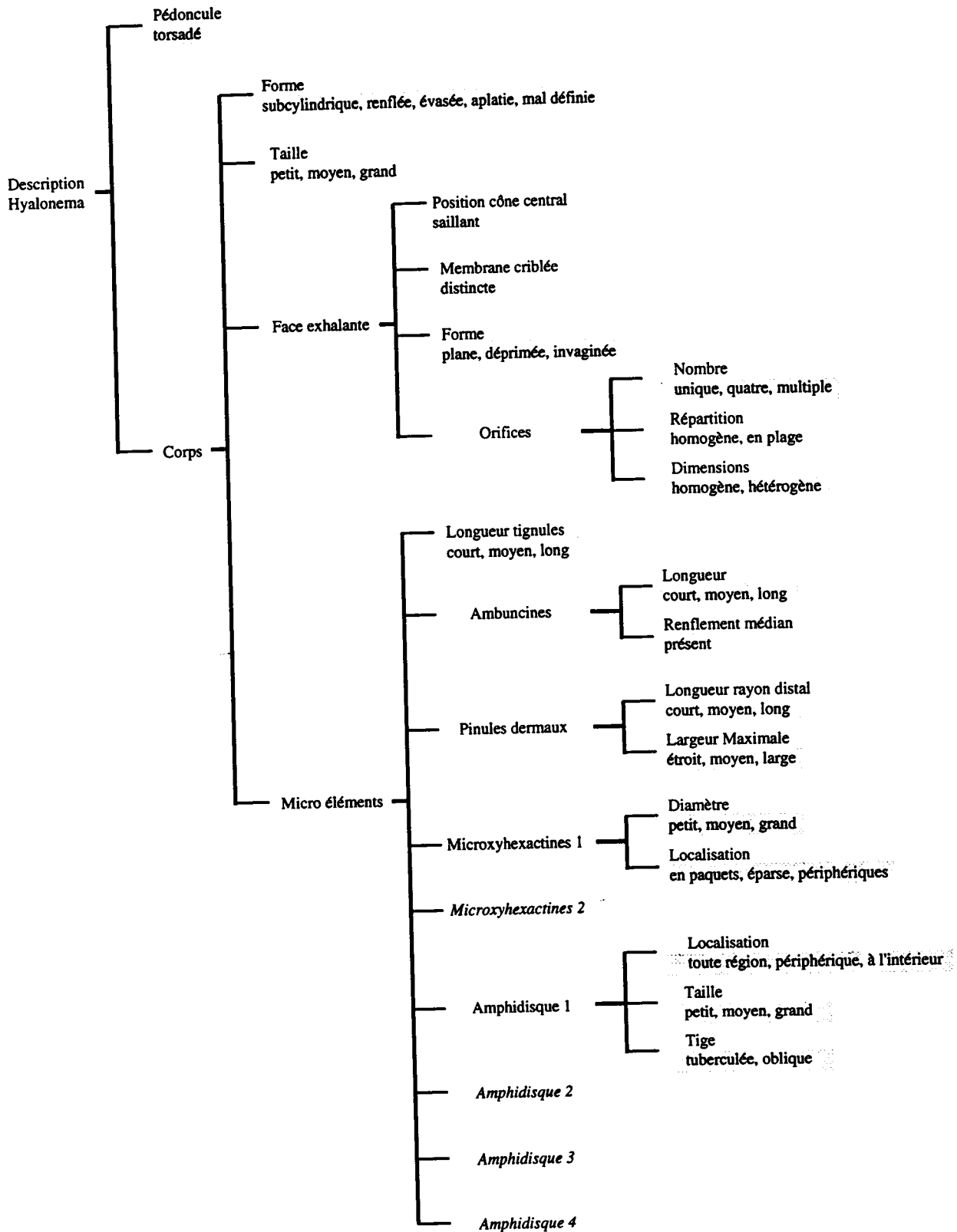


FIG. 10.1 - Le modèle de l'expérimentation

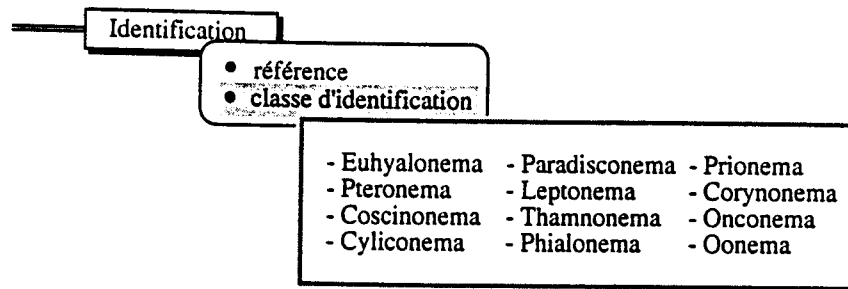


FIG. 10.2 - Les différentes classes de Hyalonema

composé :

- de 16 individus du sous genre *Cyliconema* constituant l'ensemble des exemples \mathcal{E} , et
- de 23 éponges appartenant de manière aléatoire aux autres sous-genres formant l'ensemble des contre-exemples $C_{\mathcal{E}}$.

Les quantificateurs du critère G sont "la plupart" pour la complétude et "peu de" pour la consistance.

10.1.2 Sélection des attributs discriminants

La table suivante indique le nombre de descriptions obtenues à partir des 16 exemples composant le treillis α -discriminantes. Les paramètres décrits dans le chapitre précédent sont le modificateur de validité pour la fonction de reconnaissance et le seuil α de contre-exemples maximum que peut reconnaître une description.

Type	modificateur	seuil α		
		0	1	2
Précis	<i>tout à fait</i>	800	1149	1300
Imprécis	<i>presque</i>	6	9	15

Cette table illustre deux cas extrêmes justifiant l'utilisation d'une procédure de recherche d'attributs les plus discriminants. En raison du nombre élevé d'attributs, nous remarquons que :

- pour le modificateur *tout à fait*¹ la méthode de discrimination proposée est inefficace, car le treillis engendré pour 16 observations est trop important.
- pour le modificateur *presque*, l'effet est inverse, le treillis des descriptions discriminantes est insuffisant pour permettre de choisir les descriptions. En effet, comme on traite d'un même genre d'éponges, la présence d'attributs peu discriminants pour le concept influe

1. L'appartenance définie à partir du modificateur *tout à fait* pour les fonctions de reconnaissance correspond à la relation "est inférieur à".

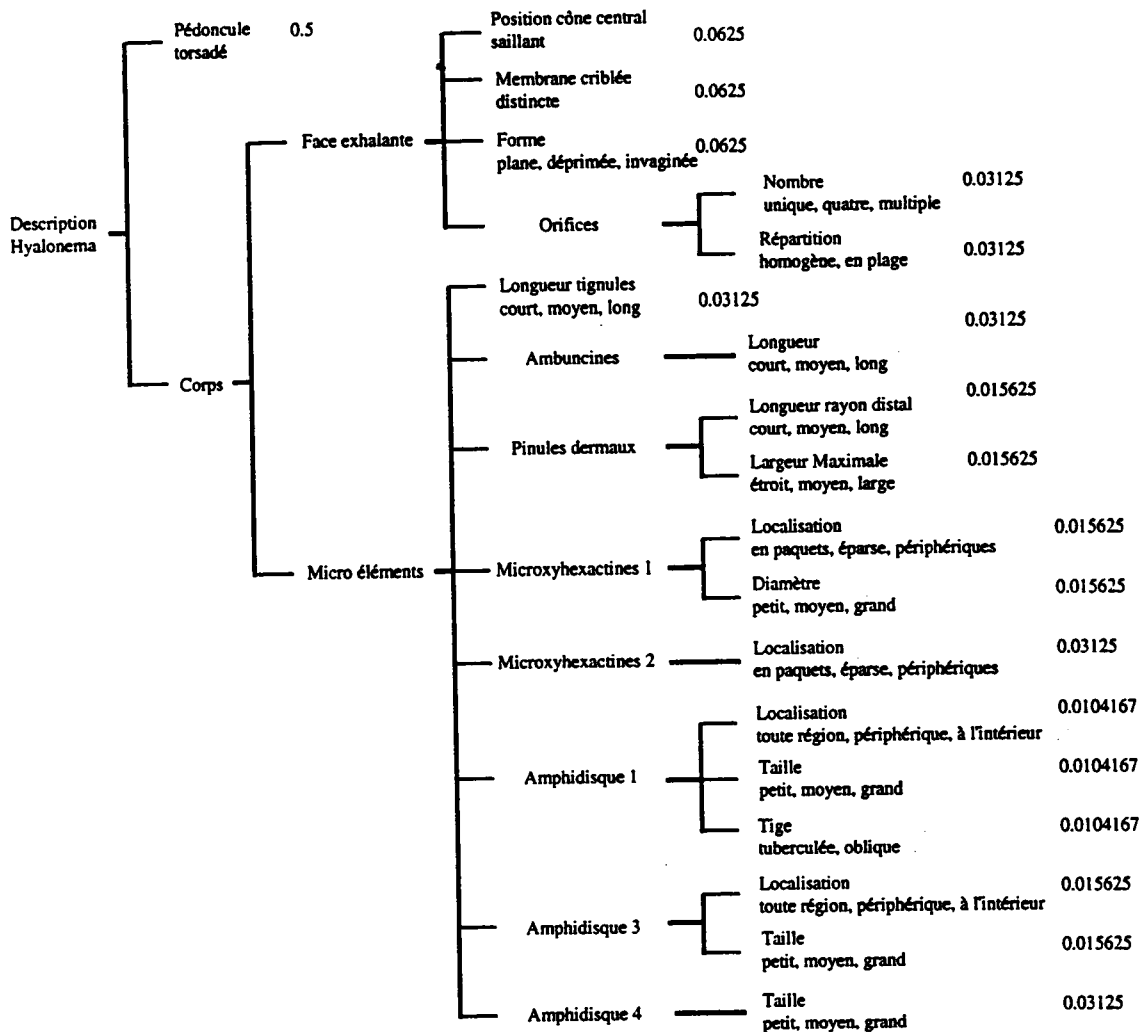


FIG. 10.3 - Modèle à 19 attributs

sur l'évaluation de la similarité.

Les figures 10.3 et 10.4 représentent respectivement les attributs *les plus* discriminants pour un seuil $\varepsilon = 0$ et $\varepsilon = 5\%$. Ces figures indiquent en plus, le poids de chaque attribut de type simple calculé en fonction de la structure modifiée.

10.1.3 Caractérisation de la classe des *Cyliconema*

• Exemple de caractérisation sur le modèle à 19 attributs

Les concepts \mathcal{H}_1 et \mathcal{H}_2 dont les intentions sont décrites sur les figures 10.5 et 10.6 ont été obtenus respectivement pour les paramètres $\alpha = 0$ et $\alpha = 2$ avec une qualité de recouvrement $G = 1$ et $G = 0.9$. Pour ces deux concepts, le modificateur "tout à fait" est utilisé pour mesurer d'une part la validité de la reconnaissance et d'autre part la

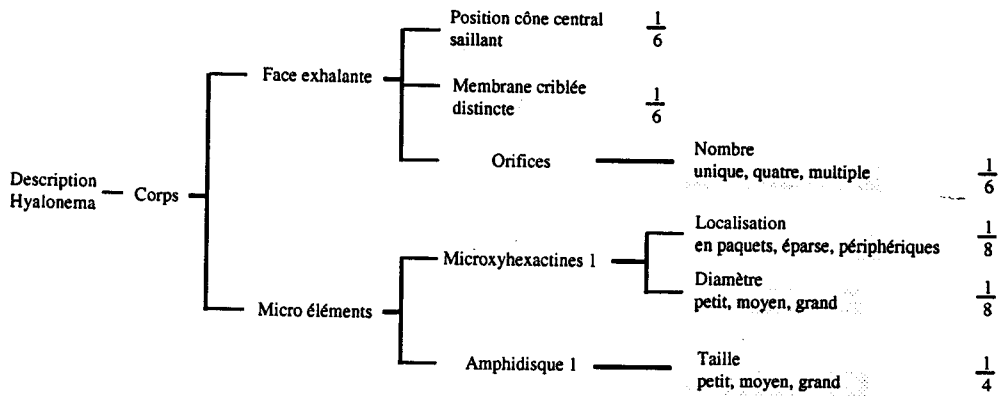


FIG. 10.4 - Modèle à 6 attributs

qualité de recouvrement.

Sur ces deux figures, les attributs dont la valeur est inconnue pour toutes les descriptions des concepts ne sont pas représentés. Le nombre élevé de valeurs inconnues pour ces descriptions (> 10) explique la mauvaise qualité des résultats, lorsque le modificateur "presque" est utilisé au lieu du modificateur "tout à fait". En effet, la

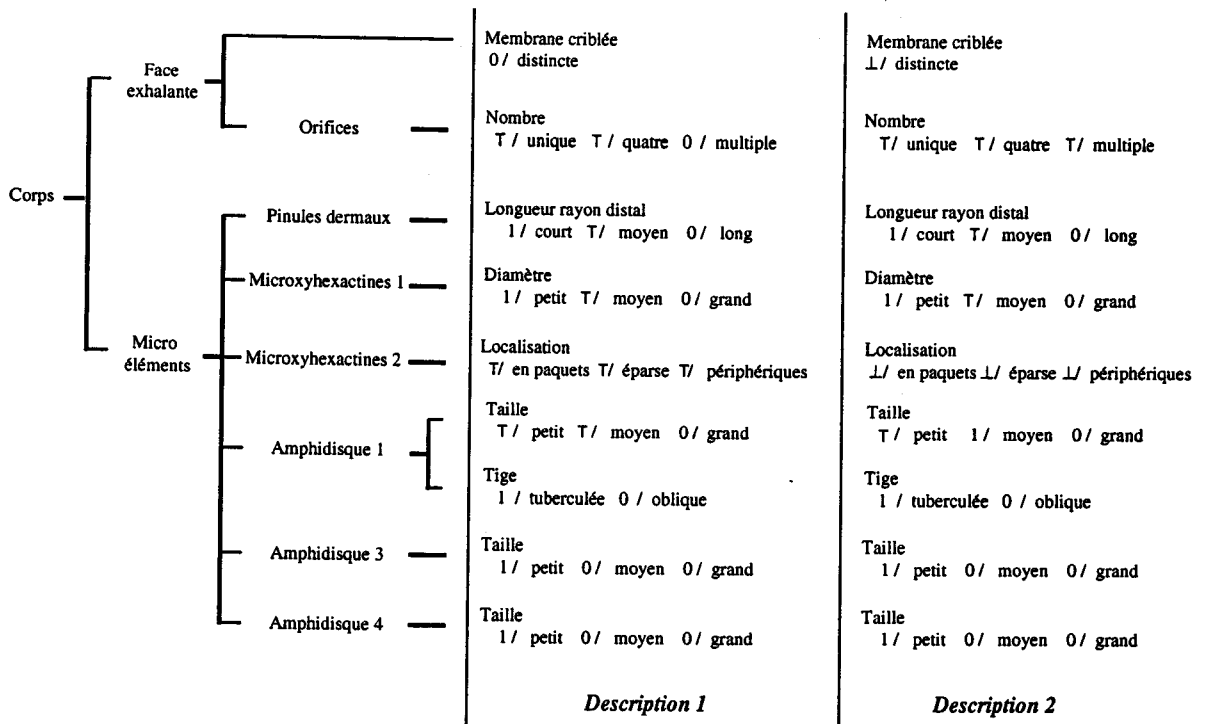


FIG. 10.5 - Intention du concept \mathcal{H}_1

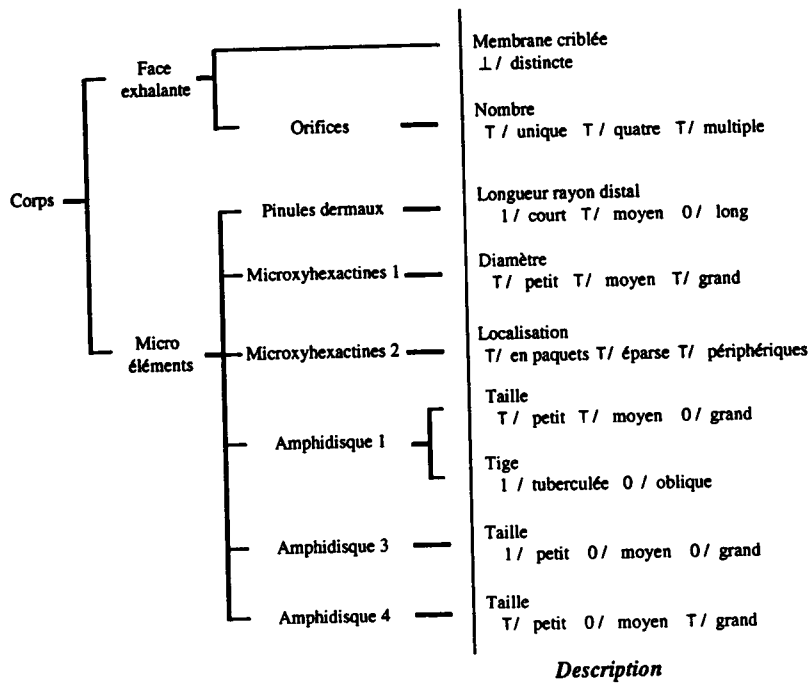


FIG. 10.6 – Intention du concept \mathcal{H}_2

restriction au modèle à 19 attributs ne permet pas de diminuer de manière significative la similarité moyenne des exemples et des contre-exemples.

L'utilisation de la version optimisée de l'algorithme vue au chapitre précédent permet d'obtenir une caractérisation formée de 3 descriptions et de qualité G maximale avec un temps de calcul très inférieur selon les cas (4 fois inférieur pour $\alpha = 0$ et 16 fois inférieur pour $\alpha = 2$).

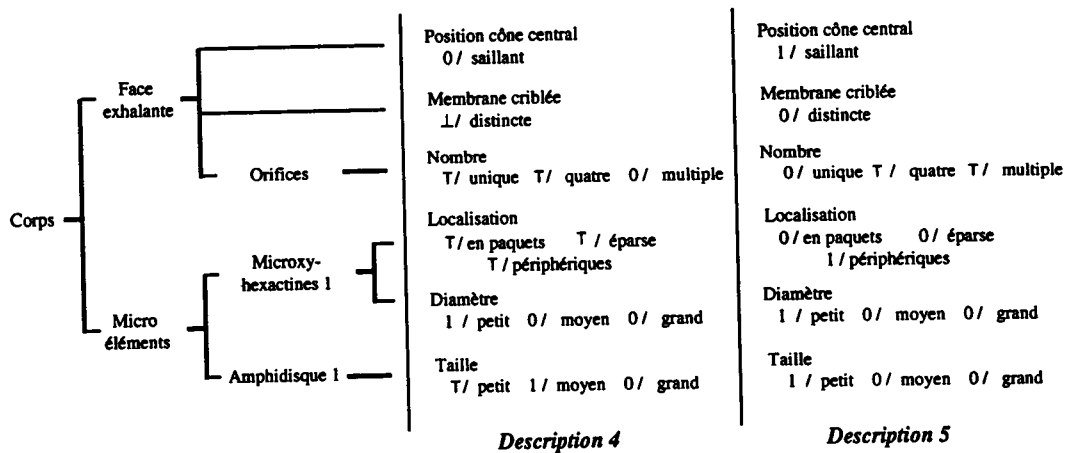


FIG. 10.7 – Intention du concept \mathcal{H}_3

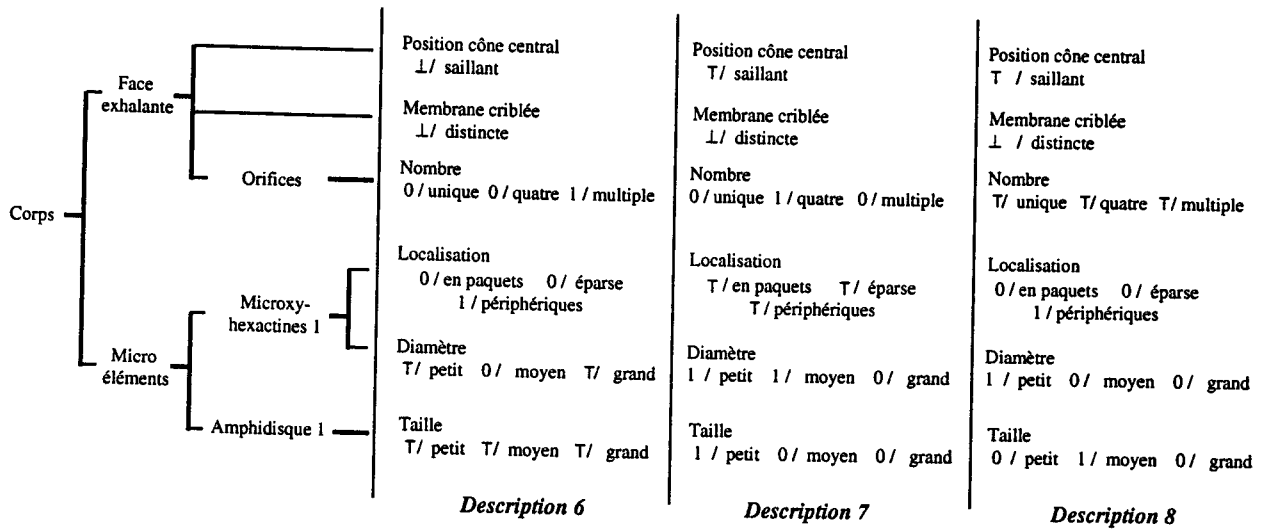


FIG. 10.8 - Intention du concept \mathcal{H}_4

• Exemple de caractérisation sur le modèle à 6 attributs

Les concepts \mathcal{H}_3 et \mathcal{H}_4 sont les résultats des deux algorithmes appliqués au modèle à 6 attributs avec comme paramètres $\alpha = 2$ et $m = \mu = \text{“presque”}$. Le concept \mathcal{H}_4 est issu de l’algorithme optimisé et possède une qualité G de recouvrement de 1 contre 0.91 pour le concept \mathcal{H}_3 .

Ainsi pour \mathcal{H}_4 , les résultats s’interprètent par :

- La plupart des exemples sont presque reconnus par le concept \mathcal{H}_4
- Peu de contre-exemples sont presque reconnus par le concept \mathcal{H}_4

La figure 10.9 donne un aperçu de l’extension des concepts \mathcal{H}_3 et \mathcal{H}_4 sur l’ensemble des observations relatives au concept $C = (\mathcal{E}, C_{\mathcal{E}})$.

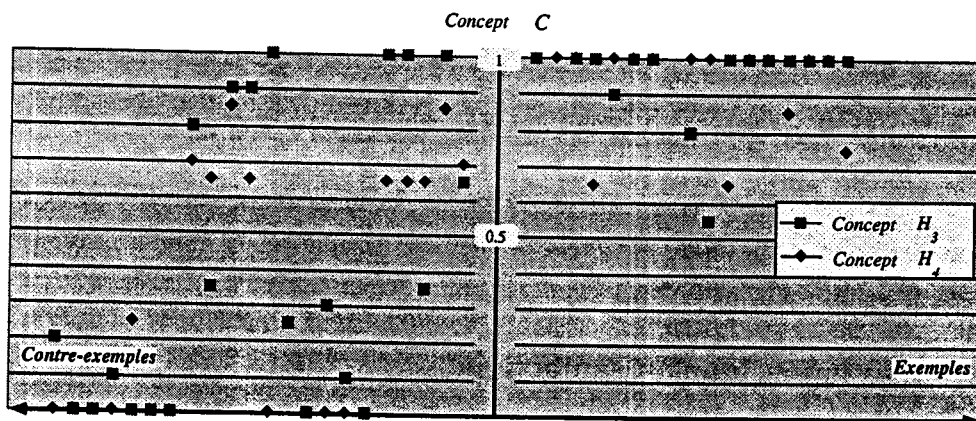


FIG. 10.9 - Extension des concepts \mathcal{H}_3 et \mathcal{H}_4

10.2 Application sur les Votes

10.2.1 Les votes

Les *Votes* sont une base de données numériques, concernant les élections européennes de 1984 et législatives de 1986. Chaque vecteur de données correspond aux pourcentage de suffrages exprimés pour un département sur un ensemble de 10 attributs, regroupant les principales tendances politiques pour les années 1984 et 1986.

Exemple 10.2.1

	<i>pc84</i>	<i>ps84</i>	<i>dr84</i>	<i>ere</i>	<i>vert</i>	<i>fn84</i>	<i>ps86</i>	<i>pc86</i>	<i>dr86</i>	<i>fn86</i>
<i>Ain</i>	7.18	20.66	45.94	3.48	3.17	12.53	32.13	5.57	50.93	10.13
<i>Aisne</i>	15.32	19.42	38.46	2.96	3.75	9.68	34.42	16.65	39.12	9.18
<i>Allier</i>	20.7	19.25	38.78	2.96	3.06	7.67	26.51	22.22	39.52	6.58
<i>Alpes H.P</i>	14.16	22.91	37.57	3.23	3.74	12.19	30.83	14.2	42.15	10.59
<i>Hte Alpes</i>	10.22	21.07	47.02	2.89	3.55	8.60	33.77	6.90	45.0	6.81
<i>Alpes M</i>	11.17	15.7	42.77	2.48	2.41	21.39	23.51	8.61	44.01	20.88
<i>Ardeche</i>	11.84	21.46	46.11	2.59	3.09	8.32	33.19	9.97	48.56	8.36

Soit Ω un ensemble de n individus $\Omega = \{w_1, \dots, w_n\}$ décrits sur p attributs $X = \{X_1, \dots, X_p\}$. On note $X_j(w_i)$, la $j^{\text{ème}}$ composante de l'individu w_i . Chaque attribut X_j est considéré comme une variable numérique $X_j = \{x_1^j, \dots, x_n^j\}$ où $x_i^j = X_j(w_i)$. Le tableau numérique $(x_i^j)_{\substack{1 \leq j \leq p \\ 1 \leq i \leq n}}$ représente ainsi la base de données.

Ces données sont partitionnées en quatre classes par la méthode des nuées dynamiques dans le cas des centres de gravité. Ces classes ont été obtenues sur le logiciel SICLA à l'aide de la commande MNDQAN.

On appelle $P = (P_1, \dots, P_l, \dots, P_k)$ une partition de l'ensemble des observations et $L = (L_1, \dots, L_l, \dots, L_k)$ sa représentation, où L_l correspond au centre de gravité de P_l . La commande MNDQAN cherche le couple (P, L) minimisant le critère d'adéquation W , qui s'écrit :

$$W(P, l) = \sum_{l=1}^k \sum_{x_i \in P_l} p_i d_M^2(x_i, L_l) = \sum_{l=1}^k \sum_{x_i \in P_l} p_i \frac{(x_i^j - m_j)^2}{\sigma_j^2}$$

où d_M est la distance normalisée de \mathbb{R}^p et m_j est la moyenne de la variable X_j .

Il est montré que la méthode recherche le meilleur couple (P, L) , maximisant l'inertie inter-classe et minimisant l'inertie intra-classe.

10.2.2 Du numérique au symbolique

Pour coder de manière symbolique des données numériques, l'idée est de déterminer une partition floue du domaine de valeurs de l'attribut X_j . Dans le cas des votes, les

valeurs sont des pourcentages. Le domaine des valeurs possibles pour un attribut se ramène à l'intervalle $[0, 100]$.

Le choix des méthodes est suggéré par l'hypothèse que la variable X_j suit une loi normale. La partition floue utilisée est composée de trois sous-ensembles flous *Inf*, *Moy*, *Sup* obtenus par l'étude de l'ensemble des valeurs de la variable X_j , restreintes aux intervalles $I = [0, m - \sigma]$, $M =]m - \sigma, m + \sigma[$ et $S = [m + \sigma, 100]$ où m est la moyenne de X_j et σ son écart-type.

m_S , m_M , m_I sont les moyennes respectives des ensembles de valeurs $X_j \cap I$, $X_j \cap M$, $X_j \cap S$ et σ_S , σ_M , σ_I sont leurs écart-types.

Comme le montre la figure 10.10, les sous-ensembles *Inf*, *Moy*, *Sup* sont construits comme des sous-ensembles flous de forme trapézoïdale, où :

- $Inf = (0, m_I - \sigma_I, m_I + \sigma_I, m_M - \sigma_M)$,
- $Moy = (m_I + \sigma_I, m_M - \sigma_M, m_M + \sigma_M, m_S - \sigma_S)$,
- $Sup = (m_M + \sigma_M, m_S - \sigma_S, m_S + \sigma_S, 100)$.

Lorsque l'un des intervalles $X_j \cap I$, $X_j \cap M$, $X_j \cap S$ est vide, *Inf*, *Moy* ou *Sup* correspondent respectivement aux valeurs précises 0, m ou 100. A partir de cette partition floue,

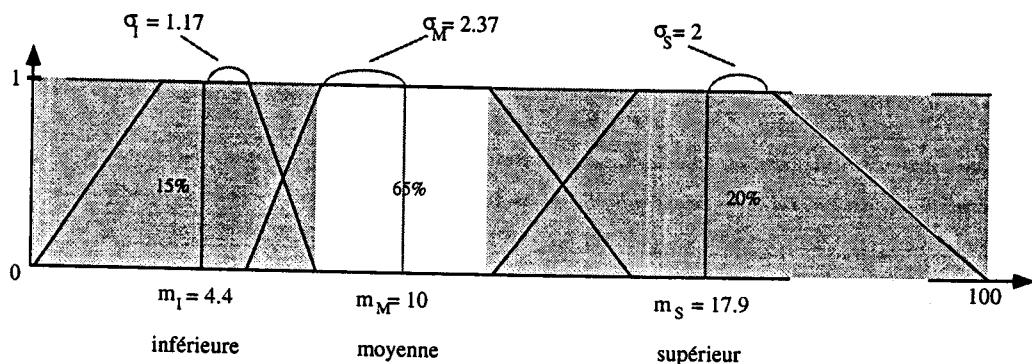


FIG. 10.10 - Etape de fuzzification des valeurs

chaque attribut numérique est codé sous la forme d'un attribut symbolique nuancé de domaine $D = \{Inf, Moy, Sup\}$ et de nuance le treillis d'intensité V_4 (fig. 10.11).

La fonction de transformation t , définie par :

$$\begin{array}{rcl}
 t : & [0, 1] & \longrightarrow & V_4 \\
 & x = 1 & \longmapsto & \text{totalement} \\
 & 0.8 \leq x < 1 & \longmapsto & \text{très} \\
 & 0.2 < x < 0.8 & \longmapsto & \text{plus ou moins} \\
 & 0 < x \leq 0.2 & \longmapsto & \text{peu} \\
 & 0 & \longmapsto & \text{pas du tout}
 \end{array}$$

permet d'obtenir pour chaque individu w_i , la valeur nuancée de l'attribut X_j :
 $\{t(\mu_{Inf}(x_i^j))/Inf, t(\mu_{Moy}(x_i^j))/Moy, t(\mu_{Sup}(x_i^j))/Sup\}$.

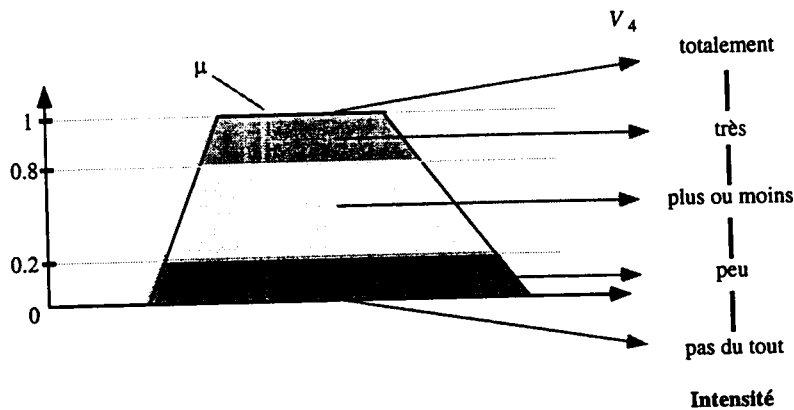
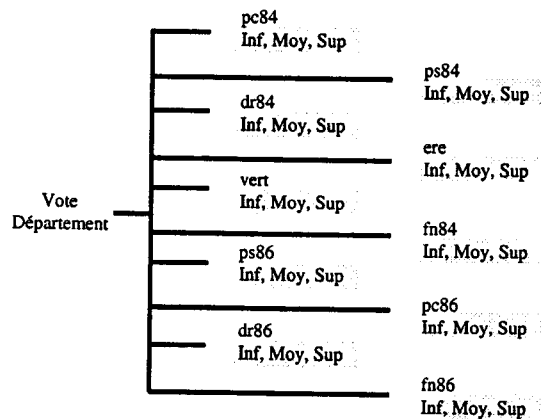


FIG. 10.11 - Du flou au symbolique

En appliquant ces résultats aux départements de l'Ain et de l'Aisne, nous obtenons les descriptions symboliques nuancées suivantes, décrites sur le modèle de la figure 10.12.

Ain

$pc84 \{ \text{peu} / Inf, \text{plus ou moins} / Moy, \text{pas du tout} / Sup \}$
 $ps84 \{ \text{pas du tout} / Inf, \text{totalement} / Moy, \text{pas du tout} / Sup \}$
 $dr84 \{ \text{pas du tout} / Inf, \text{totalement} / Moy, \text{pas du tout} / Sup \}$
 $ere \{ \text{pas du tout} / Inf, \text{totalement} / Moy, \text{pas du tout} / Sup \}$
 $vert \{ \text{pas du tout} / Inf, \text{totalement} / Moy, \text{pas du tout} / Sup \}$
 $fn84 \{ \text{pas du tout} / Inf, \text{plus ou moins} / Moy, \text{plus ou moins} / Sup \}$
 $ps86 \{ \text{pas du tout} / Inf, \text{totalement} / Moy, \text{pas du tout} / Sup \}$
 $pc86 \{ \text{plus ou moins} / Inf, \text{plus ou moins} / Moy, \text{pas du tout} / Sup \}$
 $dr86 \{ \text{pas du tout} / Inf, \text{peu} / Moy, \text{plus ou moins} / Sup \}$
 $fn86 \{ \text{pas du tout} / Inf, \text{totalement} / Moy, \text{pas du tout} / Sup \}$

FIG. 10.12 – Le modèle de représentation des *Votes*

Aisne

pc84 { *pas du tout / Inf, peu / Moy, très / Sup* }
ps84 { *pas du tout / Inf, totalement / Moy, pas du tout / Sup* }
dr84 { *plus ou moins / Inf, plus ou moins / Moy, pas du tout / Sup* }
ere { *pas du tout / Inf, totalement / Moy, pas du tout / Sup* }
vert { *pas du tout / Inf, plus ou moins / Moy, plus ou moins / Sup* }
fn84 { *pas du tout / Inf, totalement / Moy, pas du tout / Sup* }
ps86 { *pas du tout / Inf, totalement / Moy, pas du tout / Sup* }
pc86 { *pas du tout / Inf, pas du tout / Moy, totalement / Sup* }
dr86 { *plus ou moins / Inf, plus ou moins / Moy, pas du tout / Sup* }
fn86 { *pas du tout / Inf, totalement / Moy, pas du tout / Sup* }

Contrairement au cas des Hyalonema, ces données ne possèdent pas de structure complexe.

10.2.3 Caractérisation des classes

Sélection des attributs discriminants

Lors de l'expérimentation, la classe à caractériser désigne l'ensemble des exemples et est en opposition avec les autres classes dont l'union forme l'ensemble des contre-exemples. Ainsi, chaque classe dispose de sa propre sélection d'attributs les plus discriminants.

La table ci-dessous présente les attributs les plus discriminants par rapport à sa

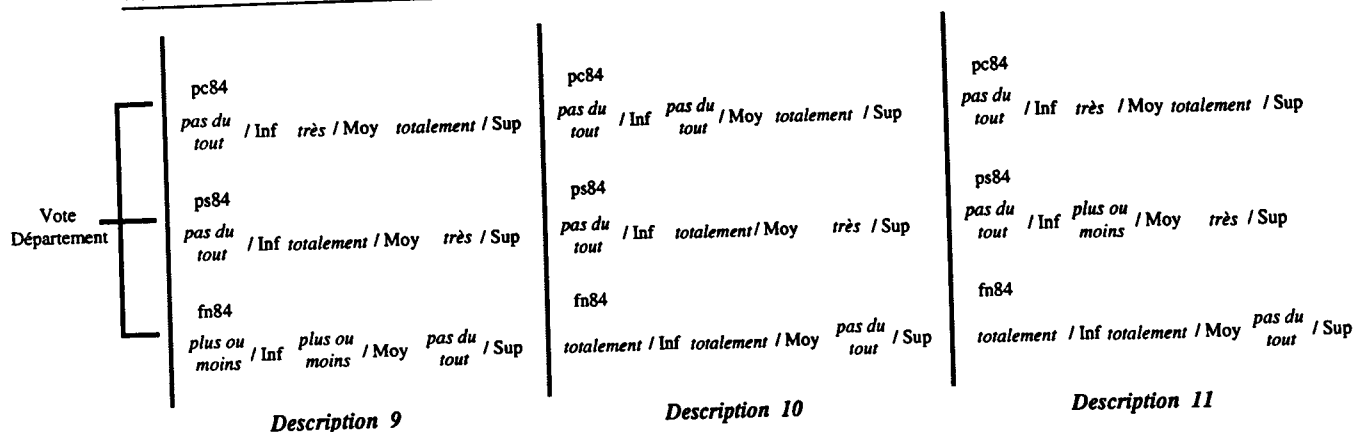


FIG. 10.13 – Intention du concept \mathcal{H}'_4

classe, déterminé pour un seuil $\varepsilon = 5\%$.

	pc84	ps84	dr84	ere	vert	fn84	ps86	pc86	dr86	fn86
1		■			■	■		■		
2					■		■	■		■
3			■			■				
4	■	■	■							

Exemples de caractérisations

Les classes C_1, C_2, C_3, C_4 sont respectivement caractérisées par les concepts $\mathcal{H}'_1, \mathcal{H}'_2, \mathcal{H}'_3$ et \mathcal{H}'_4 . Ces derniers ont été obtenus par l'algorithme optimisé avec comme paramétrage $\alpha = 2$ et $m = \mu = \text{“presque”}$.

Tous ces concepts possèdent une qualité de recouvrement maximale de 1, à l'exception de \mathcal{H}'_3 dont la qualité est de 0.99. Ces concepts recouvrent la plupart (Q_1) des exemples et peu (Q_2) de contre-exemples.

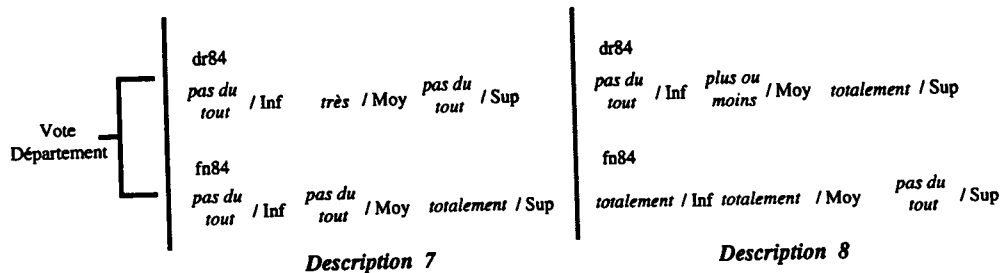


FIG. 10.14 – Intention du concept \mathcal{H}'_3

Notons que dans le cas de la classe 3, les algorithmes ne trouvent aucun concept valable à partir des descriptions discriminant tous les contre-exemples. Par contre,

l'utilisation de règles moins strictes, telles que celles qui reconnaissent au plus 2 contre-exemples, permet de prendre en compte la présence d'erreurs dans la classe et de déterminer un concept de qualité.

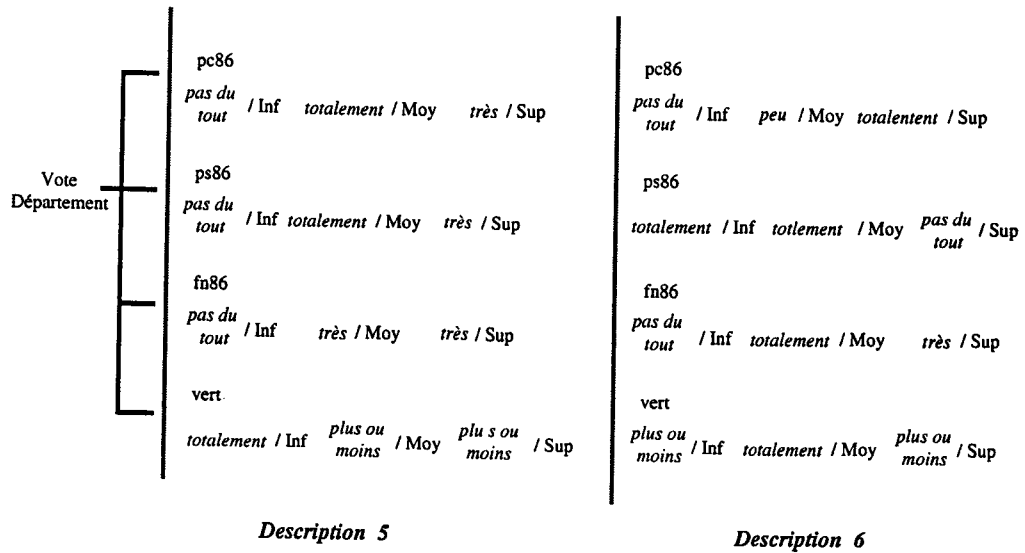
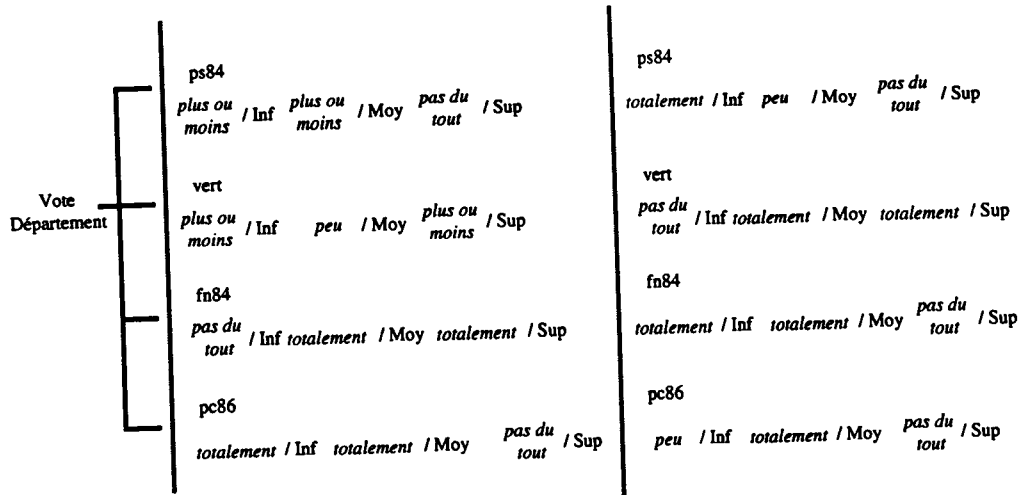


FIG. 10.15 – Intention du concept \mathcal{H}'_2

Interprétation

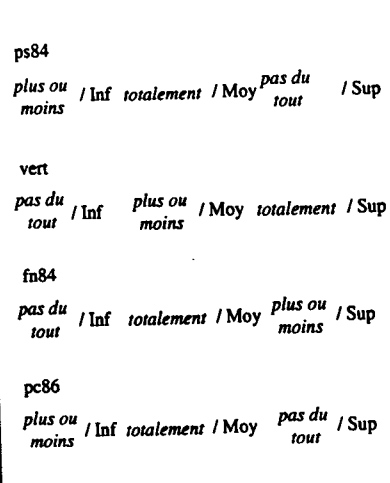
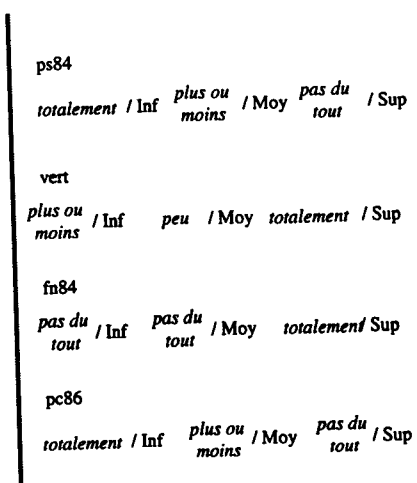
D'après les différents résultats, nous pouvons dire que :

- la majorité des départements appartenant à la classe 1, ont des votes supérieurs à la moyenne nationale pour les *verts* et le *front national* et inférieurs à la moyenne nationale pour le *parti socialiste* aux élections européennes de 1984.
- la classe 2 est caractérisée par un *parti communiste* et un *front national* forts aux législatives de 1986 avec soit un *parti socialiste* fort et des *verts* faibles ou soit un *parti socialiste* faible et des votes situés dans la moyenne pour les *verts*.
- la plupart des départements de la classe 3 sont caractérisés soit par une *droite* forte et un *front national* aux votes inférieurs ou égaux à la moyenne, soit par une *droite* située dans la moyenne et un *front national* fort.
- la classe 4 regroupe les départements ayant une tendance de gauche et un *front national* inférieur ou égal à la moyenne nationale pour les élections de 84.



Description 1

Description 2



Description 3

Description 4

FIG. 10.16 – Intention du concept \mathcal{H}'_1

Conclusion

Dans cette thèse, nous avons développé une méthode de discrimination conceptuelle pour des objets imprécis et structurés. L'élaboration de ce système nous a amené à développer des outils de base pour la représentation des données et des concepts imprécis et nous avons proposé des algorithmes de recherche de fonctions discriminantes pour de tels concepts.

La représentation des connaissances

L'utilisation de termes linguistiques pour nuancer les modalités d'une observation, permet de réduire le biais entre l'observation faite par l'expert et sa description dans le modèle de représentation.

- Dans un premier temps, nous avons muni l'espace des objets structuré de plusieurs métriques, prenant en compte les particularités du modèle :
 - D'une part, nous avons montré que les attributs de type simple disposent de distances telles que les distances de Hamming, Bi-partite et Min-max lorsque les nuances possèdent une structure de treillis modulaire.
 - D'autre part, les distances ou similarités proposées pour les objets structurés, accordent de l'importance à la structure hiérarchique des attributs, soit par l'intermédiaire d'un poids attribué à chaque attribut de type simple en fonction de la structure, soit par l'intermédiaire de la représentation structurelle des données.

L'effet désiré pour ces distances est de diminuer l'influence des attributs les plus profonds dans l'arborescence du modèle. Les travaux sur les métriques ont donné lieu aux publications suivantes [28], [29]. Ces distances vont permettre ensuite de définir des fonctions de reconnaissance associées à une description.

- Dans un deuxième temps, nous avons défini à partir de la notion de concept imprécis, des critères mesurant la qualité de discrimination d'un ensemble de

fonctions discriminantes. L'appartenance d'une observation à un concept peut être graduelle. Ainsi, l'imprécision du concept s'exprime par des frontières floues et dépend de la fonction de reconnaissance utilisée.

Par le biais de modificateurs linguistiques, la qualité de représentation d'un concept peut être précisée, ce qui permet d'augmenter et de contrôler la validité des résultats. Ces modificateurs influent sur les frontières floues d'un concept.

Les méthodes de recherche de concepts

- A partir de ce critère, nous avons élaboré une méthode de discrimination sur le modèle des objets structurés qui se divise en deux étapes :

- Dans la première, l'ensemble des descriptions α -discriminantes est généré à partir des *meilleurs* exemples du concept. Cette procédure utilise lors de la construction, la structure de treillis des descriptions discriminantes afin de limiter la production de règles non-discriminantes.
- La deuxième étape réalise si possible, un recouvrement satisfaisant des exemples à partir des descriptions α -discriminantes. Dans cette étape, la structure d'ordre du treillis détermine la méthode de parcours des descriptions α -discriminantes.

- Ensuite nous avons présenté une optimisation dont le principe consiste à fusionner les deux étapes de la première méthode. Il s'agit à chaque étape de conserver la meilleure couverture des exemples et les descriptions discriminantes qu'elle recouvre. Contrairement à la méthode précédente, celle-ci dépend de l'ordre de généralisation des exemples.

Les applications

L'expérimentation sur les Hyalonema montre que les deux méthodes utilisées, déterminent des fonctions discriminant convenablement les exemples des contre-exemples.

Cependant, l'utilisation de fonction de reconnaissance basée sur les similarités nécessitent de disposer d'un modèle composé d'attributs les plus discriminants. En effet, les éponges de l'expérimentation appartiennent au même genre, les Hyalonema. Aussi, pour le modèle de représentation global, leurs descriptions sont semblables.

Pour pallier à ce problème, nous avons proposé une méthode de sélection des attributs les plus discriminants pour un concept $C = (\mathcal{E}, C_{\mathcal{E}})$.

D'autre part, le modèle sélectionné comporte en général beaucoup moins d'attributs

que le modèle de représentation de données. Ainsi, chaque étape de généralisation, chaque étape d'évaluation du critère de discrimination, . . . sont de moindre complexité.

Les perspectives

- Au niveau logiciel, les travaux de cette thèse se situent dans le cadre du projet IKBS (Iterative Knowledge Base System) [11], qui propose un atelier logiciel permettant d'une part de saisir les informations relatives à un exemple, d'autre part d'acquérir de nouvelles connaissances. Ainsi, il s'agira d'inclure nos différents modules dans le système IKBS.

- Au niveau de la représentation de la connaissance, nous souhaitons faire une étude plus complète de la logique induite par l'utilisation de treillis de nuances. Comme l'ont fait Sallantin [37] ou Ginsberg [14] pour le treillis de croyances, il s'agirait de déterminer les familles d'opérateurs possible en fonction du type de treillis.

- Au niveau des méthodes de recherche de concepts, une direction d'étude à explorer est l'utilisation de l'aspect structurel des données en amont du processus de discrimination. Jusqu'à présent, la structure des données intervient dans le calcul des similarités et dans la procédure de sélection des attributs les plus discriminants.

L'idée est d'utiliser un processus de classification sur les descriptions structurelles. Cette méthode de classification peut se baser sur la distance structurelle ou encore sur la relation d'ordre entre les différents états $A \prec P \prec I$, issue de celle des nuances.

Ainsi, sur chaque classe *structurelle*, nous pouvons appliquer nos procédures de discrimination. La restriction des exemples et contre-exemples pour chaque classe *structurelle* détermine un type de concept local. L'intérêt est de diminuer la complexité des procédures de recherche, car les descriptions *structurelles* possèdent une représentation simplifiées par rapport aux valeurs.

Annexe A

Les opérateurs de généralisation et de spécialisation pour les taxonomies

A.1 La généralisation de taxonomies

A.1.1 Preuve de la proposition 4.3.1

Rappel: L'application $\vee_{[A]}$ de $[A] \times [A]$ dans V_1^D vérifiant,

$$\forall \mu, \nu \in [A], \quad \mu \vee_{[A]} \nu(h) = \begin{cases} \top & \text{si } (\mu \vee \nu)(h) = 1 \text{ et} \\ & \exists h' \in D, \quad h' > h \text{ et } (\mu \vee \nu)(h') = \top \\ (\mu \vee \nu)(h) & \text{sinon} \end{cases}$$

définit la borne supérieure des éléments μ et ν dans $[A]$.

preuve :

- Montrons que pour tout μ, ν de $[A]$, $\mu \vee_{[A]} \nu$ est une hiérarchie nuancée cohérente.

- Soit $h \in D$ tel que $\mu \vee_{[A]} \nu(h) \neq \top$

$$\text{par définition de } \vee_{[A]} \quad \mu \vee_{[A]} \nu(h) = \mu \vee \nu(h) = \begin{cases} 0 \\ 1 \\ \perp \end{cases},$$

donc $\mu(h) \leq \nu(h)$ ou $\nu(h) \leq \mu(h)$ par définition du treillis de nuance V_1 ,
c'est à dire que $\mu \vee_{[A]} \nu(h) = \mu(h)$ ou $\nu(h)$,

si $\mu \vee \nu(h) = \mu(h)$ par définition de \vee et comme $\mu \in [A]$,

- . si $\mu(h) = 0$ ou $\mu(h) = \perp$ alors

$$\forall h' \subset h, \quad \mu(h') \leq \mu(h) \text{ et } \nu(h') \leq \nu(h) \leq \mu(h)$$

$$\text{donc } \mu \vee_{[A]} \nu(h') \leq \mu \vee_{[A]} \nu(h)$$

$$\exists h' \in D \text{ tel que } \mu(h') = \mu(h) \text{ et } \mu \vee_{[A]} \nu(h') = \mu \vee_{[A]} \nu(h)$$

par conséquent $\mu \vee_{[A]} \nu(h) = \nabla_{h' \subset h} \mu \vee_{[A]} \nu(h')$.

. si $\mu(h) = 1$ alors $\exists h' \in D$, $h' \subset h$ tel que $\mu(h') = 1$ ou $\mu(h') = \top$

donc $\mu \vee \nu(h') \geq 1$ par définition de V_1 et,

$\mu \vee_{[A]} \nu(h') \geq \mu \vee \nu(h') \geq 1$ d'où $\nabla_{h' \subset h} \mu \vee_{[A]} \nu(h') = \top$ ou $\nabla_{h' \subset h} \mu \vee_{[A]} \nu(h') = 1$

si $\mu \vee \nu(h) = \nu(h)$, la démonstration est symétrique.

- Soit $h \in D$ tel que $\mu \vee_{[A]} \nu = \top$

par définition de $\vee_{[A]}$,

. soit $(\mu \vee \nu)(h) = 1$ et $\exists h' \in D$ tel que $h' > h$ et $(\mu \vee \nu)(h') = \top$

par définition de V_1 nous avons $\begin{cases} \mu(h) = 1 \text{ et } \nu(h) \leq 1 \\ \text{ou} \\ \nu(h) = 1 \text{ et } \mu(h) \leq 1 \end{cases}$,

si $\mu(h) = 1$ (resp. $\nu(h) = 1$) alors $\exists h'' \in D$ tel que $\mu(h'') = 1$ ou \top (resp. $\nu(h'') = 1$ ou \top) et,

$\forall h'' \in D$ tel que $\mu(h'') = 1$ ou \top , alors $\mu \vee_{[A]} \nu(h'') = \top$ par définition de $\vee_{[A]}$ car $(\mu \vee \nu)(h') = \top$ et $h' > h''$,

donc $\mu \vee_{[A]} \nu(h) = \nabla_{h' \subset h} \mu \vee_{[A]} \nu(h')$.

. soit $(\mu \vee \nu)(h) = \top$ et par définition de V_1 nous avons $\begin{cases} \mu(h) = 1 \text{ et } \nu(h) = 0 & (1) \\ \mu(h) = 0 \text{ et } \nu(h) = 1 & (2) \\ \mu(h) = \top & (3) \\ \nu(h) = \top & (4) \end{cases}$

Dans le cas (1) (resp. (2)), comme $\mu \in [A]$ (resp. $\nu \in [A]$), $\exists h' \in D$ tel que $h \subset h'$ et $\mu(h') = 1$ ou $\mu(h') = \top$ (resp. pour ν), donc $\mu \vee \nu(h') \geq 1$ et par définition de $\vee_{[A]}$ et comme $(\mu \vee_{[A]} \nu)(h) = \top$ alors $\mu \vee \nu(h') = \top$,

et d'après l'opérateur ∇ , $\mu \vee_{[A]} \nu(h) = \nabla_{h' \subset h} \mu \vee_{[A]} \nu(h')$

Dans les cas (3) et (4), $\exists h' \in D$ tel que $\mu \vee \nu(h') = \top$ et,

par définition de $\vee_{[A]}$, $\forall h' \in D$, $\mu \vee \nu(h') = 1 \implies \mu \vee_{[A]} \nu(h) = \top$,

d'où $\mu \vee_{[A]} \nu(h) = \nabla_{h' \subset h} \mu \vee_{[A]} \nu(h')$

• Montrons que $\mu \vee_{[A]} \nu$ est la plus petite hiérarchie nuancée inférieure à μ et ν . En d'autres termes,

$$\forall \theta \in [A], \quad \theta \geq \mu \text{ et } \theta \geq \nu \implies \theta \geq \mu \vee_{[A]} \nu$$

Soient $\theta \in [A]$, $\theta \geq \mu$ et $\theta \geq \nu \implies \theta \geq \mu \vee \nu$

- Si $\mu \vee \nu(h) = 1$ et $\exists h' \in D$ tel que $h' > h$ et $\mu \vee \nu(h') = \top$ alors,

$\theta(h') = \top$, $\theta(h) \geq 1$ et $h' > h$ donc $\theta(h) = \top$ car $\theta \in [A]$,

et $\theta(h) \geq \mu \vee_{[A]} \nu(h)$ par définition de V_1 .

- Sinon $\mu \vee_{[A]} \nu(h) = \mu \vee \nu(h)$ par définition de $\vee_{[A]}$ donc $\theta(h) \geq \mu \vee_{[A]} \nu(h)$.

◇

A.1.2 Procédure de généralisation

Les taxonomies sont implantées comme des objets *palier* dont la structure est semblable à l'objet *nœud* défini au chapitre 8. Cet objet est composé d'un label indiquant une valeur du domaine, de la nuance appliquée à cette valeur ainsi que les informations relatives aux *paliers* fils, c'est à dire le nombre de ses paliers fils et un tableau contenant ses paliers.

classe palier

variables

<i>contenu</i>	[\mathcal{M}]
<i>d</i>	chaîne de caractères
μ	V_1
<i>nb fils</i>	entier
<i>fils</i>	table de palier

méthodes

	généralisation(palier, palier, booléen)
$V_1 \leftarrow$	spécialisation(palier, palier)
	modifier()

A partir de la proposition 4.3.1, nous définissons la procédure de généralisation. x et y désignent les éléments à généraliser et la variable booléenne *Test* indique lorsqu'elle est vraie, qu'un palier père a pour valeur \top . Lors de l'appel initial, la variable *test* doit avoir la valeur *faux*. Les fils sont tous rangés dans le même ordre pour tous les paliers de même niveau.

procédure généralisation(palier x , palier y , booléen *Test*)

$i \leftarrow 0$ entier

$d = x.d$

$\mu = x.\mu \vee y.\mu$

si *Test* =vrai et $\mu = 1$ alors

$\mu = \top$

fin de si

si $\mu = \top$ alors

Test =vrai

fin de si

pour tout $i = 1$ à *nb fils*

 généralisation($x.fils[i]$, $y.fils[i]$)

fin de pour tout

Par définition, une hiérarchie est une structure arborescente où chaque palier possède au plus un palier père. L'appel à la procédure de généralisation est effectuée au plus deux fois pour chaque palier. Cette procédure parcourt la hiérarchie en profondeur puis en largeur en passant par tous les paliers. Par conséquent, la complexité en nombre d'opération \vee est en $o(|D|)$.

A.2 La spécialisation de taxonomies

Nous avons montré dans le chapitre 4 que deux valeurs hiérarchiques cohérentes possèdent une borne inférieure. Toute relation de borne supérieure suffit à définir une borne inférieure, dès qu'il existe un plus petit élément au treillis. Dans ce cas, la borne inférieure de deux éléments u et v est définie à partir de la relation d'ordre, comme le plus grand élément inférieur à u et v . Procéder ainsi ne permet de disposer d'une opération de spécialisation avec une complexité polynomiale. Ainsi, nous allons définir ci-dessous, une procédure de spécialisation avec une complexité du même ordre que la procédure de généralisation.

fonction spécialisation(palier x , palier y) $\rightarrow V_1$

$valeur \leftarrow \perp$ V_1
 $i \leftarrow 0$ entier

$d = x.d$
 $\mu = x.\mu \wedge y.\mu$
si $nb\text{fils} = 0$ **alors**
 retourner μ
fin de si
sinon pour tout $i = 1$ à $nb\text{fils}$
 $valeur = valeur \nabla fils[i] \rightarrow$ spécialisation($x.fils[i]$, $y.fils[i]$)
fin de pour tout
si $valeur \leq \mu$ **alors**
 $\mu = valeur$
sinon si ($\mu = 1$ et $valeur \neq 1$ et $valeur \neq \top$) ou
 ($\mu \neq 1$ et $valeur \neq \mu$) **alors**
 $\mu = \top$ /* initialisation */
 modifier()
fin de si
retourner μ

La complexité en nombre d'opérations \wedge et ∇ est de $|D|$. De même, la fonction **modifier()** est exécutée $|D|$ fois. Chaque palier effectue au plus une fois la fonction modifiée et chaque palier ne peut être appelé au plus une fois par son palier père. En effet, dès que μ vaut \perp la procédure **modifier()** s'arrête. Ainsi la complexité en opérations élémentaires est de l'ordre de $o(|D|)$.

procédure modifier()

si $\mu \neq \perp$ alors
 $\mu = \perp$
 pour tout $i = 1$ à $nb\text{fils}$
 $\text{fils}[i] \rightarrow \text{modifier}()$
 fin de pour tout
 fin de si

Nous devons montrer que cette procédure détermine effectivement la borne inférieure de taxonomies. En d'autres termes, il s'agit de justifier que la taxonomie w obtenue par la procédure de spécialisation sur les taxonomies u et v , est la plus grande des plus petites hiérarchies cohérentes inférieures à u et v .

- Vérifions que w est une hiérarchie inférieure à u et v .

Soit h un palier de D ,

par définition de la procédure de spécialisation, on a :

- $\mu_w(h) = \mu_u(h) \wedge \mu_v(h)$,
- $\mu_w(h) = \nabla_{h' \subset h} (\mu_w(h'))$ si $\nabla_{h' \subset h} (\mu_w(h')) \leq \mu$,
- $\mu_w(h) = \top$ lorsque le test ($\mu = 1$ et valeur $\neq 1$ et valeur $\neq \top$) ou ($\mu \neq 1$ et valeur $\neq \mu$) est vérifié.

Ainsi, pour tout palier h de D , $\mu_w(h) \leq \mu_u(h) \wedge \mu_v(h)$ donc $\mu_w(h) \leq \mu_u(h)$ et $\mu_w(h) \leq \mu_v(h)$.

◇

- Vérifions qu w est une hiérarchie cohérente.

L'existence d'un palier h pour la hiérarchie w qui soit nuancé par la valeur,

$$\mu_w \neq 1 \text{ et } \mu_w(h) \neq \nabla_{h' \subset h} (\mu_w(h')) \text{ ou soit par,}$$

$$\mu_w(h) = 1 \text{ et } \nabla_{h' \subset h} (\mu_w(h')) \neq 1 \text{ ou } \neq \top$$

est impossible. En effet, le test ($\mu = 1$ et valeur $\neq 1$ et valeur $\neq \top$) ou ($\mu \neq 1$ et valeur $\neq \mu$) serait vérifié. Dans ce cas la procédure de spécialisation fait appel à la fonction *modifier* qui affecte la valeur \perp au palier h et aux paliers h' qui lui sont inférieurs.

Par conséquent, pour tout palier h de D , on a soit

$$- \mu_w \neq 1 \text{ et } \mu_w(h) = \nabla_{h' \subset h} (\mu_w(h')) \text{ ou soit,}$$

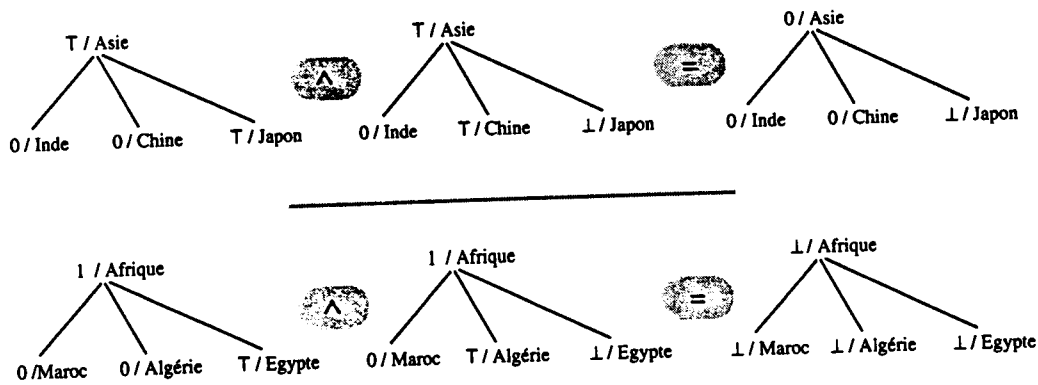


FIG. A.1 - Exemple de spécialisation appliquée aux sous-hiérarchies

- $\mu_w(h) = 1$ et $\nabla_{h' \subset h} (\mu_w(h')) = 1$ ou $= \top$
 c'est à dire que w est une hiérarchie cohérente.

◇

• Vérifions que w est la plus grande des hiérarchies cohérentes inférieures à u et v .

Soit t une hiérarchie cohérente vérifiant $\mu_w \leq \mu_t \leq \mu_u \wedge \mu_v$, nous allons montrer que pour chaque palier h , la procédure de spécialisation donne la borne inférieure restreinte à cette sous-hiérarchie.

Soit h un palier de D , alors nous distinguons les cas suivants :

i) $\mu_u \wedge \mu_v(h) \neq 1$ et $\mu_u \wedge \mu_v(h) = \nabla_{h' \subset h} \mu \wedge \nu(h')$

ii) $\mu_u \wedge \mu_v(h) = 1$ et $\nabla_{h' \subset h} (\mu_u \wedge \mu_v(h')) = 1$ ou $= \top$

iii) $\mu_u \wedge \mu_v(h) \neq 1$ et $\mu_u \wedge \mu_v(h) \neq \nabla_{h' \subset h} \mu \wedge \nu(h')$

iv) $\mu_u \wedge \mu_v(h) = 1$ et $\nabla_{h' \subset h} (\mu_u \wedge \mu_v(h')) \neq 1$ et $\neq \top$

Dans le cas i) et ii), la procédure de spécialisation laisse inchangée la valeur μ initialisée à $x.\mu \wedge y.\mu$, c'est à dire $\mu_w(h) = \mu_u \wedge \mu_v(h)$, par conséquent $\mu_t(h) = \mu_w(h) = \mu_u \wedge \mu_v(h)$. Ces deux cas indiquent que la sous-hiérarchie de racine h est cohérente.

Dans le cas iii), $\nabla_{h' \subset h} \mu \wedge \nu(h') \neq 1$ car le cas contraire aboutirait à la contradiction $\mu_u \wedge \mu_v(h) \neq 1$ et $\mu_u \wedge \mu_v(h) = 1$. Par conséquent, par définition de ∇ on a pour tout $h' \subset h$, $\mu_u \wedge \mu_v(h') \neq 1$.

Dans ce cas si $\mu_u \wedge \mu_v(h) < \nabla_{h' \subset h} \mu \wedge \nu(h')$ alors il existe $h' \subset h$ vérifiant $\mu_u \wedge \mu_v(h) < \mu \wedge \nu(h')$, soit par définition de l'opérateur \wedge , $\mu_u(h) < \mu_u(h')$ et $\mu_v(h) < \mu_v(h')$ ce qui

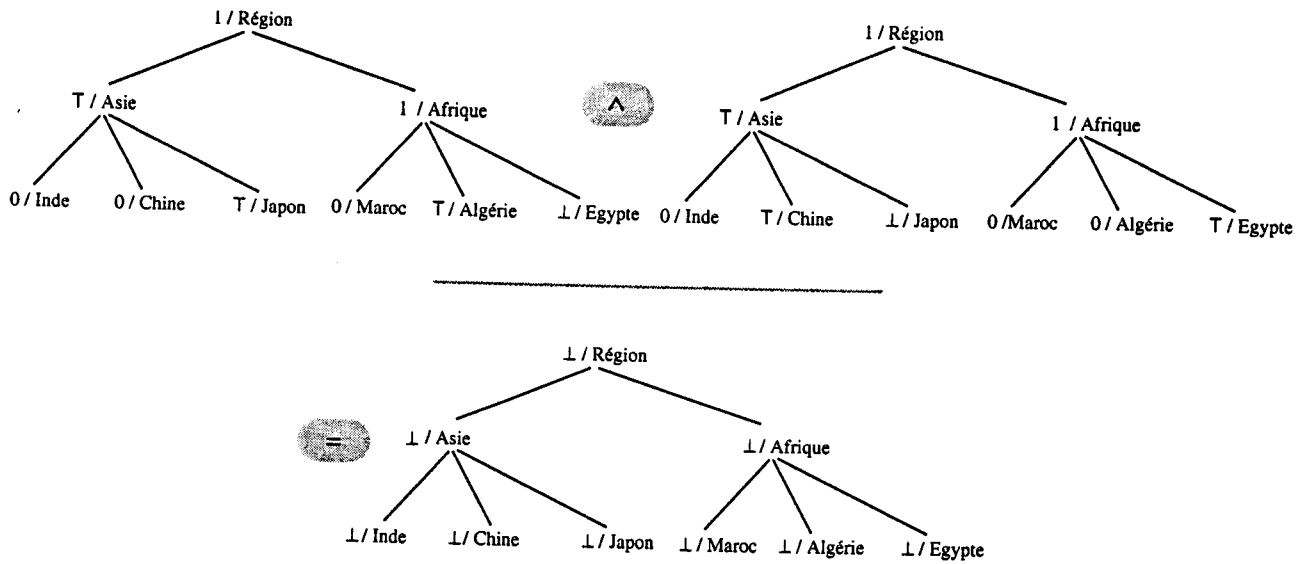


FIG. A.2 - Exemple de spécialisation sur les régions

n'est pas possible pour une hiérarchie cohérente que si $\mu_u(h) = 1$ et $\mu_u(h') = T$ (resp. $\mu_v(h) = 1$ et $\mu_v(h') = T$) d'où la contradiction $\mu_u \wedge \mu_v(h) = 1$.

Ainsi $\mu_u \wedge \mu_v(h) > \bigvee_{h \subset h'} \mu_u \wedge \mu_v(h')$. Comme toutes les valeurs $\mu_u \wedge \mu_v(h')$ sont restreintes aux valeurs de la chaîne $\{\perp, 0, T\}$, \bigvee est compatible avec la relation d'ordre et $\mu_t(h) = \bigvee_{h \subset h'} \mu_u \wedge \mu_v(h') < \bigvee_{h \subset h'} \mu_u \wedge \mu_v(h')$ or la procédure de spécialisation affecte la valeur $\bigvee_{h \subset h'} \mu_u \wedge \mu_v(h')$ à $\mu_w(h)$, d'où $\mu_w(h) = \mu_t(h)$.

Dans le cas *iv*), comme $\bigvee_{h' \subset h} (\mu_u \wedge \mu_v(h')) \neq 1$ et $\neq T$ alors pour tout $h' \subset h$, $\mu_u \wedge \mu_v(h') \neq 1$ et $\neq T$. Or la restriction de \bigvee à $\{\perp, 0, T\}$ est compatible avec la relation d'ordre, d'où $\mu_t(h) = \bigvee_{h' \subset h} (\mu_t(h')) \leq 0$. Or la seule valeur, qui dans ce cas est inférieure à $\mu_u \wedge \mu_v(h) = 1$ et à 0 est la valeur \perp et $\mu_t(h) = \perp$. La procédure de spécialisation dans ce cas, fait appel à la procédure modifier qui affecte la valeur T pour la hiérarchie w à tous les paliers inférieurs à h , ainsi $\mu_w(h) = \mu_t(h)$.

Pour chaque palier et à chaque étape de la spécialisation, $\mu_t(h) = \mu_w(h)$, il en résulte que w est la plus grande hiérarchie cohérente inférieure à u et v .

Bibliographie

- [1] M. Barbut and B. Monjardet. *Ordre et Classification*. Hachette, 1970.
- [2] P. Bertier and J.M. Bourroche. *analyse des données multi-dimensionnelles*. Presse universitaires de France, 1981.
- [3] Garret Birkhoff. Lattice theory. In *Colloquium Publications*, volume 25. American Mathematical Society, 1993.
- [4] G. Bisson. *Induction de Bases de Connaissances en Logique des Prédicats*. PhD thesis, Université de Paris-Sud Centre d'Orsay, 1993.
- [5] B. Bouchon-Meunier. Fuzzy partitions. In M. Singh, editor, *Systems Control Encyclopedia*. Pergamon Press, 1989.
- [6] B. Bouchon-Meunier. Fuzzy logic and knowledge representation using linguistic modifier. In Lofti Zadeh and Janusz Kacprzyk, editors, *Fuzzy Logic for the Management of Uncertainty*. Wiley, 1992.
- [7] B. Bouchon-Meunier. *La Logique floue*. Presse Universitaire de France, Que sais-Je?, 1993.
- [8] G. Celeux. *Analyse discriminante sur variables continues*. Inria, 1990.
- [9] G. Celeux, E. Diday, Y. Govaert, G. Lechevallier, and H. Ralambondrainy. *Classification automatique des données; environnement statistique et informatique*. Dunod, 1989.
- [10] N. Conryut. *Amélioration de la Robustesse des Systèmes d'aide à la Description, à la Classification et à la Détermination des Objets Biologiques*. PhD thesis, Université Paris-IX Dauphine, 1994.

- [11] N. Conryut, D. Grosser, and H. Ralambondrainy. Ikbs: An iterative knowledge base system for improving description, classification and identification of biological objects. In *Indo-French Seminar on Symbolic Data Analysis and its applications*, Paris, 23-25 septembre 1997.
- [12] T.G. Dietterich and R.S. Michalsky. Etude comparative d'un choix de méthodes d'apprentissage à partir d'exemples. In Y. Kodratoff, editor, *Apprentissage symbolique vol. 1*. Toulouse, 1993.
- [13] D. Dubois and H. Prade. *Théorie des Possibilités. Application à la Représentation des Connaissances en Informatique*. Masson 2^e édition, 1988.
- [14] M.L. Ginsberg. Multivalued logics: A uniform approach to inference in artificial intelligence. *Computational Intelligence*, 4:265-316, 1988.
- [15] R. Girard. *Classification Conceptuelle sur des Données Arborescentes et Imprécises*. PhD thesis, Université de la Réunion, 1997.
- [16] J.A. Goguen. L-fuzzy sets. *Journal of Mathematics, Analysis and Applications*, (18):145-174, 1967.
- [17] A. Guénoche. Algorithmes pour la construction de fonctions complètes et consistantes. In *Induction Symbolique et Numérique*. Cepadues-Editions, 1982.
- [18] A. Guénoche. Generalization and conceptual classification: Indices and algorithm. In Edwin Diday, editor, *Nova Science Publisher*, pages 503-510. Data Analysis, Learning Symbolic and Numerical Knowledge, 1989.
- [19] JR.Quinland. Induction of decision trees. *Maching Learning*, 1:81-106, 1991.
- [20] J. Kacprzyk and C. Iwansky. Fuzzy logic with linguistic quantifier in inductive learning. *Fuzzy Logic for the Management of Uncertainty*, 1992.
- [21] A. Kaufmann. Applications à la classification et à la reconnaissance des formes, aux automates et aux systèmes, et aux choix des critères. In *Introduction à la théorie des sous-ensembles flous vol. 3*. Masson, 1975.
- [22] A. Kaufmann. *Introduction à la théorie des sous-ensembles flous - Eléments théoriques de base*. Masson, 1977.

- [23] G.J. Klir and T.A. Folger. *Fuzzy sets, uncertainty, and information*. Prentice-Hall International Editions, 1992.
- [24] Y. Kodratoff and E. Diday. *Induction symbolique et numérique à partir de données*. Cépaduès-Éditions, Toulouse, 1991.
- [25] IC. Lerman and J. Nicolas. Une classification conceptuelle : une approche centrée sur la similarité. In Y. Kodratoff and E. Diday, editors, *Induction symbolique et numérique à partir de données*, pages 153–177. Cépaduès-Éditions, Toulouse, 1991.
- [26] D. Loutchmia and H. Ralambondrainy. Apprentissage inductif d'un concept flou. *Rencontre francophone sur la logique floue et ses applications / LFA '95*, pages 148–154, 1995.
- [27] D. Loutchmia and H. Ralambondrainy. Inductive learning using similarity measures on lattice-fuzzy set. *Proceedings of 6th IEEE International conference on Fuzzy Systems*, pages 1307–1313, 1997.
- [28] D. Loutchmia and H. Ralambondrainy. Mesure de similitude sur les objets structurés à valeurs imprécises. *Rencontre francophone sur la logique floue et ses applications / LFA '97*, pages 277–284, 1997.
- [29] D. Loutchmia and H. Ralambondrainy. Similarity measure on structured objects with imprecise values. *International Symposium on Engineering of Intelligent Systems/ EIS'98*, 1998.
- [30] R.S. Michalsky. A theory and methodology of inductive learning. In R.S. Michalsky, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach, vol.1*. Morgan Kaufman, Los Altos, 1983.
- [31] T. Mitchell. Version spaces: A candidate elimination approach to rule learning. *I.J.C.A.I.*, pages 305–310, 1977.
- [32] H. Prade. A two-layer fuzzy pattern matching procedure for the evaluation of conditions involving vague quantifiers. *Journal of Intelligent and Robotic System*, 3:93–101, 1990.

- [33] H. Ralambondrainy. Apprentissage dans le contexte d'un schéma de base de données. In Y. Kodratoff and Edwin DiDay, editors, *Induction Symbolique et Numérique*. Cépaduès-Édition, 1991.
- [34] H. Ralambondrainy. A conceptual version of the k-means algorithm. *Pattern Recognition Letters*, pages 1147–1157, 1995.
- [35] J. Rives. Fid3: Fuzzy induction tree. In *Proceedings of IEEE*, pages 457–462, 1990.
- [36] R. Tomassone, E. Lesquoy, and C. Millier. *La régression : nouveaux regards sur une ancienne méthode statistique*. Masson, 1983.
- [37] J. Sallantin, J. Quinqueton, C. Barboux, and J.P. Aubert. Théorie semi-empiriques : éléments de formalisation. *Revue d'intelligence artificielle*, 5:69–92, 1 1991.
- [38] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.
- [39] LÉA SOMBÉ. *Raisonnements sur des informations incomplètes en Intelligence Artificielle*. Teknea, 1989.
- [40] R.E. Stepp and R.S. Michalsky. Classification conceptuelle : Apprentissage à partir d'observations et classification d'objets structurés avec objectif. In Y. Kodratoff, editor, *Apprentissage symbolique vol. 1*. Toulouse, 1993.
- [41] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [42] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1:193–218, 1978.

RÉSUMÉ

Les travaux de cette thèse se situent dans le cadre de la discrimination conceptuelle. Notre objectif est de caractériser des classes d'objets imprécis et structurés, à partir de concepts imprécis. Le modèle de représentation des données utilise un formalisme objet pour décrire les relations de dépendance entre les attributs et modélise l'imprécision à l'aide de termes linguistiques organisés en structure de treillis.

Dans un premier temps, nous avons défini des fonctions de reconnaissance floues permettant de mesurer l'appartenance graduelle d'une observation à un concept. Ces fonctions sont construites à partir de métriques qui prennent en compte les différents aspects du modèle.

Ensuite, à partir des notions de complétude et de consistance, nous avons élaboré des critères évaluant la qualité de discrimination d'un concept au contour imprécis. L'utilisation de quantificateurs linguistiques permet de préciser cette qualité de discrimination en tenant compte des erreurs de classement.

Enfin, nous avons développé des méthodes d'acquisition de concepts imprécis à partir d'exemples et de contre-exemples. Ces méthodes déterminent un recouvrement des exemples à partir de descriptions reconnaissant au plus un contre-exemple. Nos méthodes de discrimination sappuient, pour déterminer efficacement des concepts, sur les opérateurs de généralisation et sur la structure de treillis des descriptions discriminantes. Afin d'améliorer la qualité et la fiabilité des résultats, une procédure de sélection des attributs les plus discriminants a été proposée. Implantées dans un système d'apprentissage, ces différentes méthodes sont appliquées à plusieurs expérimentations.