Mesures de discrimination et leurs applications en apprentissage inductif

Thanh Ha Dang

UNIVERSITE PIERRE & MARIE CURIE
LA SCIENCE A PARIS

Mesures de discrimination et leurs applications en apprentissage inductif

Thèse de doctorat de l'Université de Paris 6

présentée pour obtenir le grade de

Docteur de l'Université Paris 6
(spécialité informatique)

par

Thanh Ha Dang

soutenue le 10 juillet 2007

devant le jury composé de

M. Alain Boucher (Professeur, IFI)	Co-encadrant de thèse
Mme Bernadette Bouchon-Meunier (Directeur de recherche, CNRS)	Directrice de thèse
M. Christophe Marsala (Maître de conférences, Université Paris 6)	Co-encadrant de thèse
M. Patrice Perny (Professeur, Université Paris 6)	Examinateur
M. Mohammed Ramdani (Professeur, FST Mohammedia)	Rapporteur
M. Djamel A. Zighed (Professeur, Université Lyon 2)	Rapporteur

Remerciements

En premier lieu, je tiens à exprimer ici ma profonde gratitude envers Bernadette Bouchon-Meunier, qui m'a accueilli dans son équipe de recherche depuis mon stage de fin d'études de Master. Sous sa direction j'ai eu l'honneur et la chance d'effectuer ma thèse dont le présent mémoire est l'aboutissement. Pendant ces années, elle m'a fait bénéficier sans relâche de son savoir, de ses compétences scientifiques et de sa grande disponibilité. J'ai également pu profiter de ses conseils, de son soutien, et de ses encouragements tout au long de ces années. Tout cela est particulièrement précieux, notamment pour moi, étudiant qui vient d'un pays lointain.

J'adresse mes plus chaleureux remerciements à Christophe Marsala pour m'avoir encadré tout au long de ma thèse. Les séances de travail que j'ai eues avec lui ont été toujours intéressantes grâce à ses compétences, sa disponibilité et sa pédagogie. Elles m'ont permis de faire progresser mes recherches. Je lui dois également beaucoup pour ses encouragements, ses conseils, sa gentillesse et son soutien ainsi que des discussions sur de nombreux sujets.

Je tiens à remercier vivement Alain Boucher pour son co-encadrement lors de mes séjours à l'IFI. Il m'a toujours donné des conditions de travail favorables. C'est lui qui m'a motivé à travailler sur l'évaluation des classifieurs. Il m'a également beaucoup apporté par son soutien enthousiaste et ses conseils précieux.

Je remercie vivement Mohammed Ramdani de s'être intéressé à mon travail et d'avoir accepté d'en être rapporteur. Il est pour moi un juge incontournable pour mon travail. Les discussions que nous avons eues me permettent d'élargir les perspectives de mes travaux, et notamment de me donner une vision plus appliquée de ce que j'ai fait.

Je remercie Djamel A. Zighed pour avoir accepté de juger ma thèse. Je le remercie pour ses commentaires et ses suggestions qui ont permis d'améliorer ce travail.

Je suis honoré de la présence de Patrice Perny dans mon jury de thèse. Je le remercie d'avoir accepté d'y participer et de le présider. Ses remarques et ses suggestions ouvrent de nombreuses perspectives à mon travail.

Je voudrais remercier les personnes avec lesquelles j'ai eu occasion de collaborer pendant ma thèse : Thomas Delavallade avec qui j'ai développé la méthode de traitement de données manquantes, Marc Damez avec qui j'ai travaillé sur le système de classification de traces, Lê Thị Lan avec qui j'ai étudié la caractérisation des bases d'images. Ces travaux en collaboration constituent une partie importante de ma thèse.

Je tiens également à remercier les ingénieurs et le personnel administratif du

LIP6 et de l'IFI qui ont contribué à l'aboutissement de ma thèse par les facilités qu'ils m'ont apportées et leur gentillesse.

Je suis sensible à la chance que j'ai eue d'être intégré dans une équipe de recherche formidable : l'équipe LOFTI. Je tiens à remercier tous les membres et les anciens membres de l'équipe : Herman, Maria, Liva, Delphine, Jason, Thomas, Nicolas, Adrien, Marcin et tous les autres, qui ont rendu ce travail possible par leur aide et leur contribution. J'ai particulièrement apprécié l'ambiance dans l'équipe. Les déjeuners, les pauses café et les pauses en fin après-midi sont des moments intéressants où on a partagé plein de choses, où ils m'ont fait découvrir différentes cultures.

Je remercie particulièrement Marie-Jeanne Lesot pour la relecture du manuscrit de ma thèse, le soutien et les encouragements qu'elle m'apportés pendant les années de thèse. Malgré son emploi du temps chargé, j'ai toujours obtenu rapidement ses remarques très pertinentes. J'ai pu bénéficier de sa grande disponibilité et de sa sincérité pour avoir des conseils scientifiques et pratiques, qui sont d'ailleurs toujours très précieux pour moi. Elle est aussi l'origine de nombreuses idées d'activités hors de la recherche qui m'ont permis de découvrir la culture française.

Cette thèse a été financée par une bourse de formation à la recherche de l'Agence Universitaire de la Francophonie (AUF) et une bourse d'études du gouvernement français. Je les en remercie.

Sur un plan plus personnel, je ne saurais oublier le soutien de mes parents, ma petite sœur Thanh Hai, mon petit frère Ngọc Lân. Même si nous sommes géographiquement éloignés les uns des autres, nous sommes toujours restés très proches. Ils ont été à mes côtés au long de ces années et ils ont toujours une forte confiance dans ma réussite.

Je voudrais remercier également toutes les personnes qui m'ont entouré pendant ces années de thèse. Leurs amitiés, leur soutien et leurs encouragements ont rendu ma vie plus heureuse et plus facile.

J'exprime finalement toute ma gratitude à Hiêú pour son amour, son courage et sa générosité. Elle a constitué, pour moi, un des principaux moteurs pour achever mon travail de thèse.

Résumé

De nos jours, les données disponibles deviennent de plus en plus volumineuses et elles peuvent être de nature très diverse : vagues, manquantes, numériques, symboliques par exemple. Or ce qui importe à l'utilisateur, ce ne sont pas les données elles-mêmes, mais les connaissances qu'on peut en extraire. Face à la quantité de données disponibles, le traitement efficace de données est problématique. Dans cette thèse, nous adoptons une approche d'extraction de connaissances à partir de données basée sur *l'apprentissage inductif*, plus précisément, par arbres de décision.

De façon générale, un système construit par apprentissage inductif a pour but de discriminer les individus de différentes classes. Sa qualité dépend de la *capacité de discrimination* qu'il acquiert au cours de l'apprentissage au travers des données. En particulier, un algorithme de construction d'arbre de décision procède par évaluation successive de la capacité de discrimination des attributs pour construire l'arbre de décision.

Nos travaux concernent l'étude des mesures de discrimination tant classiques que floues, et leurs applications en apprentissage inductif.

D'une part, nous nous intéressons aux mesures de discrimination dans la construction des arbres de décision. Dans un premier temps, ces mesures font l'objet d'une étude selon une approche axiomatique. Nous développons un nouveau modèle pour caractériser les mesures de discriminations floues. Dans un deuxième temps, nous proposons d'utiliser ces mesures dans les différentes étapes de la construction des arbres de décision flous.

D'autre part, nous étudions l'utilisation de ces mesures de discrimination pour d'autres aspects de l'apprentissage. Nous examinons tout d'abord le problème de l'évaluation des classifieurs et proposons une méthode basée sur l'utilisation de la notion de capacité de discrimination. Enfin, nous considérons le problème du traitement des données manquantes et proposons une technique de substitution des valeurs manquantes, qui restitue la capacité de discrimination des attributs.

Ces travaux sont validés sur des données conventionnelles et appliqués à des données réelles dans le cadre de deux applications qui concernent la classification de courriers électroniques et la classification de traces d'interactions homme-machine.

Mots-clés : apprentissage inductif, arbre de décision, mesure de discrimination, entropie, traitement de données manquantes, évaluation de classifieurs

Abstract

Nowadays, the available data become more and more voluminous and diverse by nature : vague data, missing data, numerical or symbolic data can be encountered. However, users are more interested in the knowledge which can be extracted from the data, than by the data themselves. Vis-à-vis the great quantity of available data, the effective processing of data is very cumbersome. In this thesis we adopt an approach of knowledge extraction from data based on *inductive learning*, more precisely by using the decision tree technique.

In general, the purpose of a system constructed by inductive learning is to discriminate the individuals belonging to different classes. Its quality depends on its *discrimination power* which is acquired during the learning phase through the data. In particular, an algorithm of construction of a decision tree works by successively evaluating the discrimination power of the attributes.

In this thesis, we investigate the measures of discrimination, both classical and fuzzy, and their applications in inductive learning.

On the one hand, we consider discrimination measures for the construction of decision trees. We begin by studying these measures following an axiomatic approach and develop a new model which permits to characterize fuzzy measures of discrimination. Then, we propose to use these measures during the various stages of construction of fuzzy decision trees.

On the other hand, we study the use of these measures of discrimination during other steps of the learning process. Firstly, we examine the classifier evaluation process and propose an evaluation criteria based on the concept of discrimination power. Next, we consider the missing data problem and propose a new technique of imputation by restoring the discrimination power of attributes.

This work is validated on conventional data and is applied to some real problems such as email classification and human-computer interaction traces classification.

Keywords : inductive learning, decision tree, discrimination measure, entropy, missing data handling, classifier evaluation

Table des matières

Table des figures

Notations

ξ : ensemble d'exemples de la base

N : nombre d'exemples dans la base

e, e_i : exemple ($1 \leq i \leq N$)

$\mathcal{C}$: ensemble de classes

n : nombre de classes

C, C_i : classe ($1 \leq i \leq n$)

$e(C)$: la classe de l'exemple e

$\mathcal{A}$: ensemble d'attributs

K : nombre d'attributs (hors attribut de classe)

A, A_k : attribut A (dans le cas général), attribut A_k ($1 \leq k \leq K$)

$e(A)$: valeur de l'attribut A pour l'exemple e

m, m_k : nombre de valeurs d'un attribut A (dans le cas général), de l'attribut A_k

v_j, v_{k_j} : valeur de l'attribut considéré (par défaut attribut A), valeur de l'attribut A_k ($1 \leq j \leq m$, $1 \leq k_j \leq m_k$)

c, c_i : coupure

δ : étalement

D, D_k : domaine d'attribut A, A_k

$P_i, P(C_i), P_i^*, P^*(C_i)$: probabilité et probabilité floue de la classe C_i dans un ensemble d'exemples

$P(v_j), P^*(v_j)$: probabilité et probabilité floue que l'attribut en question prenne v_j comme valeur

$P(C_i|v_j)$: probabilité de la classe C_i conditionnée par la valeur v_j de l'attribut considéré

$I, I_S, I_D^\beta, I_R^\beta, I_{R-norme}^\beta$: entropie, entropie de Shannon, entropie d'ordre β de Daróczy, de Rényi et la R-norme entropie (d'un événement, d'un ensemble ou d'une distribution de probabilité)

β : coefficient de l'entropie, $\beta > 0$ et $\beta \neq 1$

F, G, H : les fonctions au niveau $\mathcal{F}, \mathcal{G}, \mathcal{H}$ du modèle $\mathcal{FHG}$

F^*, G^*, H^* : les fonctions au niveau $\mathcal{F}^*, \mathcal{G}^*, \mathcal{H}^*$ du modèle $\mathcal{FHG}^*$, une extension vers la théorie des sous-ensembles flous du modèle $\mathcal{FGH}$

μ : fonction d'appartenance

$\triangle I$: gain d'information

τ : taux de gain d'information

log : sans précision, se comprend « logarithme en base 2 » ($\log_2$).

INTRODUCTION

Introduction

Grâce aux progrès scientifiques, en particulier sur les techniques d'acquisition de données et les matériels de stockage, les données disponibles deviennent de plus en plus volumineuses. On peut citer comme exemples de données gigantesques : les données disponibles sur l'Internet (texte, image, vidéo, son) ou des données provenant d'applications spécifiques, comme, par exemple, des données biologiques ou des données issues de traces d'interactions homme-machine. Ces données peuvent être de nature très diverse : elles peuvent être de différents types ou avoir des caractéristiques variées. Plus précisément, on peut distinguer les données numériques, symboliques ou linguistiques, etc. Concernant les caractéristiques, elles peuvent être considérées comme étant imprécises et/ou incertaines, complètes ou incomplètes, structurées ou non structurées...

Or ce qui importe à l'utilisateur, ce ne sont pas les données même, mais les connaissances qu'on peut en extraire - celles-ci sont évidemment plus significatives et plus intéressantes pour les être humains. Le traitement de données fait partie des activités habituelles chez les êtres humains. Pourtant, face à la quantité de données disponibles, le traitement efficace de données est problématique et dépasse la capacité du traitement manuel. Autour des volumes de données, plusieurs questions se posent donc : comment exploiter automatiquement une telle quantité de données de manière efficace pour en déduire des connaissances utiles dans un but concret ? Comment extraire des connaissances utiles qui aient un sens pour les êtres humains ? Comment caractériser ces données ou caractériser les phénomènes représentés par ces données ?

Apprentissage inductif par arbres de décision

Nous nous plaçons dans le domaine de l'apprentissage automatique. Plusieurs techniques automatiques ont été développées pour répondre à ces problèmes. Il s'agit d'un ensemble de techniques qui s'inspirent de processus d'apprentissage mis en œuvre par un être humain et visent à doter l'ordinateur de cette capacité et le rendre *intelligent*. On distingue deux approches : un processus d'apprentissage peut être réalisé de manière déductive ou de manière inductive. La différence essentielle entre l'apprentissage déductif et l'apprentissage inductif est la façon de construire des règles. En apprentissage déductif, des nouvelles règles sont déduites à partir des anciennes. En apprentissage inductif (apprentissage par l'exemple), on essaie de

trouver des règles à partir d'un ensemble de cas connus, structurés dans une base d'apprentissage, selon une méthode de généralisation, c'est-à-dire du particulier au général. Chaque cas, appelé « exemple » ou « individu » est décrit par des attributs et les valeurs qui leur sont associées ainsi qu'une classe. Le but ici est d'apprendre l'association entre les attributs et les classes. Après avoir obtenu des règles qui résument ces exemples, le système peut traiter de nouveaux cas et déterminer leur classe. Les règles doivent donc être capables de discriminer les exemples des différentes classes.

L'apprentissage inductif est l'une des approches pour résoudre des problèmes d'extraction de connaissances à partir de données. Il apparaît dans plusieurs types d'applications, par exemple, en classification, en prédiction, en extraction de règles à partir de données ou en résumé de données, ou encore en généralisation... Plusieurs techniques ont été proposées et utilisées, on peut citer comme exemples : réseaux neuronaux, arbres de décision, réseaux bayésiens, etc. Dans notre thèse, parmi ces techniques, nous nous intéressons aux arbres de décision. Il s'agit d'une technique d'extraction de connaissances à partir de données performante et très populaire. Dans [40], les auteurs rapportent le résultat d'enquêtes menées en août 2001 par Piatetsky-Shapiro[1] sur son site dédié au marché industriel de l'extraction de connaissances à partir de données. D'après cette étude les arbres de décision sont l'outil le plus utilisé. Cette technique est utilisée par 19% des personnes interrogées. Il faut noter que selon la même enquête, 35% des personnes interrogées utilisent souvent des outils plutôt symboliques (arbres de décision, règles d'association, etc) et non des outils plutôt statistiques (réseaux neuronaux, régression logistique, etc) ni d'autres outils utilisés dans la fouille de textes, fouille de la toile, etc. Ceci signifie que plus de la moitié des utilisateurs d'outils symboliques utilisent les arbres de décision. Très récemment, à la question « Quelles sont les techniques de fouille de données que vous utilisez souvent dans les derniers 12 mois ? » posée sur le site de KDnuggets[2], 62.6% des personnes citent les arbres de décision. Ceci confirme encore que les arbres de décision sont très utilisés ces dernières années.

Il existe plusieurs algorithmes efficaces de construction d'arbres de décision. Les méthodes existantes permettent un traitement homogène de presque tous les types d'attributs : numérique, symbolique, flou ou probabiliste. Ces algorithmes sont également adaptés à différentes problématiques : la présence de données manquantes, erronées, imprécises, incertaines, etc et à différentes situations : construction incrémentale, construction en parallèle, etc. Une fois l'arbre construit, son utilisation est très simple et peu coûteuse. Enfin, un avantage très attractif de ces techniques d'arbres de décision réside dans l'interprétabilité des résultats qu'ils fournissent. Cela permet de savoir pourquoi un exemple est ainsi traité. En particulier, lorsqu'une valeur d'un attribut est exigée, sa contribution dans l'identification de classe peut être justifiée. L'interprétabilité est très appréciée car, d'une part elle est nécessaire dans certaines applications réelles et d'autre part, elle fait partie des comportements qualifiés intelligents, similaires à ceux d'un être humain.

[1] http ://www.kdnuggets.com/gps.html

[2] http ://www.kdnuggets.com/polls/2007/data_mining_methods.htm, le 7 mai 2007

Prise en compte de l'imperfection dans les données

Dans notre travail, nous nous sommes intéressés aux arbres de décision dans un contexte particulier : le cas où les données sont imprécises, incertaines et/ou manquantes. L'existence des imperfections dans une base de données est très fréquente. Celles-ci proviennent d'une part de la phase d'obtention des données à partir du réel. En particulier, cela résulte des limites des instruments d'observation, de la capacité humaine, de la représentation des données, des difficultés (voire de l'impossibilité) à obtenir des données, ou de l'indisponibilité des données. D'autre part, elle peut provenir de la nature même des données, qui peuvent être intrinsèquement imprécises. Par exemple, on ne peut pas spécifier exactement le moment du passage du jour à la nuit [20]. Pour effectuer des analyses adéquates, il est donc important de tenir compte de ces problèmes.

La logique floue est un concept introduit par Zadeh en 1965 qui fournit un formalisme pour modéliser l'imprécision et l'incertitude [163]. En particulier, les sous-ensembles flous ont été introduits pour généraliser la notion d'appartenance et pour éviter les passages brusques d'une classe à une autre. Pour cela, un élément peut appartenir partiellement à un sous-ensemble et il peut simultanément appartenir à plusieurs sous-ensembles flous. Tandis que l'appartenance d'un individu à un sous-ensemble classique est un concept binaire, l'appartenance floue est continue et caractérisée par une valeur de l'intervalle [0,1]. Cette théorie trouve de multiples applications dans des domaines extrêmement variés [21].

Depuis plus d'une quinzaine d'années, les techniques issues de la théorie des sous-ensembles flous ont été incorporées dans les méthodes de construction d'arbres de décision (un exemple d'arbre de décision flou est présenté dans la figure 1). D'abord, elle fournit une méthodologie pour représenter des incertitudes et imprécisions des données. En particulier, lorsque la valeur d'un attribut est vague, elle peut être décrite par un sous-ensemble flou défini sur le domaine de l'attribut. Par exemple, le concept de grande vitesse pour un train est vague et peut être décrit par un sous-ensemble flou. Quand les classes ne sont pas très bien séparées, une classe peut être caractérisée par un sous-ensemble flou des exemples. Ainsi, on peut déduire des connaissances graduelles à partir de ces données. L'introduction de ces techniques permet d'enrichir les capacités des arbres de décision en prenant en compte des incertitudes et imprécisions des données. En particulier elle offre un traitement des données plus souple, plus robuste et plus précis.

Nous considérons le problème de l'extraction automatique de connaissances en considérant les deux types d'outils présentés ci-dessus : les arbres de décision et la logique floue.

Mesure de discrimination en apprentissage inductif

Comme nous l'avons mentionné, un système construit par l'apprentissage inductif a pour but de discriminer les exemples des différentes classes. Un tel système est appelé « classifieur » ou « modèle de classification » dans cette thèse. Sa qualité

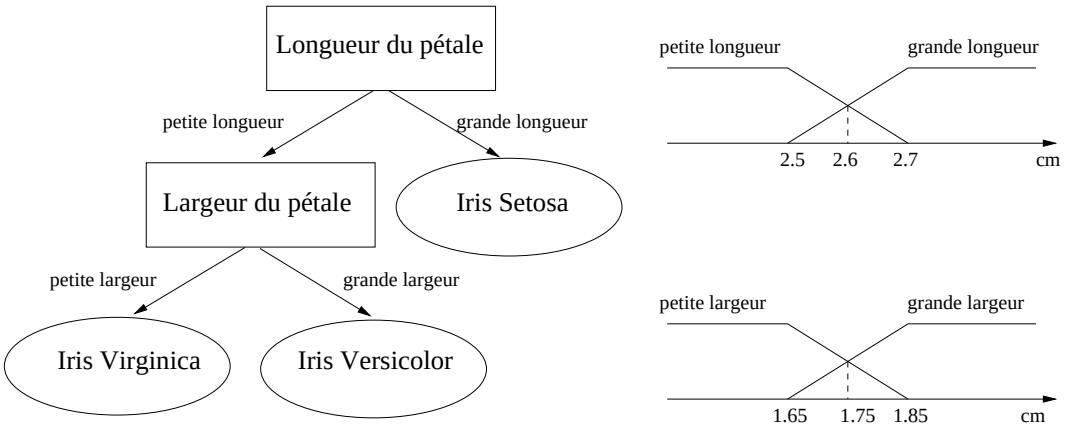

FIG. 1 – Exemple d'arbre de décision flou pour les données Iris

dépend de la *capacité de discrimination* qu'il a acquise au cours de l'apprentissage. L'hypothèse que l'on pose ici est que les exemples des classes différentes ont une ou plusieurs caractéristiques différentes. Un attribut décrit une caractéristique des exemples. L'hypothèse suppose donc que les attributs possèdent une certaine capacité à discriminer les classes. Dans la phase de construction du modèle de classification, une fouille des données est effectuée afin de trouver des caractéristiques intéressantes. Ce sont elles qui serviront ensuite à discriminer des exemples.

En particulier, dans la construction d'un arbre de décision, la capacité de discrimination d'un attribut est considérée à chaque étape. Elle est généralement évaluée par une mesure issue de la théorie de l'information, telle que l'entropie de Shannon. Un algorithme de construction d'arbre de décision procède par l'accumulation processive de la capacité de discrimination des attributs dans l'arbre de décision. Or, l'entropie de Shannon n'est pas la seule mesure habilitée à être utilisée dans un tel processus. Plusieurs mesures, en particulier celles issues de la théorie de l'information, sont utilisables, sous l'une ou l'autre forme, en apprentissage inductif. Cependant, il manque une étude approfondie qui, d'une part, justifie l'utilisation de ces mesures dans un tel processus, notamment en présence des imprécisions et des incertitudes dans les données, et d'autre part généralise l'utilisation de ces mesures aux autres étapes de l'apprentissage inductif, telle que le traitement de données manquantes, l'évaluation de classifieurs. Notre thèse se propose de réaliser une telle étude.

Description de nos travaux

Nous proposons d'étudier les mesures de discrimination dans le contexte de l'apprentissage inductif. Dans un premier temps, ce type de mesures fait l'objet d'une étude selon une approche axiomatique. Dans cette étude, les mesures de discrimination classiques sont caractérisées dans le cadre du modèle hiérarchique proposé par Marsala [102]. Pour les mesures de discrimination floues, nous développons un nou-

veau modèle, plus général, afin de les caractériser. Dans un deuxième temps, nous proposons d'utiliser ces mesures dans la construction des arbres de décision flous. Un système d'induction automatique d'arbres de décision est développé et utilisé dans le cadre de nos études. Les mesures sont caractérisées à chaque étape de ce processus, en particulier, pour la sélection d'attributs et pour la discrétisation des attributs numériques.

Nous généralisons ensuite l'utilisation des mesures de discrimination à d'autres aspects de l'apprentissage. Nous examinons tout d'abord le problème de l'évaluation des classifieurs et proposons une méthode basée sur l'utilisation de la notion de capacité de discrimination. Enfin, nous considérons le problème du traitement des données manquantes et proposons une technique de substitution des valeurs manquantes d'un attribut restituant la capacité de discrimination de ce dernier.

Structure de la thèse

Cette thèse est structurée en trois parties. Elles sont organisées de l'aspect théorique à l'aspect applicatif.

La première partie est consacrée à l'étude des mesures de discrimination. Elle se compose du chapitre 1. Ces mesures sont étudiées à l'aide d'un modèle hiérarchique qui définit un ensemble de propriétés des mesures de discrimination. Nous montrons que plusieurs mesures existantes dans l'état de l'art satisfont ce modèle. Nous proposons ensuite une généralisation de ce modèle pour prendre en compte les mesures de discrimination floues. Le modèle général obtenu sert à caractériser les mesures utilisées dans la construction d'arbres de décision flous.

La deuxième partie présente les applications des mesures de discrimination en apprentissage inductif. Elle s'articule en 3 chapitres présentant différents aspects de nos travaux.

Le chapitre 2 aborde la construction des arbres de décision, en particulier des arbres de décision flous. Nous recensons les travaux existants et proposons une taxonomie des méthodes de construction d'arbres de décision flous. Nous examinons entre autres la sélection d'attributs et la discrétisation des attributs numériques. Nous généralisons ensuite l'utilisation des mesures de discrimination classique et floue. Un système de construction d'arbres de décision avec de nouvelles mesures est décrit.

Le chapitre 3 présente une étude sur la capacité de discrimination d'un modèle de classification. Nous proposons d'évaluer des classifieurs à l'aide d'une mesure de discrimination.

Dans le chapitre 4, une nouvelle technique de traitement des données manquantes basée sur les mesures de discrimination est présentée. Elle est validée par une série d'expérimentations.

La dernière partie de la thèse est consacrée au développement d'applications réelles. Dans le chapitre 5, l'application des arbres de décision à deux problèmes réels est présentée. Le premier problème consiste à caractériser des utilisateurs à partir de traces d'interactions homme-machine. Le deuxième concerne la classification de courriels.

Enfin, dans le dernier chapitre, nous présentons les conclusions que nous pouvons retirer de notre étude. Un certain nombre de propositions d'approfondissement de nos travaux ainsi que les perspectives qu'ils ouvrent sont également présentées.

Première partie

MESURES DE DISCRIMINATION

Chapitre 1

Modèle hiérarchique pour les mesures de discrimination

1.1 Introduction

D'après le dictionnaire « Trésor de la Langue Française Informatisé » [1] la *discrimination* est définie comme :

Discrimination : *[Sans idée de traitement inégal] Action, fait de différencier en vue d'un traitement séparé (des éléments) les uns des autres en (les) identifiant comme distincts.*

D'après le dictionnaire Petit Larousse (1997), la *discrimination* est définie comme :

Discrimination : *1. Action d'isoler, et de traiter différemment certains individus, un groupe entier par rapport aux autres, 2. Distinction.*

Une mesure de discrimination est un indice qui quantifie les différences entre des individus ou entre des classes d'individus. Elle s'appuie donc généralement sur les caractéristiques que possède un individu mais pas les autres. Cela permet également d'identifier, de distinguer, d'isoler, de reconnaître ou de classer des individus.

Une telle mesure peut s'appliquer, soit à des couples d'objets (mesure de différence, mesure de séparation entre deux individus), soit à un discriminateur qui sépare des objets de différentes catégories. Nous donnons ci-dessous quelques exemples pour différents types d'objets et différents types de discriminateurs.

Mesure de discrimination pour un couple d'objets

Une mesure de discrimination se comprend comme la quantification de la différence entre deux individus. Elle mesure à quel point deux individus sont différents et à quel point un individu est discriminé vis-à-vis d'un autre. Dans un espace avec une mesure de distance, plus deux individus sont éloignés, plus ces deux individus sont discriminés. Par exemple, la distance est une mesure de discrimination entre deux villes.

[1] http ://atilf.atilf.fr/tlf.htm

Dans [39] la mesure de discrimination entre deux sous-ensembles flous A et B définis dans un univers $\mathcal{X}$ quelconque, est définie comme une mesure de séparation :

$$s(\mu_A, \mu_B) = 1 - \sup_{x \in \mathcal{X}} |\min(\mu_A(x), \mu_B(x))|$$

où μ_A (respectivement μ_B) est la fonction d'appartenance de A (respectivement B). Deux sous-ensembles flous sont ainsi discriminés au maximum s'ils n'ont aucun point de chevauchement.

En statistique, une mesure de discrimination entre deux distributions de probabilité évalue leur incohérence. Notant $\mathcal{P}$ et $\mathcal{Q}$ deux distributions de probabilité sur un ensemble fini $\mathcal{X}$. L'information discriminante entre ces deux distributions est définie par :

$$D_{KL}(\mathcal{P}, \mathcal{Q}) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

C'est la distance de Kullback-Leibler entre les deux distributions. Elle évalue également la quantité d'information nécessaire pour distinguer $\mathcal{P}$ de $\mathcal{Q}$. Dans la transmission des signaux, $D_{KL}(\mathcal{P}, \mathcal{Q})$ mesure par exemple la difficulté de distinguer les signaux de la source $\mathcal{P}$ de ceux de la source $\mathcal{Q}$ [68]. Un ensemble de propriétés de la distance de Kullback-Leibler a été démontré dans le même article.

Dans [146], une autre mesure de discrimination a été introduite pour deux distributions de probabilité $\mathcal{P}$ et $\mathcal{Q}$:

$$L_\beta(\mathcal{P}, \mathcal{Q}) = \frac{\beta}{1-\beta} \log \left(\sum_{x \in \mathcal{X}} P(x) \left(\sum_{y \in \mathcal{X}} \left(\frac{Q(y)}{Q(x)} \right)^\beta \right)^{\frac{\beta}{1-\beta}} \right) - I_R^\beta(\mathcal{P})$$

où $\beta > 0$ et $I_R^\beta(\mathcal{P})$ est l'entropie d'ordre β de Rényi (cf. page 26).

Mesure de discrimination pour un discriminateur

Une mesure de discrimination quantifie la capacité de discrimination d'un discriminateur vis-à-vis des individus. Elle doit ainsi mesurer la manière avec laquelle le discriminateur sépare l'ensemble des individus. Plus des individus différents sont discriminés par le discriminateur, plus sa capacité de discrimination est élevée.

Par exemple, dans le cadre d'examens scolaires, lorsqu'elle est appliquée à une question d'une interrogation, une mesure de discrimination évalue à quel point la question distingue les bons des mauvais élèves. Si tous les élèves peuvent répondre à la question posée, la question n'est pas discriminante. Il en est de même si tous les élèves donnent une mauvaise réponse. La mesure de discrimination peut être définie comme la différence entre la proportion des bons élèves qui répondent correctement à la question et la proportion des mauvais élèves qui y répondent correctement. Plus sa valeur absolue est grande, plus la question est discriminante.

On peut également citer comme discriminateurs un modèle de classification, un attribut, une coupure ou une surface, etc.

Notion de discrimination en apprentissage inductif

Nous nous intéressons plutôt à la seconde catégorie des mesures de discrimination. À notre connaissance, il n'existe pas de définition formelle et universelle de cette catégorie de mesures. La définition et l'utilisation d'une mesure de discrimination sont souvent limitées dans un contexte concret, tel que l'apprentissage inductif ou la classification. Dans notre contexte de l'apprentissage inductif par arbres de décision, on rencontre souvent des problèmes liés à la discrimination. Nous les mentionnerons brièvement ici. Une description plus détaillée des arbres de décision et des différentes étapes de leur construction et utilisation est donnée dans le chapitre 2.

L'apprentissage inductif pour un problème de classification a pour objectif de construire un modèle qui a le meilleur pouvoir de discrimination vis-à-vis de la classe pour un ensemble d'exemples. En ce sens, le modèle cherche à capturer des caractéristiques possédées par une classe mais pas par d'autres pour discriminer les classes entre elles. Naturellement, la capacité de discrimination d'un modèle est un critère apprécié dans l'évaluation et la comparaison des arbres. Nous étudierons cet aspect dans le chapitre 3 (page 91).

Les mesures de discrimination servent également dans l'étape de discrétisation des attributs numériques (cf. section 2.4, page 67) : dans certains cas, il est intéressant de traiter les valeurs numériques comme des valeurs symboliques, entre autres, pour réduire le nombre de valeurs à traiter, renforcer la tolérance au bruit qui différencie les mêmes valeurs originales, ou mieux exploiter la proximité entre valeurs. Il faut donc regrouper les valeurs numériques qui sont proches, les délimiter par des coupures et représenter toutes les valeurs d'un intervalle par une seule valeur symbolique. L'idée générale est d'essayer de discriminer le plus possible les différentes classes d'exemples par des coupures. Idéalement, on souhaite que les valeurs proches qui se trouvent dans un même intervalle soient associées à des exemples de même classe.

Enfin, le choix du meilleur attribut joue un rôle primordial dans la construction des arbres de décision (cf. section 2.3, page 55). Pendant la construction d'un arbre de décision, une mesure de discrimination rend compte du pouvoir discriminant d'un attribut vis-à-vis de la classe pour un ensemble de exemples afin de choisir les attributs qui serviront à caractériser les individus. Une mesure de discrimination représente ainsi la différence entre les distributions par classes des exemples ayant une même valeur pour l'attribut en question et celle de la base d'apprentissage entière. Plus la mesure de discrimination est élevée, plus l'attribut est préféré pour servir en premier à la formation des nœuds dans l'arbre de décision. Autrement dit, l'attribut choisi est celui qui diminue le plus l'hétérogénéité de la base. Nous aborderons en détails plus tard le processus de construction d'arbres de décision et les mesures utilisées dans un tel processus.

Les mesures utilisées sont généralement des mesures issues de la théorie de l'information, en particulier l'entropie de Shannon, les entropies généralisées de Daróczy et de Rényi, ou des mesures issues de la statistique.

On considère alors donné un ensemble d'individus, qui constituent une base de

données $\xi = \{e_1, e_2..., e_N\}$, décrits par un ensemble d'attributs $\mathcal{A} = \{A_1, A_2.., A_K\}$ et qui appartiennent à une classe C d'un ensemble $\mathcal{C} = \{C_1, C_2, .., C_n\}$. L'attribut A_k prend sa valeur dans l'ensemble des m_k valeurs $\{v_{k_1}, v_{k_2}, .., v_{k_{m_k}}\}$. Dans le cas où il n'y a pas d'ambiguïté, on peut ignorer l'indice k dans v_{k_j}. Pour simplifier, on notera également A_k l'ensemble $\{v_{k_1}, v_{k_2}, .., v_{k_{m_k}}\}$. Dans un premier temps, nous considérons le cas où les valeurs v_{k_j} sont symboliques. On note enfin ξ_{C_i} l'ensemble des exemples appartenant à la classe C_i et ξ_{v_j} l'ensemble des exemples dont la valeur pour l'attribut A est v_j.

Supposons qu'un discriminateur (un classifieur, un attribut, une discrétisation) découpe l'ensemble en plusieurs sous-ensembles $\xi_1, \xi_2, .., \xi_m$. Le pouvoir de discrimination du discriminateur est généralement quantifié par la différence entre l'incertitude de l'ensemble ξ et la somme pondérée des incertitudes des ensembles $\xi_1, \xi_2, .., \xi_m$:

$$I(\xi) - \sum_{j=1}^{m} w_j I(\xi_j)$$

où I est souvent une mesure d'entropie et les w_j sont des poids. Cette partie est approfondie au cours de la thèse sous différents angles.

Motivation et objectif

La notion de discrimination en apprentissage est donc très importante. Les mesures d'entropie sont généralement utilisées comme mesures de discrimination dans la construction des arbres de décision. Cependant, la justification de l'adéquation de telles mesures pour ces processus reste à étudier.

Les justifications existantes sont généralement empiriques. Une bonne mesure de discrimination dans la construction des arbres de décision est heuristiquement le meilleur moyen de limiter la taille de l'arbre et de donner une cohérence sémantique aux nœuds qui le composent. En considérant un arbre de décision comme un recodage de la base d'apprentissage pour conserver l'information utile inhérente à cette base, l'usage de la théorie de l'information dans la construction des arbres de décision est justifiée. Mais de façon générale, la justification doit être faite de manière plus adéquate. Dans sa thèse [102], Marsala a recensé et analysé les propriétés exigées d'une mesure statistique qui rendent compte de la pertinence d'une caractéristique pour la sélection de caractéristiques dans le système de reconnaissance statistique. Il a décrit également la méthode de validation usuelle de Breiman *et al.* [27] qui exige un certain nombre de propriétés pour une mesure de pouvoir discriminant comme la minimalité, la maximalité, la symétrie, l'indépendance. Il a ensuite proposé un modèle hiérarchique [102, 131] pour valider de manière théorique et sémantique si une mesure est adéquate pour la construction des arbres de décision.

Dans le cadre de la construction des arbres de décision flous, le choix du meilleur attribut peut être effectué par un critère basé sur l'entropie floue [130] qui est une extension de l'entropie de Shannon aux événements flous ou la mesure d'ambiguïté de

classification proposée par Yuan et Shaw [162]. Plusieurs mesures telles que l'entropie d'ordre β de Daróczy et l'entropie d'ordre β de Rényi, ont été introduites pour la construction des arbres de décision [45]. Ces mesures sont étendues aux mesures floues et leur extension peut alors être employée pour la construction des arbres de décision flous en tant que mesure de discrimination floue.

L'objectif principal de ce chapitre est d'abord de justifier l'utilisation de plusieurs mesures de discrimination classiques dans le cadre du modèle hiérarchique introduit dans [102]. Grâce à ce modèle, nous pouvons introduire et justifier l'utilisation d'autres entropies et leurs formules conditionnelles existantes dans l'état de l'art mais rarement utilisées dans un tel processus, en particulier les entropies généralisées telles que l'entropie de Rényi et l'entropie de Daróczy.

Nous proposons ensuite un nouveau modèle de validation pour les mesures de discrimination floues défini comme une extension du modèle hiérarchique classique. La validation des mesures usuelles sera également présentée. Notre but ici n'est pas de comparer les mesures entre elles à la manière proposée par [108] (comparaison qui a d'ailleurs soulevé les critiques de [30]), mais de valider leur utilisation pour la construction d'un arbre.

Ce chapitre est organisé de la manière suivante. Dans la section 1.2, nous rappelons des travaux existants. Tout d'abord, nous établissons un état de l'art des mesures d'entropie et des mesures d'entropie conditionnelle. Nous décrivons ensuite le modèle hiérarchique pour les mesures de discrimination classiques. Dans la section 1.3, nous examinons, à l'aide du modèle hiérarchique, les mesures de discrimination classiques. Les mesures existantes sont caractérisées et introduites dans le processus de construction d'arbres de décision. Quelques discussions sur la généralisation des fonctions liées au modèle $\mathcal{FGH}$ et sur le rapport entre ce modèle et l'approche axiomatique sont présentées à la fin de cette section. Dans la section 1.4, nous proposons un nouveau modèle hiérarchique pour les mesures de discrimination floues. Dans la section 1.5, nous examinons des mesures de discrimination floues. Plusieurs mesures de discrimination floues sont validées par le modèle proposé pour l'usage dans la construction des arbres de décision flous. Dans la dernière section, nous concluons nos travaux concernant les modèles hiérarchiques.

1.2 État de l'art

1.2.1 Entropies et entropies conditionnelles

1.2.1.1 Introduction

La notion d'entropie est fondamentale en physique statistique. Elle apparaît en particulier dans la deuxième loi de la thermodynamique, ainsi qu'en mécanique statistique. En informatique, elle intervient dans la théorie de l'information, sous la forme célèbre de l'entropie de Shannon entre autres. Parmi les premières études sur cette entropie, on peut citer celle de Nyquist et Hartley (pendant les années 1920). Mais la vraie naissance de l'entropie est marquée par l'étude de Shannon. En fait,

la formule de Shannon est similaire à celle de Boltzmann en mécanique statistique mais Shannon a montré la signification de la formule comme une mesure d'information. Depuis, la notion d'entropie est devenue très importante en informatique théorique et appliquée, entre autres pour la transmission d'information (codage de source, codage de chaîne, détecteur d'erreur), pour l'inférence statistique, pour la cryptographie et pour l'algorithmique. L'entropie est utilisée comme mesure de la quantité d'information ou comme mesure d'incertitude.

Il existe plusieurs approches pour définir l'entropie : l'approche combinatoire, l'approche probabiliste, l'approche algorithmique et l'approche axiomatique que nous présentons brièvement ci-dessous. Associée à la notion d'entropie, la notion *d'entropie conditionnelle* représente l'entropie restante d'un événement si on connaît l'occurrence d'un autre. Cependant, toutes les entropies n'ont pas de forme conditionnelle.

L'approche combinatoire fournit une définition contextuelle de la quantité d'information d'un événement, c'est-à-dire dépendant du contexte dans lequel cet événement a lieu ou est considéré. Mais le poids (la contribution) de chaque événement au contexte n'est pas pris en compte. L'entropie de Hartley d'un événement e dans un contexte $\mathcal{X}$ se composant de n événements possibles est définie par[2] :

$$I(e) = \log n$$

Cette approche est la plus simple mais est plus ou moins naïve. Elle marque le premier pas dans la recherche de mesures d'information. Il n'existe pas de définition d'entropie conditionnelle associée. Actuellement, cette approche est rarement poursuivie.

L'approche algorithmique est introduite par Kolmogorov et Solomonoff [144]. Selon cette approche, on cherche la nature de la quantité d'information traitée à travers un ordinateur. Par exemple, la quantité d'information d'un événement e par rapport à une machine M est définie comme étant la longueur du plus court programme exécuté sur M pour calculer (c'est-à-dire décrire de façon algorithmique) e.

L'approche probabiliste fournit une définition de la quantité d'information d'un événement qui dépend à la fois du contexte et des poids des événements dans le contexte. Cette approche est la plus appropriée et sans doute la plus appliquée en apprentissage inductif. Entre autres, on peut citer l'entropie de Shannon [141], de Rényi [6, 132] et de Daróczy [49].

L'approche axiomatique semble s'attaquer en profondeur à la nature de la mesure d'information. On établit des axiomes désirés (ou propriétés désirées) pour une

[2]dans ce document, sauf mention contraire, $\log x$ est compris comme $\log_2 x$.

fonction mesurant la quantité d'information puis on cherche des fonctions qui vérifient ces axiomes. On peut distinguer deux sous-approches. La première considère a priori que la quantité d'information d'un événement est une fonction de sa probabilité. Alors, le système d'axiomes est constitué des contraintes imposées à une fonction définie sur des probabilités. C'est dans cette optique que l'entropie de Shannon a initialement été introduite [141]. Dans la deuxième approche, on vise à construire une fonction d'entropie sur l'ensemble des événements (sans considérer les probabilités). On peut définir une entropie conditionnelle de la même façon [15, 22, 50, 51].

1.2.1.2 Entropies

Dans cette section, un court état de l'art sur les mesures d'entropie est présenté. Nous nous intéressons aux mesures d'entropie potentiellement utilisables directement dans le processus d'apprentissage inductif. Les mesures développées dans l'approche probabiliste sont donc détaillées. Nous considérons d'abord les mesures d'entropie puis examinons les entropies conditionnelles.

Dans ce qui suit, on note $(P_1, P_2, .., P_n)$ une distribution de probabilité sur un ensemble d'événements : $\sum_{i=1}^{n} P_i = 1$ et $P_i \geq 0$. Dans le contexte de l'apprentissage, chaque événement correspond au fait qu'un exemple appartient à une classe déterminée et P_i est donc la probabilité de trouver un exemple de la classe C_i. Dans le contexte de l'apprentissage, au lieu de considérer un ensemble d'événements on peut considérer un ensemble d'exemples ξ et la distribution des exemples dans les classes. Pour simplifier, on utilise la même notation générique I pour l'entropie d'un ensemble d'événements $I(\xi)$, et pour l'entropie de la distribution de probabilité correspondante $I((P_1, P_2, .., P_n))$ ou plus simplement $I(P_1, P_2, .., P_n)$; de même pour l'entropie d'un événement $I(e)$ ou $I(P(e))$ s'il n'y a pas d'ambiguïté. Dans ce qui suit, $I(P_1, P_2, .., P_n)$ est l'entropie d'une distribution de probabilité et si ce n'est pas nécessaire, on ne précisera plus $\sum_{i=1}^{n} P_i = 1$ et $P_i \geq 0$. Les indices $S, R, D, R-norme$ sont utilisés pour différencier les entropies de Shannon, de Rényi, de Daróczy ou la R-norme entropie.

L'entropie de Shannon [141] est la plus connue et la plus appliquée. Elle définit d'abord la quantité d'information apportée par un événement : plus la probabilité d'un événement est faible (il est rare), plus la quantité d'information qu'il apporte est grande :

$$I_S(P_i) = -\log P_i$$

Elle mesure donc la surprise quand l'événement a lieu.

L'entropie d'un ensemble est ensuite définie comme la quantité d'information moyenne de ses éléments :

$$I_S(P_1, P_2, .., P_n) = -\sum_{i=1}^{n} P_i \log P_i \tag{1.1}$$

L'entropie de Hartley est un cas particulier de l'entropie de Shannon quand chaque événement a une probabilité égale à $\frac{1}{n}$.

L'entropie de Rényi (également nommée entropie d'ordre β de Rényi) a été proposée dans [6, 132] :

$$I_R^\beta(P_1, P_2, .., P_n) = \frac{1}{1-\beta} \log \sum_{i=1}^{n} P_i^\beta \tag{1.2}$$

où $\beta > 0$ et $\beta \neq 1$.

L'entropie de Daróczy (également nommée entropie d'ordre β de Daróczy) est définie par [49] (c'est aussi la définition de Tsallis en physique [1, 2]) :

$$I_D^\beta(P_1, P_2, .., P_n) = \frac{2^{\beta-1}}{2^{\beta-1}-1} \left(1 - \sum_{i=1}^{n} P_i^\beta\right) \tag{1.3}$$

où $\beta > 0$ et $\beta \neq 1$.

L'indice de diversité de Gini, aussi appelé l'entropie quadratique, est un coefficient statistique, utilisé par exemple dans [27]. C'est un cas particulier de l'entropie de Daróczy quand $\beta = 2$.

Daróczy et Rényi n'ont pas défini la quantité d'information d'un seul événement comme l'a fait Shannon, ils n'ont défini que l'entropie d'un ensemble d'événements en connaissant les probabilités de chaque événement.

La R-norme entropie a été proposée par Boekee et Van der Lubbe (1980) et abordée dans [86]. Dans la formule suivante, le coefficient originel R (R dans le terme $R - norme$) est noté β pour des raisons d'homogénéité de nos notations.

$$I_{R-norme}^\beta(P_1, P_2, .., P_n) = \frac{\beta}{\beta-1} \left(1 - \left(\sum_{i=1}^{n} P_i^\beta\right)^{\frac{1}{\beta}}\right) \tag{1.4}$$

où $\beta > 0$ et $\beta \neq 1$.

Propriété : Ces mesures peuvent également être retrouvées selon l'approche axiomatique. Les axiomes pour déterminer les entropies se trouvent dans [2, 5]. Grâce aux systèmes d'axiomes, l'existence de chacune de ces mesures est justifiée par la satisfaction des propriétés exigées. Il peut arriver que plusieurs systèmes d'axiomes caractérisent une seule entropie. Les propriétés de ces entropies sont recensées dans [6, 49, 132] et surtout dans l'article d'Aczel [5].

Par ailleurs, ces entropies possèdent la propriété :

$$\begin{aligned} \lim_{\beta \to 1} I_R^\beta(P_1, P_2, .., P_n) &= \lim_{\beta \to 1} I_D^\beta(P_1, P_2, .., P_n) \\ &= \lim_{\beta \to 1} I_{R-norme}^\beta(P_1, P_2, .., P_n) \\ &= I_S(P_1, P_2, .., P_n) \end{aligned}$$

Pour simplifier, on étend la définition des entropies de Rényi et de Daróczy (qui sont initialement définies pour $\beta \neq 1$) pour $\beta = 1$ en notant $I_R^1 = I_S$, $I_D^1 = I_S$ et $I_{R-norme}^1 = I_S$.

L'entropie asymétrique et l'entropie décentrée

Les entropies présentées ci-dessus proviennent de la théorie de l'information. Nous évoquons dans la suite une famille d'entropies récemment introduite en apprentissage inductif.

Une des caractéristiques communes des entropies présentées ci-dessus concerne la maximalité. L'incertitude maximale correspond à la distribution uniforme :

$$0 = I(0,..,0,1,0,..,0) \leq I(P_1, P_2, .., P_n) \leq I\left(\frac{1}{n}, \frac{1}{n}, .., \frac{1}{n}\right) \tag{1.5}$$

où I est une des entropies abordées ci-dessus.

Cette propriété est fondée sur l'hypothèse que l'incertitude ne dépend que de la probabilité des événements. Cette hypothèse est valable dans le contexte de la théorie de l'information, dans laquelle les mesures ont été proposées initialement.

Dans une optique différente, directement reliée à la recherche d'une mesure adéquate utilisée dans l'induction d'arbres de décision, Zighed *et al.* [100] ont proposé une entropie asymétrique. L'asymétrie signifie que la distribution pour laquelle la mesure est maximale n'est pas forcément la distribution uniforme. En pratique, cette distribution est choisie par les utilisateurs en fonction d'un coût d'erreur et de la distribution des exemples dans la base d'apprentissage. Les propriétés requises pour l'entropie asymétrique sont la minimalité, la maximalité et la concavité.

Soit $\theta = (\theta_1, \theta_2, .., \theta_n)$ $(\sum_{i=1}^{n} \theta_i = 1)$ la distribution pour laquelle la mesure d'entropie est maximale. En supposant qu'il existe une fonction rationnelle sous la forme du rapport entre une fonction du second degré et une fonction linéaire, qui vérifie les propriétés requises, la fonction suivante est trouvée :

$$I_{asym}(P_1, P_2, .., P_n) = -\sum_{i=1}^{n} \frac{P_i(1 - P_i)}{(-2\theta_i + 1)P_i + \theta_i^2} \tag{1.6}$$

Dans des travaux plus récents [168], les même auteurs étendent leur approche en introduisant une nouvelle propriété, sur la consistance de l'entropie, et ainsi relâchent la propriété de maximalité. La consistance d'une mesure d'entropie signifie que la mesure prend en compte la taille de l'échantillon : pour une même distribution fréquentielle, l'entropie devrait être plus faible sur un effectif plus grand. Cette propriété tient compte de la précision de l'estimation de la distribution réelle par la distribution empirique. La propriété de minimalité exige simplement que l'entropie d'une variable certaine tende vers 0 quand la taille de l'échantillon devient grand. L'objectif visé est d'aboutir à une entropie que l'on pourrait qualifier d'empirique qui tienne mieux compte des caractéristiques pratiques liées au problème d'apprentissage inductif par arbres de décision. Pour cela, les fréquences théoriques P_i sont évaluées par leur estimateur de Laplace qui est légèrement différent de l'estimation usuelle :

$$P_i = \frac{|\xi_{C_i}| + 1}{N + n}$$

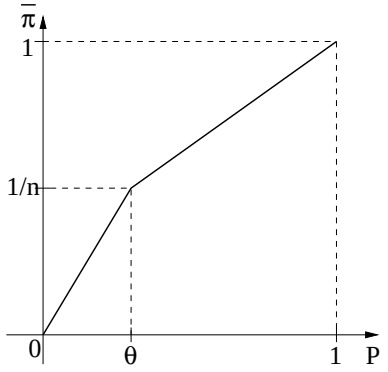

FIG. 1.1 – Fonction de transformation de P à π'

où N est le nombre d'exemples, n est le nombre de classes et $|\xi_{C_i}|$ est le nombre d'exemples de classe C_i.

Au lieu de donner une seule entropie décentrée, Lallich *et al.* [92] proposent une technique plus générale qui s'adapte à n'importe quel type d'entropies, que ce soit une entropie de Shannon, une entropie de Daróczy ou d'autres. Comme l'entropie asymétrique, l'entropie décentrée prend sa valeur maximale sur une distribution de probabilité $(\theta_1, \theta_2, .., \theta_n)$ définie par les utilisateurs et qui n'est pas obligatoirement uniforme. Cette technique s'appuie sur une transformation de variables : l'entropie est une fonction définie à travers des variables $\pi_1, \pi_2, .., \pi_n$ et non directement sur $P_1, P_2, .., P_n$. Pour être analogues à des fréquences de la distribution de référence, les π_i doivent vérifier :

1. $0 \leq \pi_i \leq 1$
2. $\sum_{i=1}^{n} \pi_i = 1$

La transformation doit satisfaire :

1. π_i passe de 0 à $1/n$, lorsque P_i passe de 0 à θ_i.
2. π_i passe de $1/n$ à 1, lorsque P_i passe de θ_i à 1.

Considérons une fonction linéaire par morceaux qui satisfait ces deux dernières contraintes (cf. figure 1.1) :

$$\pi'_i = \begin{cases} \dfrac{P_i}{n\theta_i} & \text{si } 0 \leq P_i \leq \theta_i \\ \dfrac{n(P_i - \theta_i) + 1 - P_i}{n(1 - \theta_i)} & \text{si } \theta_i < P_i \leq 1 \end{cases} \tag{1.7}$$

Pour résoudre le problème concernant la condition de normalisation, qui n'est automatiquement vérifiée que pour le cas de deux classes ($n = 2$), il suffit de normaliser les π_i :

$$\pi_i = \frac{\pi'_i}{\sum_{i=1}^{n} \pi'_i} \tag{1.8}$$

L'entropie décentrée pour la distribution $(P_1, P_2, .., P_n)$ est :

$$I_{dc}(P_1, P_2, .., P_n) = I_c(\pi_1, \pi_2, .., \pi_n)$$

où I_c est une entropie centrée, comme par exemple l'entropie de Shannon, l'entropie de Rényi, l'entropie de Daróczy ou la R-norme entropie. Le décentrage des entropies généralisées est détaillé dans [92]. Malgré des propriétés communes, l'entropie asymétrique proposée par Zighed *et al.* est différente des entropies décentrées construites par Lallich *et al.* dans [92].

1.2.1.3 Entropies conditionnelles

L'entropie conditionnelle quantifie l'entropie restante provenant d'une variable aléatoire, si l'on connaît la valeur d'une seconde variable aléatoire.

En apprentissage, l'entropie $I(\xi|A)$ de ξ conditionnée par les valeurs de l'attribut A est généralement définie comme une somme pondérée des mesures conditionnées par chacune des valeurs de A :

$$I(\xi|A) = \sum_{j=1}^{m} w_j I(\xi|v_j) = \sum_{j=1}^{m} w_j I(\xi_{v_j}) \tag{1.9}$$

C'est l'entropie de la base conditionnée par la valeur v_j de l'attribut A. Cette mesure quantifie le pouvoir de discrimination de la valeur v_j pour la base d'apprentissage ξ. Les w_j sont les poids qui permettent de caractériser différents types de mesures [45]. La valeur de w_j est souvent croissante en fonction de $P(v_j)$.

À partir de la littérature, nous proposons une typologie pour les définitions des entropies conditionnelles. Cette typologie permet de retrouver les entropies conditionnelles classiques, de plus l'application du Type 4 à l'entropie de Shannon correspond au gain ratio. Nous désignons par Type 1, Type 2, Type 3 et Type 4 les formules conditionnelles suivantes :

Type 1 :

$$I(\xi|A) = \sum_{j=1}^{m} P(v_j) I_\beta(\xi|v_j) \tag{1.10}$$

Type 2 :

$$I(\xi|A) = \sum_{j=1}^{m} P^\beta(v_j) I_\beta(\xi|v_j) \tag{1.11}$$

Type 3 :

$$I(\xi|A) = \sum_{j=1}^{m} \frac{P^\beta(v_j)}{\sum_{k=1}^{m} P^\beta(v_k)} I_\beta(\xi|v_j) \tag{1.12}$$

Type 4 :

$$I(\xi|A) = \sum_{j=1}^{m} \frac{P(v_j)}{-\sum_{k=1}^{m} P(v_k) \log P(v_k)} I_\beta(\xi|v_j) \tag{1.13}$$

En fait, dans la littérature, il existe des formes de Type 1 [159], de Type 2 [49], de Type 3 [3] pour l'entropie conditionnelle de Daróczy et seulement de Type 1 pour l'entropie conditionnelle de Rényi [6, 132]. Remarquons que dans son article original [49], Daróczy n'a introduit que la forme de Type 2 pour le conditionnement. La formule (1.13) n'est pas utilisée pour une entropie dans l'état de l'art. Cependant, elle a été introduite pour le gain ratio [124]. Parmi les formes ci-dessus, seules la première et la troisième vérifient la propriété : la somme des poids des entropies est égale à 1.

Pour les pondérations de Type 2 et de Type 3, quand β est grand, l'entropie conditionnelle dépend essentiellement de l'entropie issue de l'événement le plus fréquent (autrement dit la valeur la plus fréquente). Plus β est petit, plus les événements (des valeurs) sont considérés de manière égale. La pondération de Type 4 a été conçue pour défavoriser les attributs ayant plusieurs valeurs.

Étant donné une entropie, nous proposons d'étudier les entropies conditionnelles obtenues par chacune des formules conditionnelles pour le processus d'induction. Ceci nous mène, en particulier, à utiliser l'entropie conditionnelle de Daróczy du Type i, l'entropie conditionnelle de Rényi du Type i, la R-norme entropie du Type i ($i = 1, 2, 3, 4$) et l'entropie conditionnelle de Shannon pour choisir des attributs adéquats pendant la construction des arbres de décision. En général, on peut aussi combiner l'entropie de Shannon avec tous les types d'entropie conditionnelle ci-dessus. Il faut noter que, dans la littérature, pour définir une nouvelle entropie conditionnelle à partir d'une entropie donnée, les propriétés exigées pour une mesure conditionnelle sont rarement étudiées. Les modèles proposés par Kampé de Fériet [50, 51] et celui proposé par Benvenuti [15] peuvent être cités comme exemples de modèles qui permettent une telle étude.

Nous validerons par la suite l'utilisation de ces mesures dans le processus d'induction par le modèle hiérarchique, appelé $\mathcal{FGH}$. On verra que ce sont des cas particuliers des opérations d'agrégation au niveau $\mathcal{H}$ du modèle et que d'autres agrégations sont également possibles.

Notons également que dans la théorie de l'information, à partir d'une mesure d'entropie de variables aléatoires, on peut définir une mesure d'entropie conditionnelle liée à deux variables comme la différence entre l'entropie de leur conjonction et celle de la variable conditionnée. $I(X|Y) = I(XY) - I(Y)$ où X, Y sont des variables aléatoires.

Cette stratégie a été utilisée pour définir l'entropie conditionnelle de Shannon [141]. Avec cette approche, on retrouve l'entropie conditionnelle de Shannon Type 1 et l'entropie conditionnelle de Daróczy de Type 2. Quand I est l'entropie de Rényi, aucune des formules conditionnelles énumérées ci-dessus n'est retrouvée.

1.2.2 Modèle hiérarchique pour les mesures de discrimination classiques

Lors de la construction d'un arbre de décision, la mesure utilisée pour choisir le meilleur attribut doit satisfaire un certain nombre de propriétés. Dans [102, 131], un modèle hiérarchique pour les mesures de discrimination classiques a été proposé afin de vérifier si une mesure donnée est adéquate pour un processus de construction d'arbres de décision.

Ce modèle hiérarchique, appelé modèle $\mathcal{FGH}$, se compose des définitions de 3 types de fonctions organisés de manière hiérarchique en 3 niveaux : $\mathcal{F}$-fonction, $\mathcal{G}$-fonction et $\mathcal{H}$-fonction. Ils correspondent respectivement à 3 niveaux imbriqués : niveau $\mathcal{F}$, niveau $\mathcal{G}$ et niveau $\mathcal{H}$. Les fonctions du niveau supérieur sont des fonctions agrégeant des fonctions du niveau inférieur. À chaque niveau, un ensemble de propriétés est imposé aux fonctions. En pratique, une fonction au niveau $\mathcal{F}$ peut être une entropie d'un événement qui quantifie la quantité d'information relative à l'évènement ; une fonction du niveau $\mathcal{G}$ peut être une entropie qui évalue la quantité d'information moyenne d'un ensemble d'évènements ; et une fonction du niveau $\mathcal{H}$ peut être une entropie conditionnelle qui mesure l'entropie restante d'un ensemble d'événements relativement à un autre. Ces 3 niveaux servent également à construire des mesures de discrimination en construisant des fonctions du niveau le plus bas $\mathcal{F}$ au niveau le plus élevé $\mathcal{H}$. La validation d'une mesure revient à vérifier qu'il existe une fonction à chacun de ces 3 niveaux pour la mesure en question.

Pour que les définitions soient générales, on considèrera dans les définitions un ensemble d'objets $\mathcal{X}$. Notons $\mathcal{S}[\mathcal{X}]$ l'ensemble des sous-ensembles de $\mathcal{X}$ et $\mathbb{P}[\mathcal{X}]$ l'ensemble des partitions de $\mathcal{X}$. En apprentissage inductif, $\mathcal{X}$ est l'ensemble des exemples dans une base ξ. La définition de chaque niveau du modèle $\mathcal{FGH}$ est décrite comme suit (l'étude formelle et détaillée du modèle se trouve dans [102]).

1.2.2.1 Niveau $\mathcal{F}$

Définition 1.2.1. ($\mathcal{F}$-fonction). *Une $\mathcal{F}$-fonction est une fonction F :*

$$F : \mathcal{S}[\mathcal{X}] \times \mathcal{S}[\mathcal{X}] \to \mathbb{R}^+$$

telle que :

1. *$F(U, V)$ est minimum quand $U \subseteq V$.*
2. *$F(U, V)$ est maximum quand $U \cap V = \emptyset$.*
3. *$F(U, V)$ est strictement décroissante avec $U \cap V$ c'est-à-dire : si $U \cap V_1 \subset U \cap V_2$ alors $F(U, V_1) > F(U, V_2)$.*

Le niveau $\mathcal{F}$ concerne les fonctions utiles pour mesurer l'inadéquation entre 2 ensembles, c'est-à-dire qu'il concerne les fonctions qui mesurent la non-inclusion de deux ensembles U et V. Dans le cadre de l'apprentissage inductif, une telle fonction est par exemple appliquée à l'ensemble ξ_{C_i} des exemples de classe C_i et à l'ensemble ξ_{v_j} des exemples ayant v_j comme valeur d'un attribut A. Par abus de notation, on

note également $F(C_i|v_j)$, une fonction d'événements conditionnels. Cette fonction est définie par :

$$F(C_i|v_j) = F(\xi_{v_j}, \xi_{C_i})$$

Si les deux sous-ensembles sont disjoints, c'est-à-dire, si aucun exemple de classe C_i n'a v_j comme valeur pour A, alors v_j n'est pas adéquate pour reconnaître la classe C_i et F atteint sa valeur maximale. Si tous les exemples ayant v_j pour valeur de l'attribut A ont la classe C_i, alors la valeur v_j est totalement adéquate pour la reconnaissance de la classe C_i. La valeur v_j induit alors l'appartenance à la classe C_i pour un exemple. Plus les exemples d'une classe appartiennent à ξ_{v_j}, plus la valeur v_j est compatible avec la classe.

1.2.2.2 Niveau $\mathcal{G}$

Soit une $\mathcal{F}$-fonction F et une suite de fonctions réelles continues $g_k : \mathbb{R}^k \longrightarrow \mathbb{R}^+$, $k \in \mathbb{N}$.

Définition 1.2.2. ($\mathcal{G}$-fonction). *Une $\mathcal{G}$-fonction est une fonction G :*

$$G : \mathcal{S}[\mathcal{X}] \times \mathbb{P}[\mathcal{X}] \to \mathbb{R}^+$$

telle que :

$$G(U, \mathcal{V}) = g_n\left(F(U, V_1), F(U, V_2), .., F(U, V_n)\right) \tag{1.14}$$

où $\mathcal{V} = \{V_1, V_2, .., V_n\}$ est une partition de $\mathcal{X}$ et :

1. *$G(U, \mathcal{V})$ est minimum quand il existe V_i $(1 \leq i \leq n)$ tel que $U \subseteq V_i$.*
2. *$G(U, \mathcal{V})$ est maximum quand $F(U, V_1) = F(U, V_2) = .. = F(U, V_n)$.*

Le niveau $\mathcal{G}$ concerne les fonctions agrégeant des fonctions du niveau $\mathcal{F}$. En apprentissage, ces fonctions quantifient le pouvoir discriminant d'une valeur v_j de l'attribut A relativement à l'ensemble des classes. Elles sont notées dans le contexte de l'apprentissage par $G(\mathcal{C}|v_j)$ et :

$$G(\mathcal{C}|v_j) = G\left(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}\right)$$

Si v_j est adéquate à toutes les classes à même degré, la présence de v_j n'est pas significative pour l'identification de la classe d'un exemple et G va prendre sa valeur maximale. Au contraire, dans le cas où la valeur v_j est adéquate à une seule classe, la valeur v_j implique directement cette classe et G va prendre sa valeur minimale.

1.2.2.3 Niveau $\mathcal{H}$

Soit une $\mathcal{G}$-fonction G et une suite de fonctions réelles continues $h_k : \mathbb{R}^{+k} \longrightarrow \mathbb{R}^+$, $k \in \mathbb{N}$.

Définition 1.2.3. ($\mathcal{H}$-fonction). *Une $\mathcal{H}$-fonction est une fonction H :*

$$H : \mathbb{P}[\mathcal{X}] \times \mathbb{P}[\mathcal{X}] \rightarrow \mathbb{R}^+$$

telle que :

$$H(\mathcal{U}, \mathcal{V}) = h_m \left(G(U_1, \mathcal{V}), G(U_2, \mathcal{V}), .., G(U_m, \mathcal{V}) \right)$$

où $\mathcal{U} = \{U_1, U_2, .., U_m\}$ et $\mathcal{V}$ sont des partitions de $\mathcal{X}$.

Le niveau $\mathcal{H}$ regroupe les fonctions agrégeant des fonctions du niveau $\mathcal{G}$. En apprentissage, ces fonctions sont utilisées pour mesurer le pouvoir discriminant d'un attribut A relativement à l'ensemble des classes. Elles sont notées dans ce contexte par $H(\mathcal{C}|A)$.

$$H(\mathcal{C}|A) = H\left(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\} \right)$$

1.3 Étude des mesures de discrimination par le modèle hiérarchique

Cette section est consacrée à la validation des mesures de discrimination à l'aide du modèle hiérarchique classique. Cela nous permet de justifier l'usage de ces mesures en apprentissage inductif. La validation présentée dans cette section sera complétée plus tard par une étude expérimentale décrite dans le chapitre 2 (page 49). Cette section se termine par une étude sur le rapport entre le modèle hiérarchique et l'axiomatisation des entropies.

Comme le modèle $\mathcal{FGH}$ se compose de 3 niveaux et chacun correspond à un type de fonctions, pour valider une mesure, il est donc nécessaire de retrouver les fonctions de ces 3 niveaux de la mesure. Il faut évidement justifier que ces fonctions satisfont les propriétés demandées. Une mesure sera donc validée s'il est possible de mettre en évidence une $\mathcal{F}$-fonction, une $\mathcal{G}$-fonction et une $\mathcal{H}$-fonction pour elle. Les validations de certaines mesures de discrimination par le modèle $\mathcal{FGH}$ ont été présentées dans [102] (pour l'entropie de Shannon, l'indice de diversité de Gini, la mesure d'information de Kampé de Fériet).

Dans la suite de cette section, nous continuons sur cette voie pour étudier en détails les mesures de discrimination notamment les mesures d'entropie autre que celle de Shannon. En particulier, les mesures d'entropie de Rényi, celle de Daróczy, la R-norme entropie, l'entropie symétrique et l'entropie décentrée sont validées comme des mesures de discrimination. De plus, nous considérons également la mesure de Kolmogorov-Smirnov. Pour cela, nous exhibons la $\mathcal{F}$-fonction, la $\mathcal{G}$-fonction et la $\mathcal{H}$-fonction qui leur correspondent.

1.3.1 Entropie de Rényi, entropie de Daróczy, R-norme entropie

Considérons une entropie définie sur la distribution de probabilité conditionnée par v_j $(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j))$:

$$P(C_i|v_j) = \frac{|\xi_{v_j} \cap \xi_{C_i}|}{|\xi_{v_j}|}$$

Niveau $\mathcal{F}$

Considérons pour $\beta > 0$ la fonction suivante :

$$F(\xi_{v_j}, \xi_{C_i}) = -\beta \log \frac{|\xi_{v_j} \cap \xi_{C_i}|}{|\xi_{v_j}|} \quad \text{soit aussi} \quad F(C_i|v_j) = -\beta \log P(C_i|v_j)$$

Cette formule généralise l'entropie de Shannon par le coefficient β. Elle sera utilisée à la fois pour l'entropie de Rényi, pour l'entropie de Daróczy et pour la R-norme entropie.

F est une $\mathcal{F}$-fonction car :

1. Si $\xi_{v_j} \subseteq \xi_{C_i}$ alors $P(C_i|v_j) = 1$ donc $\log P(C_i|v_j) = 0$.
2. Si $\xi_{v_j} \cap \xi_{C_i} = \emptyset$ alors $P(C_i|v_j) = 0$ donc $-\log P(C_i|v_j) = +\infty$.
3. $F(\xi_{v_j}, \xi_{C_i})$ est une fonction strictement décroissante de $|\xi_{C_i} \cap \xi_{v_j}|$ quand $|\xi_{v_j}|$ est constante.

Niveau $\mathcal{G}$

Entropie de Rényi

Considérons l'agrégation suivante des $\mathcal{F}$-fonctions :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \frac{1}{1-\beta} \left(\log \sum_{i=1}^{n} 2^{-F(\xi_{v_j}, \xi_{C_i})} \right)$$

soit aussi :

$$G(\mathcal{C}|v_j) = \frac{1}{1-\beta} \left(\log \sum_{i=1}^{n} 2^{-F(C_i|v_j)} \right)$$

Remplaçons $F(C_i|v_j)$ par $-\beta \log P(C_i|v_j)$ on retrouve la formule de l'entropie de Rényi (1.2) :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = I_R^{\beta}(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j))$$

G satisfait bien les propriétés requises :

1. $G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = 0$, c'est la valeur minimale quand il existe C_i $(1 \leq i \leq n)$ telle que $\xi_{v_j} \subseteq \xi_{C_i}$ car : $P(C_i|v_j) = 1$ et $\forall l \neq i \ : P(C_l|v_j) = 0$ (sachant que $\{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}$ $(1 \leq i \leq n)$ est une partition de ξ donc $\sum_{i=1}^{n} P(C_i|v_j) = 1$).
2. Si $F(\xi_{v_j}, \xi_{C_1}) = F(\xi_{v_j}, \xi_{C_2}) = .. = F(\xi_{v_j}, \xi_{C_n})$, c'est-à-dire $\forall i, l : 1 \leq i \neq l \leq n \ : P(C_i|v_j) = P(C_l|v_j)$, $G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$ atteint sa valeur maximale d'après (1.5).

Entropie de Daróczy

Considérons l'agrégation suivante des $\mathcal{F}$-fonctions :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \frac{2^{\beta-1}}{2^{\beta-1}-1}\left(1 - \sum_{i=1}^{n} 2^{-F(\xi_{v_j}, \xi_{C_i})}\right)$$

soit aussi :

$$G(\mathcal{C}|v_j) = \frac{2^{\beta-1}}{2^{\beta-1}-1}\left(1 - \sum_{i=1}^{n} 2^{-F(C_i|v_j)}\right)$$

Remplaçons $F(C_i|v_j)$ par $-\beta \log P(C_i|v_j)$ on retrouve la formule de l'entropie de Daróczy (1.3) :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = I_D^{\beta}(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j))$$

De même que pour l'entropie de Rényi et d'après (1.5), G est maximale dans le cas où les $\mathcal{F}-$fonctions qui la composent sont équivalentes. G est minimale quand ξ_{v_j} est incluse dans l'un des sous-ensembles de la partition $\{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}$.

R-norme entropie

Considérons l'agrégation suivante des $\mathcal{F}$-fonctions :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \frac{\beta}{\beta-1}\left(1 - \left(\sum_{i=1}^{n} 2^{-F(\xi_{v_j}, \xi_{C_i})}\right)^{\frac{1}{\beta}}\right)$$

soit aussi :

$$G(\mathcal{C}|v_j) = \frac{\beta}{\beta-1}\left(1 - \left(\sum_{i=1}^{n} 2^{-F(C_i|v_j)}\right)^{\frac{1}{\beta}}\right)$$

Remplaçons $F(C_i|v_j)$ par $-\beta \log P(C_i|v_j)$ on retrouve la formule (1.4) de la R-norme entropie :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = I_{R-norme}^{\beta}(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j))$$

De même que les entropies de Rényi et de Daróczy ci-dessus, cette fonction vérifie les deux propriétés liées à la minimalité et de la maximalité exigées pour une $\mathcal{G}$-fonction.

Notons que, entre autres, la somme ou le produit des $\mathcal{G}$-fonctions sont eux-mêmes des $\mathcal{G}$-fonctions. Cette propriété permet de construire de nouvelles $\mathcal{G}$-fonctions en combinant les fonctions existantes.

Niveau $\mathcal{H}$

Considérons l'agrégation suivante des $\mathcal{G}$-fonctions, qui donne une fonction au niveau $\mathcal{H}$:

$$H(\{\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} w_j G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$$

soit aussi :

$$H(\mathcal{C}|A) = \sum_{j=1}^{m} w_j G(\mathcal{C}|v_j)$$

Les formules conditionnelles de Type 1, 2, 3 et 4 sont retrouvées quand on affecte respectivement les valeurs

$$P(v_j), \quad P^{\beta}(v_j), \quad \frac{P^{\beta}(v_j)}{\sum_{k=1}^{m} P^{\beta}(v_k)}, \quad \frac{P(v_j)}{-\sum_{k=1}^{m} P(v_k) \log P(v_k)}$$

à w_j pour les poids et remplace $G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$ par l'une des entropies. De la même manière, le gain ratio, autrement dit l'entropie de Shannon de Type 4 d'après notre taxonomie, peut être validé comme une mesure de discrimination selon le modèle hiérarchique. Comme aucune propriété n'est imposée au niveau $\mathcal{H}$, le choix de l'opération d'agrégation est relativement libre. Cela peut être exploité pour obtenir des arbres avec des propriétés souhaitées. En particulier, le choix du Type 4 permet de réduire l'effet de préférence des attributs ayant plusieurs valeurs [124].

1.3.2 Entropie asymétrique et entropie décentrée

Considérons une entropie décentrée I_{dc} définie sur la distribution de probabilité $(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j))$. I_{dc} est obtenue par un décentrage d'une entropie I_c, dite *centrée* (pour distinguer des entropies décentrées), en particulier une des entropies considérées ci-dessus. À la différence de l'entropie centrée, cette entropie atteint sa valeur maximale quand $P(C_i|v_j) = \theta_i \geq 0 \ \ \forall i = 1..n$ et : $\sum_{i=1}^{n} \theta_i = 1$. À partir des probabilités $P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j)$, on calcule les $\pi_{1j}, \pi_{2j}, .., \pi_{nj}$ selon les formules (1.7) et (1.8). Les fonctions à chaque niveau sont définies à partir des $\pi_{1j}, \pi_{2j}, .., \pi_{nj}$ et ainsi elles sont indirectement définies à partir des probabilités $P(C_1|v_j)$, $P(C_2|v_j)$,.., $P(C_n|v_j)$.

Niveau $\mathcal{F}$

Considérons la fonction suivante :

$$F(\xi_{v_j}, \xi_{C_i}) = -\log \pi_{ij} \quad \text{soit aussi} \quad F(C_i|v_j) = -\log \pi_{ij}$$

F est une $\mathcal{F}$-fonction car :

1. Si $\xi_{v_j} \subseteq \xi_{C_i}$ alors $P(C_i|v_j) = 1$ donc $\pi_{ij} = 1$ et $F(\xi_{v_j}, \xi_{C_i}) = -\log \pi_{ij} = 0$.
2. Si $\xi_{v_j} \cap \xi_{C_i} = \emptyset$ alors $P(C_i|v_j) = 0$ donc $\pi_{ij} = 0$ et $F(\xi_{v_j}, \xi_{C_i}) = -\log \pi_{ij} = +\infty$.
3. $F(\xi_{v_j}, \xi_{C_i})$ est une fonction strictement décroissante de π_{ij} et à son tour π_{ij} elle-même est une fonction strictement croissante de $P(C_i|v_j)$ (figure 1.1, page 20). Donc $F(\xi_{v_j}, \xi_{C_i})$ est une fonction décroissante de $P(C_i|v_j)$ ou de façon équivalente : décroissante de $|\xi_{C_i} \cap \xi_{v_j}|$ quand $|\xi_{v_j}|$ est constante.

Cette $\mathcal{F}$-fonction est également valable pour l'entropie asymétrique.

Niveau $\mathcal{G}$

Entropie décentrée

Considérons la fonction suivante :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = I_{dc}(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j)) \tag{1.15}$$

soit aussi

$$G(\mathcal{C}|v_j) = I_{dc}(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j))$$

avec

$$I_{dc}(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j)) = I_c(\pi_{1j}, \pi_{2j}, .., \pi_{nj})$$

Pour faire apparaître les $\mathcal{F}$-fonctions, on peut réécrire la formule (1.15) ci-dessus comme suit :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = I_c(e^{-F(C_1|v_j)}, e^{-F(C_2|v_j)}, .., e^{-F(C_n|v_j)})$$

On démontre que G satisfait bien les propriétés requises :

1. Quand il existe C_i $(1 \leq i \leq n)$ telle que $\xi_{v_j} \subseteq \xi_{C_i}$ on a : $P(C_i|v_j) = 1$ et $P(C_l|v_j) = 0,\ \forall l \neq i$. Selon la transformation (figure 1.1) on a : $\pi_{ij} = 1$ et $\pi_{lj} = 0,\ \forall l \neq i$. Dans ce cas :

   $$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = I_c(0, 0, .., 1, .., 0) = 0$$

   C'est la valeur minimale de G.

2. Si $F(\xi_{v_j}, \xi_{C_1}) = F(\xi_{v_j}, \xi_{C_2}) = .. = F(\xi_{v_j}, \xi_{C_n})$, c'est-à-dire $\pi_{1j} = \pi_{2j} = .. = \pi_{nj} = \frac{1}{n}$, alors I_c atteint sa valeur maximale car elle est une entropie centrée. Aussi $G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$ atteint également sa valeur maximale. Cette valeur maximale correspond à l'entropie décentrée de la distribution $(\theta_1, \theta_2, .., \theta_n)$.

Entropie asymétrique

La $\mathcal{G}$-fonction est la formule (1.6) définie sur les probabilités $P(C_1|v_j)$, $P(C_2|v_j)$,.., $P(C_n|v_j)$:

$$I_{asym}(P(C_1|v_j), P(C_2|v_j), .., P(C_n|v_j)) = -\sum_{i=1}^{n} \frac{P(C_i|v_j)(1 - P(C_i|v_j))}{(-2\theta_i + 1)P(C_i|v_j) + \theta_i^2}$$

C'est une agrégation de $\mathcal{F}$-fonctions car les $\mathcal{F}$-fonctions sont bijectives et décroissantes des $P(C_i|v_j)$. La maximalité et la minimalité sont vérifiées de manière identique à l'entropie décentrée :

$$I_{asym}(0, .., 1, .., 0) \leq I_{asym}(P(C_1|v_j), .., P(C_n|v_j)) \leq I_{asym}(\theta_1, \theta_2, .., \theta_n)$$

Niveau $\mathcal{H}$

Les $\mathcal{H}$-fonctions pour les entropies décentrées et les entropies asymétriques sont identiques à celles définies pour les entropies centrées et décrites dans la section 1.3.1.

1.3.3 Mesure de Kolmogorov-Smirnov

La mesure de Kolmogorov-Smirnov n'est pas une mesure d'entropie. Elle a été introduite dans le cadre statistique pour la sélection de test. L'usage de cette mesure dans la construction des arbres de décision est décrit dans [66, 152]. Nous montrons ici les $\mathcal{F}$-fonction, $\mathcal{G}$-fonction et $\mathcal{H}$-fonction qui prouvent son appartenance au modèle $\mathcal{FGH}$.

Considérons le cas de 2 classes C_1 et C_2. Considérons l'attribut A qui possède un ensemble de valeurs possibles $v_1, v_2, ..., v_m$. La distance de Kolmogorov-Smirnov pour un test $A = v_j$ est :

$$K(\{C_1, C_2\}|A = v_j) = |P(C_1|v_j) - P(C_2|v_j)|$$

Notons $\mathcal{K}(\{C_1, C_2\}|A)$ la valeur maximale des distances de Kolmogorov-Smirnov pour chacune des valeurs de A :

$$\mathcal{K}(\{C_1, C_2\}|A) = \max_{j=1..m} K(\{C_1, C_2\}|A = v_j)$$

La valeur v_j associée à la valeur maximale de $\mathcal{K}(\{C_1, C_2\}|A)$ sera utilisée pour le test avec l'attribut A s'il est choisi.

Parmi tous les attributs, l'attribut choisi pour le test est celui qui maximise $\mathcal{K}(\{C_1, C_2\}|A)$ [152]. En d'autres termes, le choix d'un test est effectué parmi tous les tests possibles de la forme : $A = v$ et celui qui est choisi maximise la distance de Kolmogorov-Smirnov qui lui est associée. Notons que les arbres de décision construits selon cette mesure sont des arbres binaires dont chaque nœud interne correspond à un test de la forme de $A = v$.

Nous montrons maintenant les fonctions qui correspondent à chaque niveau du modèle hiérarchique.

La $\mathcal{F}$-fonction associée est définie par :

$$F(\xi_{v_j}, \xi_{C_i}) = 1 - P(C_i|v_j)$$

La satisfaction des propriétés exigées pour une $\mathcal{F}$-fonction peut être démontrée de la même manière que pour les $\mathcal{F}$-fonctions précédentes.

La $\mathcal{G}$-fonction associée est définie par :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}\}) = 1 - |P(C_1|v_j) - P(C_2|v_j)|$$

1. $G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}\}) = 0$, c'est la valeur minimale, quand il existe C_i telle que $\xi_{v_j} \subseteq \xi_{C_i}$ car : $P(C_i|v_j) = 1$ et $P(C_l|v_j) = 0,\ l \neq i$.
2. Si $F(\xi_{v_j}, \xi_{C_1}) = F(\xi_{v_j}, \xi_{C_2})$, c'est-à-dire $P(C_1|v_j) = P(C_2|v_j)$, on a :

$$G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}\}) = 1$$

   C'est la valeur maximale.

La $\mathcal{H}$-fonction associée est définie comme suit :

$$\begin{aligned} H(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}\}) &= \max_{j=1..m}(1 - G(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}\})) \\ &= \max_{j=1..m}|P(C_1|v_j) - P(C_2|v_j)| \\ &= \mathcal{K}(\{C_1, C_2\}|A) \end{aligned}$$

Dans le cas général où il y a plus de deux classes, la mesure peut être définie comme suit [152] :

Soit $(\hat{C}_1, \hat{C}_2, .., \hat{C}_n)$ une permutation des classes de $\mathcal{C}$ telle que :

$$P^*(\hat{C}_1|v_j) \leq P^*(\hat{C}_2|v_j) \leq .. \leq P^*(\hat{C}_n|v_j)$$

Supposons que i_0 $(1 \leq i_0 \leq n-1)$ soit l'indice minimal tel que :

$$P^*(\hat{C}_{i_0+1}|v_j) - P^*(\hat{C}_{i_0}|v_j) = \max_{i=1..n-1}|P^*(\hat{C}_{i+1}|v_j) - P^*(\hat{C}_i|v_j)|$$

On regroupe les classes $\hat{C}_1, \hat{C}_2, .., \hat{C}_{i_0}$ et les classes $\hat{C}_{i_0+1}, \hat{C}_{i_0+2}, .., \hat{C}_n$ pour obtenir deux super-classes SC_1 et SC_2.

La distance de Kolmogorov-Smirnov pour un test $A = v_j$ est définie à travers ces deux super-classes SC_1 et SC_2. Ainsi, les deux partitions à considérer sont la partition selon les valeurs d'un attribut et la partition selon ces deux super-classes. D'après la démonstration ci-dessus, cette mesure est validée dans le cadre du modèle $\mathcal{FGH}$.

1.3.4 Discussion

1.3.4.1 Généralisation

La différence se trouve essentiellement dans le niveau $\mathcal{G}$, d'où viennent les formules différentes des entropies. Selon les propriétés du niveau $\mathcal{F}$, $F(C_i|v_j)$ est une fonction décroissante de $P(C_i|v_j)$. Pour les entropies ci-dessus, le niveau $\mathcal{F}$ est défini comme une variante de $-\log P(C_i|v_j)$. Cependant, d'autres fonctions peuvent être utilisées dans le niveau $\mathcal{F}$ pour la construction de mesures de discrimination. On donne certains exemples simples :

$$F(C_i|v_j) = 1 - P(C_i|v_j)$$

(c'est la fonction utilisée pour le niveau $\mathcal{F}$ de la mesure de Kolmogorov-Smirnov), ou :

$$F(C_i|v_j) = \frac{1}{P(C_i|v_j)}$$

Après avoir défini une fonction du niveau $\mathcal{F}$ selon $P(C_i|v_j)$, on peut trouver la fonction inverse qui permet de calculer $P(C_i|v_j)$ à partir de $F(C_i|v_j)$. Cela nous permet de convertir une fonction au niveau $\mathcal{G}$ représentée sous la forme d'une fonction des probabilités en une fonction représentée comme une agrégation de fonctions du niveau $\mathcal{F}$. Ainsi la validation des fonctions du niveau $\mathcal{G}$ est facilitée dans le cas où celles-ci sont des fonctions des probabilités comme c'est le cas pour les entropies de Rényi, de Daróczy et les R-norme entropies.

En dehors des fonctions considérées ci-dessus, on peut utiliser d'autres fonctions. Par exemple, les fonctions suivantes satisfont les propriétés du niveau $\mathcal{G}$:

$$\begin{aligned} G(\mathcal{C}|v_j) &= 1 - \max_{i=1..n} P(C_i|v_j) + \min_{i=1..n} P(C_i|v_j) \\ G(\mathcal{C}|v_j) &= \frac{\min_{i=1..n} P(C_i|v_j)}{\max_{i=1..n} P(C_i|v_j)} \end{aligned}$$

Comme il n'y a pas de contraintes pour l'opération d'agrégation au niveau $\mathcal{H}$, rien ne nous empêche d'utiliser d'autres formules d'agrégation que celles de Type 1, Type 2, Type 3 et Type 4 (cf. formules (1.10)-(1.13)). Les opérateurs d'agrégation peuvent être appliqués à toutes les $\mathcal{G}$-fonctions. Ainsi l'utilisation des entropies conditionnelles de Rényi de Type i, des entropies conditionnelles de Daróczy de Type i, des R-norme entropies, des entropies décentrées et asymétriques de Type i et d'autres mesures dans l'induction d'arbres de décision sont justifiées par le modèle hiérarchique. L'aspect constructif du modèle est illustré puisqu'il justifie la composition des formules conditionnelles et les entropies existantes.

1.3.4.2 Relation entre le modèle $\mathcal{FGH}$ et l'approche axiomatique

Comme nous l'avons présenté au début de la section, dans le cadre de l'approche axiomatique, on établit les propriétés désirées pour une fonction mesurant la quantité

d'information puis on cherche des fonctions qui vérifient ces propriétés. Nous ne considérons ici que les définitions des entropies basées sur la notion de probabilité : on considère a priori que la quantité d'information d'un événement est une fonction de sa probabilité. Les propriétés désirées sont donc toujours imposées sur des probabilités.

Niveau $\mathcal{F}$: On essaie souvent de définir d'abord l'entropie d'un événement comme une fonction de sa probabilité. Plus il est rare (plus sa probabilité est faible), plus la quantité d'information qu'il apporte est grande. L'entropie d'un événement est uniquement déterminée par sa probabilité. C'est pourquoi, les propriétés suivantes sont désirées pour la fonction d'entropie $I(P(e))$ d'un événement e :

1. décroissante par rapport à la probabilité $P(e)$
2. continue selon $P(e)$
3. conjonctive : $I(P(e_1e_2)) = I(P(e_1))+I(P(e_2))$ si e_1 et e_2 sont deux événements indépendants.

À partir de ces propriétés, on ne peut trouver qu'une seule famille de fonctions qui convienne [141] :

$$I(P(e)) = -k \log P(e)$$

où k est une constante. C'est la formule de l'entropie de Shannon pour un événement. Cette définition correspond au niveau $\mathcal{F}$ du modèle $\mathcal{FGH}$.

Cependant, les contraintes imposées aux fonctions du niveau $\mathcal{F}$ sont plus faibles que les contraintes ci-dessus car on ne demande pas la continuité et la propriété de conjonction.

Niveau $\mathcal{G}$: L'entropie d'un ensemble d'événements peut être définie par un système d'axiomes. Dans [5], on trouve les systèmes d'axiomes pour les entropies de Shannon, Rényi, Daróczy. Il peut exister certains systèmes d'axiomes qui impliquent une même formule d'entropie. Voici un exemple d'un système d'axiomes pour l'entropie de Shannon :

1. I est une fonction continue par rapport à P_i.
2. Si toutes les P_i sont égales, I est strictement croissante par rapport à n, le nombre d'événements.
3. $$\begin{aligned} I(P_1, ., P_k, P_{k+1}, .., P_n) &= I(P_1, ., P_k + P_{k+1}, .., P_n) \\ &+ (P_k + P_{k+1}) I\left(\frac{P_k}{P_k + P_{k+1}}, \frac{P_{k+1}}{P_k + P_{k+1}}\right) \end{aligned}$$

Il a été montré que seule l'entropie de Shannon satisfait ces axiomes (théorème de Shannon [141]).

L'entropie d'un ensemble d'événements correspond au niveau $\mathcal{G}$ du modèle $\mathcal{FGH}$.

Niveau $\mathcal{H}$: Le niveau $\mathcal{H}$ du modèle $\mathcal{FGH}$ correspond à une entropie conditionnelle. Soit X, Y deux distributions de probabilité. Certaines propriétés souhaitées pour les entropies conditionnelles sont :

1. $I(XY) = I(X) + I(Y|X)$
2. Si X, Y sont indépendants
   (a) $I(XY) = I(X) + I(Y)$
   (b) $I(X|Y) = I(X)$
3. $I(X) + I(Y|X) = I(Y) + I(X|Y)$
4. $I(X|Y) \leq I(X)$
5. $I(XY) \leq I(X) + I(Y)$

Cependant, il n'y a aucune contrainte pour le niveau $\mathcal{H}$ du modèle $\mathcal{FGH}$.

1.4 Modèle hiérarchique proposé pour les mesures de discrimination floues

1.4.1 Nécessité d'un modèle pour les mesures de discrimination floues

Malgré ses performances, le modèle hiérarchique classique n'est pas assez général pour valider l'utilisation des mesures de discrimination floues qui sont utilisées dans la construction des arbres de décision flous. Dans plusieurs cas, les données disponibles sont imprécises et/ou incertaines et nous devons les manipuler telles quelles. Dans d'autres cas, quand les données disponibles sont précises et certaines, l'utilisation d'un arbre de décision flou permet de traiter les données de manière plus efficace et plus flexible [79] (voir aussi la section 2.5, page 75). En prétraitement, avant chaque choix du meilleur attribut, un attribut numérique peut être discrétisé en le segmentant par des coupures floues. Ces coupures délimitent les frontières entre des sous-ensembles flous de valeurs. Dans ce cas, chaque valeur numérique appartient à un sous-ensemble avec un certain degré d'appartenance. Par ailleurs, chaque exemple peut appartenir partiellement à une classe. Les mesures de discrimination floues sont alors nécessaires pour choisir l'attribut qui discrimine au mieux les exemples relativement à leurs classes.

Bien que l'utilisation des mesures de discriminations floues soit nécessaire et assez répandue [130, 162], il n'existe aucune méthode pour les étudier. Le modèle hiérarchique classique ne permet d'étudier que des mesures de discrimination classique. Néanmoins, dans sa thèse, Marsala [102] a suggéré l'extension de son modèle vers la théorie des sous-ensemble flous. Dans cette section, nous proposons une telle extension du modèle hiérarchique pour l'adapter à des mesures de discrimination floues [47, 44]. Il s'agit aussi d'un modèle hiérarchique, intitulé $\mathcal{FGH}^*$, qui permet de caractériser des mesures de discrimination floues. Il se décompose également en trois niveaux : $\mathcal{F}^*, \mathcal{G}^*, \mathcal{H}^*$. À chaque niveau, un ensemble de propriétés est imposé à des fonctions dont les variables sont des sous-ensembles flous ou des partitions floues. L'avantage de ce modèle est qu'il généralise le modèle classique et qu'il est utilisable non seulement dans la construction des arbres de décision classiques mais aussi dans

la construction des arbres de décision flous. Les mesures usuelles pour construire des arbres flous seront validées par ce modèle, entre autres l'entropie floue [130] et la mesure d'ambiguïté de classification de Yuan et Shaw [162]. De plus, nous proposons une extension vers les événements flous des mesures d'entropie existantes, telles que l'entropie de Rényi, l'entropie de Daróczy, la R-norme entropie. Ces extensions sont ensuite validées par ce modèle hiérarchique afin de les utiliser dans la construction des arbres de décision flous en tant que mesures de discrimination floues.

1.4.2 Description du modèle hiérarchique proposé

1.4.2.1 Choix des opérateurs flous

Afin de généraliser le modèle classique, il est essentiel de choisir entre autres les opérateurs adéquats pour manipuler des sous-ensembles flous, notamment l'intersection, ainsi que la définition de relation floue (inclusion) et celle de la partition d'un ensemble flou.

On se place dans un univers $\mathcal{X}$ donné.

Intersection : L'intersection entre deux sous-ensembles flous A et B de $\mathcal{X}$ est définie par une t-norme comme le *minimum* ou la t-norme probabiliste :

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

ou

$$\mu_{A\cap B}(x) = \mu_A(x)\mu_B(x),\ \forall x \in \mathcal{X}$$

Cette dernière t-norme a été choisie pour notre modèle car elle est plus adaptée aux mesures que nous étudions. Ce choix permet notamment de conserver les propriétés (1.16) et (1.17) que l'on verra dans la suite. Elles sont nécessaires pour assurer que la somme des probabilités de tous les événements flous correspondant à une partition floue reste égale à 1.

Inclusion : On souhaite que l'inclusion entre deux sous-ensembles flous vérifie la propriété :

$$A \subseteq B \Leftrightarrow A \cap B = A$$

Avec la mesure d'inclusion que nous avons choisie, cette propriété s'écrit :

$$\forall x \in \mathcal{X}, A \subseteq B \Leftrightarrow \mu_{A\cap B}(x) = \mu_A(x)$$

Avec ces deux définitions, on a équivalence : A est inclus dans B si et seulement si $\forall x \in \mathcal{X} : \mu_A(x) > 0$ implique $\mu_B(x) = 1$. Ce qui signifie : pour que $A \subseteq B$, il faut que $\mu_A(x)$ soit non nul pour les seuls x qui se trouvent dans le noyau de B (figure 1.2). Avec cette l'inclusion, un sous-ensemble flou ne peut pas être inclus dans

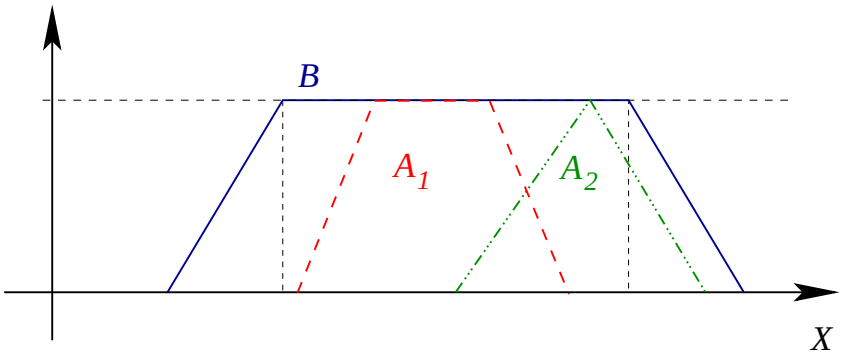

FIG. 1.2 – $A_1 \subset B$ et $A_2 \not\subset B$

deux sous-ensembles flous qui font partie d'une partition floue normale. La même propriété existe dans le cas classique et est utilisée implicitement lors de l'utilisation du niveau $\mathcal{G}$ du modèle hiérarchique classique.

Dans la théorie des sous-ensembles flous, on utilise souvent la définition :

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$$

Si B est un ensemble non flou, cette définition et la précédente sont équivalentes. Dans le contexte de l'apprentissage inductif, B correspond souvent à l'ensemble des exemples d'une classe. Quand chaque exemple appartient à une seule classe, B est un ensemble non flou.

Partition : Il existe différentes définitions des partitions floues [19] d'un ensemble classique. Une partition floue *maximale* d'un ensemble classique A est un ensemble de p sous-ensembles flous $\{A_1, A_2, .., A_p\}$ tel que : $\forall i = 1, .., p : h(A_i) = 1, h(A_i \cap A_k) < 1$ pour $\forall k \neq i$ et $\max_{i=1..p} \mu_{A_i}(x) = 1$ où h désigne la hauteur d'un sous-ensemble flou.

Une partition floue *naturelle* d'un ensemble classique A est un ensemble de p sous-ensembles flous $\{A_1, A_2, .., A_p\}$ tel que : $\forall x \in \mathcal{X}$ il existe un unique sous-ensemble flou I_l telle que : $\max_{i=1..p} \mu_{A_i}(x) = \mu_{A_l}(x)$.

Une partition floue *normale* d'un ensemble classique A est un ensemble de p sous-ensembles flous $\{A_1, A_2, .., A_p\}$ tel que :

$$\forall x \in \mathcal{X} \ : \ \sum_{i=1}^{p} \mu_{A_i}(x) = 1$$

Dans nos travaux, nous avons besoin d'une définition de partition floue pour un ensemble flou. Nous avons choisi une extension de la définition de la partition floue normale donnée dans [16]. C'est une définition de partition floue très utilisée. Une partition floue normale d'un ensemble flou A est un ensemble de p sous-ensembles flous $\{A_1, A_2, .., A_p\}$ tel que :

$$\forall x \in \mathcal{X} \ : \ \sum_{i=1}^{p} \mu_{A_i}(x) = \mu_A(x)$$

Ainsi :

$$\sum_{i=1}^{p} \frac{|A_i|}{|A|} = 1$$

où $|A|$ est la cardinalité floue de A : $|A| = \sum_{x \in X} \mu_A(x)$.

Soit $\{B_1, B_2, .., B_p\}$ une partition floue d'un ensemble flou B. Avec ces définitions, on montre les deux propriétés suivantes :

1. Si $A \subseteq B$ alors $\{A \cap B_1, A \cap B_2, .., A \cap B_p\}$ est une partition floue de A.
   On a : $\mu_{A \cap B_i}(x) = \mu_A(x)\mu_{B_i}(x)$. Alors :

$$\begin{aligned} \sum_{i=1}^{p} \mu_{A \cap B_i}(x) &= \sum_{i=1}^{p} \mu_A(x)\mu_{B_i}(x) = \mu_A(x) \sum_{i=1}^{p} \mu_{B_i}(x) \\ &= \mu_A(x)\mu_B(x) = \mu_{A \cap B}(x) \end{aligned} \tag{1.16}$$

   C'est une propriété plutôt évidente dans le cas classique. Cela donne la propriété suivante qui sera utilisée dans la suite :

$$\sum_{i=1}^{p} \frac{|A \cap B_i|}{|A|} = 1 \tag{1.17}$$

2. Si $A \subseteq B_i$ alors $\forall j \neq i \;\; A \cap B_j = \emptyset$.
   On a : $\forall x \in \mathcal{X}$ et $\mu_A(x) > 0$: $\mu_{B_i}(x) = 1$. Comme $\mu_B(x) = \sum_{i=1}^{p} \mu_{B_i}(x) \leq 1$ alors $\forall j \neq i$: $\mu_{B_j}(x) = 0$ ou bien $A \cap B_j = \emptyset$, $\forall j \neq i$. Dans le cas classique où les B_i sont des ensembles non flous, cette propriété est évidente car les B_i sont disjoints deux à deux.

L'entropie d'un ensemble d'événements flous est une fonction des probabilités floues de chaque événement de l'ensemble. Elle mesure l'incertitude sur les événements. Plus l'entropie de l'ensemble flou est grande, plus la prédiction de l'occurrence d'un événement flou est difficile. C'est une extension de l'entropie classique en remplaçant les probabilités classiques par les probabilités floues.

Soit $\mathcal{X}$ un univers d'événements classiques pour lequel chaque événement x de $\mathcal{X}$ est associé à une probabilité classique $P(x)$. Soit E un sous-ensemble flou de $\mathcal{X}$. E définit un événement flou. La probabilité floue P^* de l'événement flou E est définie par [164] :

$$P^*(E) = \sum_{x \in \mathcal{X}} \mu_E(x) P(x)$$

La probabilité conditionnelle floue de l'événement flou E_1 relativement à l'événement flou E_2 est définie par Smets [143] :

$$P^*(E_1|E_2) = \frac{P^*(E_1 \cap E_2)}{P^*(E_2)}$$

Il est intéressant de remarquer que dans un article de Zadeh [164], la définition de la probabilité conditionnelle floue est :

$$P^*(E_1|E_2) = \frac{P^*(E_1E_2)}{P^*(E_2)}$$

où le produit a été utilisé au lieu de l'intersection dans la formule de Smets. Cependant, la définition du produit dans la formule de Zadeh est le produit. Ce produit est la t-norme que l'on utilise pour l'intersection dans la formule de Smets. Ces deux définitions coïncident donc.

Si $\mathcal{X}$ est un univers contenant des événements équiprobables, la probabilité d'un sous-ensemble flou $S \subseteq \mathcal{X}$ est $P^*(S) = \frac{|S|}{|\mathcal{X}|}$.

1.4.2.2 Description du modèle proposé

Soit $\mathcal{S}^*[\mathcal{X}]$ l'ensemble des sous-ensembles flous de $\mathcal{X}$ et $\mathbb{P}^*[\mathcal{X}]$ l'ensemble des partitions floues de $\mathcal{X}$.

En apprentissage inductif, $\mathcal{X}$ est l'ensemble des exemples ξ. À la différence de la section précédente, l'ensemble des exemples d'une classe peut être flou. Ainsi, un exemple peut éventuellement appartenir à plusieurs classes. Dans la suite, nous faisons l'hypothèse que la somme des degrés d'appartenance d'un exemple à toutes les classes est 1. Si ce n'est pas le cas, les degrés doivent être d'abord normalisés.

Au lieu de considérer directement les valeurs d'un attribut, dans cette partie nous considérons plutôt les modalités associées $v_1, v_2, .., v_m$ à un attribut. Chaque valeur d'attribut appartient à une ou plusieurs modalités.

Nous supposons aussi que la somme des degrés d'appartenance d'une valeur à toutes les modalités possibles est 1. Cette fois, chaque attribut est évalué par son pouvoir discriminant vis-à-vis des classes, selon les valeurs qu'il prend.

On note ξ_{C_i} le sous-ensemble flou des exemples appartenant à la classe C_i avec un degré strictement positif, et ξ_{v_j} le sous-ensemble flou de tous les exemples dont la valeur pour l'attribut A appartient à la valeur v_j avec un degré strictement positif :

$$\xi_{v_j} = \sum_{i=1}^{N} \mu_{v_j}(e_i(A))/e_i$$

où $e_i(A)$ est la valeur de l'attribut A pour l'exemple e_i de la base ξ et $\sum$ est la notation usuelle pour représenter un sous-ensemble flou et ne désigne pas l'opérateur somme. $\{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}$ et $\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}$ sont donc des partitions floues de ξ.

Nous proposons de définir les 3 niveaux du modèle $\mathcal{FGH}^*$ comme suit.

Définition 1.4.1. ($\mathcal{F}^*$-fonction). *Une $\mathcal{F}^*$-fonction est une fonction F^* :*

$$F^* : \mathcal{S}^*[\mathcal{X}] \times \mathcal{S}^*[\mathcal{X}] \to \mathbb{R}^+$$

telle que :

1. *$F^*(U, V)$ est minimum quand $U \subseteq V$.*

2. *$F^*(U,V)$ est maximum quand $U \cap V = \emptyset$.*
3. *$F^*(U,V)$ est strictement décroissante avec $U \cap V$ c'est-à-dire si $U \cap V_1 \subset U \cap V_2$ alors $F^*(U,V_1) > F^*(U,V_2)$.*

Une $\mathcal{F}^*$-fonction mesure l'inadéquation entre deux sous-ensembles flous. En apprentissage inductif $F^*(\xi_{v_j}, \xi_{C_i})$ mesurera donc l'inadéquation de ξ_{v_j} par rapport à ξ_{C_i}.

Comme remarqué par Marsala, l'auteur du modèle $\mathcal{FGH}$, les fonctions du niveau $\mathcal{F}$ sont fondées sur une mesure de dissimilarité des sous-ensembles induits par les modalités d'un attribut envers les sous-ensembles induits par les classes. Nous ne pensons donc pas que c'est la seule extension possible des fonctions du niveau $\mathcal{F}$. Par exemple, au lieu d'utiliser la relation d'inclusion et de conjonction entre deux sous-ensembles floues, on peut envisager d'utiliser une mesure de similarité [133, 134] ou une mesure de satisfiabilité.

Soit une $\mathcal{F}^*$-fonction F^* et une suite de fonctions réelles continues $g_k^* : \mathbb{R}^k \longrightarrow \mathbb{R}^+$, $k \in \mathbb{N}$.

Définition 1.4.2. ($\mathcal{G}^*$-fonction). *Une $\mathcal{G}^*$-fonction est une fonction G^* :*

$$G^* : \mathcal{S}^*[\mathcal{X}] \times \mathbb{P}^*[\mathcal{X}] \to \mathbb{R}^+$$

telle que : $G^*(U,\mathcal{V}) = g_n^*(F^*(U,V_1), F^*(U,V_2), .., F^*(U,V_n))$ où $\mathcal{V} = \{V_1, V_2, .., V_n\}$ est une partition floue de $\mathcal{X}$ et :

1. *$G^*(U,\mathcal{V})$ est minimum quand il existe V_i $(1 \leq i \leq n)$ tel que $U \subseteq V_j$.*
2. *$G^*(U,\mathcal{V})$ est maximum quand $F^*(U,V_1) = F^*(U,V_2) = ... = F^*(U,V_n)$.*

Si on note $\xi_{\mathcal{C}}$ la partition floue de l'ensemble des exemples par leurs classes, $G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$ mesure donc le pouvoir discriminant de la valeur floue v_j relativement à $\mathcal{C}$.

On montre que les différentes formules d'entropie floues sont des $\mathcal{G}^*$-fonctions.

Soit une $\mathcal{G}^*$-fonction G^* et une suite de fonctions réelles continues $h_k^* : \mathbb{R}^{+k} \longrightarrow \mathbb{R}^+$, $k \in \mathbb{N}$.

Définition 1.4.3. ($\mathcal{H}^*$-fonction). *Une $\mathcal{H}^*$-fonction est une fonction H^* :*

$$H^* : \mathbb{P}^*[\mathcal{X}] \times \mathbb{P}^*[\mathcal{X}] \to \mathbb{R}^+$$

telle que :

$$H^*(\mathcal{U},\mathcal{V}) = h_m^*(G^*(U_1,\mathcal{V}), G^*(U_2,\mathcal{V}), .., G^*(U_m,\mathcal{V}))$$

où $\mathcal{V} = \{V_1, V_2, .., V_m\}$ et $\mathcal{U}$ sont des partitions floues de $\mathcal{X}$.

Si ξ_A est une partition floue de l'ensemble des exemples par les valeurs de l'attribut A, $H^*(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$ mesure donc le pouvoir discriminant d'un attribut A relativement à la classe.

Dans notre contexte, il y a une équivalence entre événements flous et sous-ensembles flous. Le modèle proposé modélise donc des rapports entre des événements flous ou entre des ensembles des événements flous. Une $\mathcal{F}^*$-fonction peut mesurer l'inadéquation entre deux événements flous. Une $\mathcal{G}^*$-fonction mesure le pouvoir discriminant d'un événement flou relativement à un ensemble d'événements flous. Et finalement, une $\mathcal{H}^*$-fonction évalue le pouvoir discriminant d'un ensemble d'événements flous relativement à un autre ou bien l'adéquation entre ces deux ensembles d'événements flous.

1.5 Validation des mesures de discrimination floues

Dans la suite, nous généralisons d'abord des mesures d'entropie avec des concepts de la théorie des sous-ensembles flous. Nous validons ensuite les mesures par le modèle $\mathcal{FGH}^*$. Une mesure M sera validée s'il est possible de mettre en évidence une $\mathcal{F}^*$-fonction, une $\mathcal{G}^*$-fonction et une $\mathcal{H}^*$-fonction pour M.

1.5.1 Introduction des mesures d'entropie floues

Dans cette section, les définitions des différentes mesures d'entropie sont étendues. On considèrera l'entropie des événements flous, une extension de l'entropie de Rényi [6, 132], une extension de l'entropie de Daróczy [49], une extension de la R-norme entropie et la mesure d'ambiguïté de Yuan et Shaw. Les définitions de ces entropies sont issues de la théorie de l'information et elles sont étendues aux événements flous.

Notons $P^*(C_i)$ la probabilité de la classe floue C_i dans la base d'exemples ξ ; $P^*(v_j)$ la probabilité qu'un exemple de la base ξ prenne la modalité floue v_j pour valeur d'un attribut A ; $P^*(C_i|v_j)$ la probabilité qu'un exemple prenne la modalité floue v_j pour valeur d'un attribut A appartient à la classe C_i :

$$P^*(C_i) = \frac{|\xi_{C_i}|}{|\xi|} \quad ; \quad P^*(v_j) = \frac{|\xi_{v_j}|}{|\xi|} \quad \text{et} \quad P^*(C_i|v_j) = \frac{|\xi_{C_i} \cap \xi_{v_j}|}{|\xi_{v_j}|}$$

On a donc :

$$\sum_{i=1}^{n} P^*(C_i) = 1 \quad \text{et} \quad \sum_{j=1}^{m} P^*(v_j) = 1$$

À partir de (1.17), on a aussi :

$$\sum_{i=1}^{n} P^*(C_i|v_j) = 1$$

L'entropie d'un événement flou [147] est définie par :

$$I_S^*(P_1^*, P_2^*, .., P_n^*) = -\sum_{i=1}^{n} P_i^* \log P_i^* \tag{1.18}$$

C'est une extension de l'entropie de Shannon aux événements flous. Certains auteurs [130] la nomment l'entropie étoile. Dans ce qui suit, le terme « entropie floue » sans plus de précision, désigne cette entropie.

De façon similaire à l'extension de l'entropie de Shannon, nous proposons de remplacer les probabilités classiques par les probabilités floues dans les formules correspondantes pour obtenir les mesures suivantes :

L'entropie floue de Daróczy s'écrit alors :

$$I_D^{*\beta}(P_1^*, P_2^*, .., P_n^*) = \frac{2^{\beta-1}}{2^{\beta-1}-1}\left(1 - \sum_{i=1}^{n} P_i^{*\beta}\right)$$

L'entropie floue de Rényi :

$$I_R^{*\beta}(P_1^*, P_2^*, .., P_n^*) = \frac{1}{1-\beta}\log\sum_{i=1}^{n} P_i^{*\beta}$$

La R-norme entropie floue :

$$I_{R-norme}^{*\beta}(P_1^*, P_2^*, .., P_n^*) = \frac{\beta}{\beta-1}\left(1 - \left(\sum_{i=1}^{n} P_i^{*\beta}\right)^{\frac{1}{\beta}}\right)$$

Dans ces formules d'entropie $\beta > 0$ et $\beta \neq 1$.

Remarque De manière semblable, les mesures d'entropie décentrée et asymétrique floues peuvent être construites et étudiées.

1.5.2 Entropie d'événements flous

Pour l'entropie d'événements flous, la $\mathcal{F}^*$-fonction associée est la suivante :

$$F^*(\xi_{v_j}, \xi_{C_i}) = -\log P^*(C_i|v_j)$$

F^* est une $\mathcal{F}^*$-fonction car :

1. Si $\xi_{v_j} \subseteq \xi_{C_i}$ alors $P^*(C_i|v_j) = 1$ donc $\log P^*(C_i|v_j) = 0$, qui est la valeur minimale.
2. Si $\xi_{v_j} \cap \xi_{C_i} = \emptyset$ alors $P^*(C_i|v_j) = 0$ donc $-\log P^*(C_i|v_j) = +\infty$, qui est la valeur maximale.
3. $F^*(\xi_{v_j}, \xi_{C_i})$ est une fonction strictement décroissante de $|\xi_{C_i} \cap \xi_{v_j}|$ quand $|\xi_{v_j}|$ est constante.

Le niveau $\mathcal{G}^*$ associé à l'entropie floue est la fonction suivante obtenue par agrégation des F^*-fonctions associées :

$$G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{i=1}^{n} 2^{-F^*(\xi_{v_j}, \xi_{C_i})} F^*(\xi_{v_j}, \xi_{C_i})$$

qui peut s'écrire :

$$G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = -\sum_{i=1}^{n} P^*(C_i|v_j) \log P^*(C_i|v_j)$$

G^* vérifie bien les propriétés requises :

1. $G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = 0$, c'est la valeur minimale quand il existe C_i $(1 \leq i \leq n)$ telle que $\xi_{v_j} \subseteq \xi_{C_i}$ car : $P^*(C_i|v_j) = 1$ et $P^*(C_l|v_j) = 0$, $\forall l \neq i$ (sachant que $\{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}$ $(1 \leq i \leq n)$ est une partition floue de ξ donc $\sum_{i=1}^{n} P^*(C_i|v_j) = 1$).
2. Si $F^*(\xi_{v_j}, \xi_{C_1}) = F^*(\xi_{v_j}, \xi_{C_2}) = .. = F^*(\xi_{v_j}, \xi_{C_n})$, c'est-à-dire $P^*(C_i|v_j) = P^*(C_l|v_j)$ $\forall i, l : 1 \leq i \neq l \leq n$, on a $G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \log n$ et c'est la valeur maximale.

Le niveau $\mathcal{H}^*$ nous permet de retrouver la formule d'entropie conditionnelle floue en agrégeant les G^*-fonctions associées :

$$H^*(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} P^*(v_j) G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$$

Évidemment, on peut utiliser d'autres formules d'agrégation, entre autres les formules conditionnelles de Type 2, de Type 3 et de Type 4.

1.5.3 Entropie floue de Daróczy, entropie floue de Rényi et R-norme entropie floue

Les validations de ces mesures par le modèle $\mathcal{FGH}^*$ sont similaires à celles des entropies classiques par le modèle $\mathcal{FGH}$ (cf. section 1.3 et [45]).

Les $\mathcal{F}^*$-fonctions associées à l'entropie floue de Daróczy, à l'entropie floue de Rényi et à la R-norme entropie floue sont identiques :

$$F^*(\xi_{v_j}, \xi_{C_i}) = -\beta \log P^*(C_i|v_j)$$

Les $\mathcal{G}^*$-fonctions pour ces entropies sont obtenues en remplaçant P_j^* par $2^{-F^*(\xi_{v_j}, \xi_{C_i})}$ dans les formules de l'entropie floue de Daróczy, de l'entropie floue de Rényi et de la R-norme entropie floue.

La $\mathcal{G}^*$-fonction associée à l'entropie de Daróczy floue s'écrit donc :

$$G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \frac{2^{\beta-1}}{2^{\beta-1} - 1} \left(1 - \sum_{i=1}^{n} 2^{-F^*(\xi_{v_j}, \xi_{C_i})}\right)$$

La $\mathcal{G}^*$-fonction associée à l'entropie de Rényi floue :

$$G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \frac{1}{1 - \beta} \log \sum_{i=1}^{n} 2^{-F^*(\xi_{v_j}, \xi_{C_i})}$$

La $\mathcal{G}^*$-fonction associées à la R-norme entropie floue :

$$G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \frac{\beta}{\beta - 1}\left(1 - \left(\sum_{i=1}^{n} 2^{-F^*(\xi_{v_j}, \xi_{C_i})}\right)^{\frac{1}{\beta}}\right)$$

Les $\mathcal{H}^*$-fonctions sont une agrégation de $\mathcal{G}^*$-fonctions, par exemple, une somme pondérée des $\mathcal{G}^*$-fonctions :

$$H^*(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} w_j G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$$

En particulier, on retrouve les formules des entropies conditionnelles de Type 1, 2, 3 et Type 4 (formules (1.10)-(1.13)) :

Type 1 :

$$H^*(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} P^*(v_j) G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$$

Type 2 :

$$H^*(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} P^{*\beta}(v_j) G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$$

Type 3 :

$$H^*(\{\xi_{v_1}, .., \xi_{v_m}\}, \{\xi_{C_1}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} \frac{P^{*\beta}(v_j)}{\sum_{k=1}^{m} P^{*\beta}(v_k)} G^*(\xi_{v_j}, \{\xi_{C_1}, .., \xi_{C_n}\})$$

Type 4 :

$$H^*(\{\xi_{v_1}, .., \xi_{v_m}\}, \{\xi_{C_1}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} \frac{P^*(v_j)}{-\sum_{k=1}^{m} P^*(v_k) \log P^*(v_k)} G^*(\xi_{v_j}, \{\xi_{C_1}, .., \xi_{C_n}\})$$

1.5.4 Mesure de Yuan et Shaw

Cette mesure d'ambiguïté de classification est introduite dans [162] pour la construction des arbres de décision flous. Nous démontrons ici sa validité dans le modèle hiérarchique proposé.

Soit $(P^*_{1'}, P^*_{2'}, .., P^*_{n'})$ une permutation de $(P^*_1, P^*_2, .., P^*_n)$ telle que $P^*_{1'} \geq P^*_{2'} \geq .. \geq P^*_{n'}$. Soit C_+ la classe majoritaire dont la probabilité est $P_+ = P^*_{1'}$.

Soit : $\pi(P^*_i) = \frac{P^*_i}{P_+}$ et $\pi^*_i = \pi(P^*_{i'}) = \frac{P^*_{i'}}{P_+}$. On a : $1 = \pi^*_1 \geq \pi^*_2 \geq .. \geq \pi^*_n$ et :

$$I_Y(P^*_1, P^*_2, .., P^*_n) = \sum_{i=2}^{n} \pi^*_i (\log i - \log(i-1))$$

Nous démontrons dans ce qui suit que I_Y correspond à une $\mathcal{G}^*$-fonction et on construit une $\mathcal{F}^*$-fonction et une $\mathcal{H}^*$-fonction qui lui correspondent.

$\mathcal{F}^*$-fonction associée :

$$F^*(\xi_{v_j}, \xi_{C_i}) = -\log \pi(P^*(C_i|v_j))$$

1. Si $\xi_{v_j} \subseteq \xi_{C_i}$ alors $P^*(C_i|v_j) = 1$ et $P^*(C_l|v_j) = 0,\ \forall l : 1 \leq l \neq i \leq n$. Ainsi $\pi(P^*(C_i|v_j)) = 1$ et

$$F^*(\xi_{v_j}, \xi_{C_i}) = -\log \pi(P^*(C_i|v_j)) = 0$$

2. Si $\xi_{v_j} \cap \xi_{C_i} = \emptyset$ alors $P^*(C_i|v_j) = 0$. Donc $\pi(C_i|v_j) = 0$ et

$$F^*(\xi_{v_j}, \xi_{C_i}) = -\log \pi(P^*(C_i|v_j)) = +\infty$$

3. $F^*(\xi_{v_j}, \xi_{C_i})$ est une fonction strictement décroissante de $|\xi_{C_i} \cap \xi_{v_j}|$ car

$$\pi(P^*(C_i|v_j)) = \frac{P^*(C_i|v_j)}{P^*(C_+|v_j)}$$

est une fonction strictement décroissante de $|\xi_{C_i} \cap \xi_{v_j}|$.

$\mathcal{G}^*$-fonction associée :

$$\begin{aligned} G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) &= \sum_{i=2}^{n} \pi_i^* (\log i - \log(i-1)) \\ &= \sum_{i=2}^{n} 2^{-F^*(\xi_{v_j}, \xi_{C_{i'}})} (\log i - \log(i-1)) \end{aligned}$$

C'est une agrégation de $\mathcal{F}^*$-fonctions.

1. S'il existe C_i $(1 \leq i \leq n)$ telle que $\xi_{v_j} \subseteq \xi_{C_i}$, on a : $\pi(C_i|v_j) = 1$ et $\pi(C_l|v_j) = 0,\ \forall l : 1 \leq i \neq l \leq n$ alors $\pi_1^* = 1$ et $\pi_i^* = 0,\ 1 < i \leq n$. Donc $G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = 0$, ce qui est la valeur minimale.
2. Si $F^*(\xi_{v_j}, \xi_{C_1}) = F^*(\xi_{v_j}, \xi_{C_2}) = .. = F^*(\xi_{v_j}, \xi_{C_n})$, c'est-à-dire $\pi_i^* = \pi_l^*,\ \forall i, l : 1 \leq i, l \leq n$. Comme $\pi_1^* = 1$, ainsi $\forall i : 1 \leq i \leq n : \pi_i^* = 1$. Donc :

$$G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{i=2}^{n} \pi_i^* (\log i - \log(i-1)) = \log n$$

c'est la valeur maximale de G^* car $\pi_i^* = 1,\ \forall i$ ($\pi_i^* \leq 1$ par définition).

$\mathcal{H}^*$-fonction associée :

$$H^*(\{\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) = \sum_{j=1}^{m} P^*(v_j) G^*(\xi_{v_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$$

1.6 Conclusion

Dans ce chapitre, nous avons proposé une taxonomie des mesures de discrimination en distinguant les mesures de discrimination entre deux individus par rapport aux mesures de discrimination d'un discriminateur. Ensuite, grâce au modèle hiérarchique classique existant dans l'état de l'art, nous avons justifié l'utilisation de plusieurs mesures dans la construction des arbres de décision. En plus des mesures usuelles, certaines mesures telles que l'entropie généralisée de Rényi, de Daróczy et la R-norme entropie ont été étudiées et introduites dans un tel processus.

En généralisant le modèle classique, nous avons proposé un nouveau modèle pour les mesures de discrimination floues. Il permet de vérifier si une mesure est pertinente pour la construction des arbres de décision flous. La généralisation se fait essentiellement par le choix d'opérateurs adéquats. C'est une structure hiérarchique constructive, consistant en des définitions des fonctions ayant des propriétés spécifiques pour la construction des arbres de décision. La structure des fonctions ainsi que leurs propriétés permettent de justifier l'utilisation de ces mesures de discrimination. Le modèle aide à donner une justification sémantique aux mesures de discrimination intervenant dans l'induction avec des facteurs flous. À côté des mesures connues dans l'état de l'art telles que l'entropie d'événements flous, la mesure de Yuan et Shaw, un certain nombre de mesures sont proposées et validées théoriquement par le modèle proposé. Entre autres, ce sont des mesures généralisées de l'entropie de Rényi et l'entropie de Daróczy. La validation est faite par la mise en évidence des 3 fonctions correspondant à chacun des niveaux du modèle hiérarchique, chacune de ces fonctions devant satisfaire un certain nombre de propriétés exigées.

La validation empirique de ces mesures est présentée dans la deuxième et la troisième partie de la thèse.

Deuxième partie

UTILISATION DE MESURES DE DISCRIMINATION EN APPRENTISSAGE INDUCTIF

Chapitre 2

Apprentissage par arbres de décision

2.1 Introduction

Les arbres de décision comptent parmi les techniques d'apprentissage inductif supervisé les plus utilisées. Le but est de déduire automatiquement des règles, représentées en une structure arborescente, qui décrivent les relations entre les valeurs d'attributs et les classes auxquelles appartiennent des exemples issus d'une base de données. Ainsi les règles caractérisent des ensembles de données de la base. La structure arborescente utilisée pour représenter ces règles correspond à un tri des attributs selon leur influence sur l'attribut de classe. Les connaissances obtenues peuvent être utilisées par la suite pour résumer et généraliser des données ainsi que pour classer de nouveaux exemples et pour prédire leur classe. La description et la généralisation des données par des règles conduisent à une diminution importante de leur volume et à une meilleure compréhension. En particulier, la dépendance entre des attributs et la classe, qui est un attribut particulier, peut être relevée. Dans cette thèse, nous nous intéressons à l'utilisation des arbres de décision pour la classification. En classant des exemples, les règles découvertes caractérisent également les classes et interprètent la signification de chacune des classes.

Un problème de classification se décompose en deux phases, dont la première consiste à construire un modèle de classification à partir d'une base de données. Pendant la deuxième phase, ce modèle est employé pour classer de nouveaux exemples appartenant à une base de test. L'arbre de décision est une technique de classification à plusieurs étages. La classification de nouveaux exemples est réalisée par une suite de questions dont chacune porte sur les valeurs des attributs. Dans la plupart des cas, elle porte sur la valeur d'un seul attribut. Les questions et réponses sur le chemin de la racine à une feuille déterminent une conjonction d'attributs. Cette conjonction définit un ensemble d'exemples qui possèdent des valeurs communes sur ces attributs. Les questions posées permettent de décomposer un problème complexe en un certain nombre de problèmes plus simples suivant la stratégie *diviser pour régner*. Les réponses obtenues permettent d'identifier des sous-problèmes (qui

correspondent à des sous-espaces d'entrées). Le processus se termine quand on arrive à un problème simple qui a une réponse bien identifiée. Grâce à la suite des réponses disponibles, le résultat de la classification est facile à interpréter.

Les arbres de décision possèdent les avantages suivants :

1. Le traitement de manière homogène de presque tous les types d'attributs : numérique, symbolique, flou ou probabiliste. Les méthodes existantes de construction d'arbres de décision s'adaptent à plusieurs situations : la présence de données manquantes ou erronées, la construction incrémentale des arbres [151], etc. Cela permet de trouver des applications dans plusieurs domaines de recherche car il y a peu de contraintes sur les données (*non-parametric method*). Il existe également des implémentations parallèles pour assurer un traitement adéquat des grandes bases de données.
2. La complexité des arbres de décision ne dépend pas du nombre de classes.
3. Les règles obtenues sont relativement faciles à comprendre et à interpréter. Cela facilite la communication avec les experts du domaine d'application et favorise le choix des arbres de décision en analyse des données.
4. Les arbres de décision ordonnent les attributs selon leur influence sur la classe des exemples. On peut donc savoir quels sont les attributs décisifs ainsi que leur importance relativement à la classification.
5. La phase de classification basée sur un arbre de décision n'est pas coûteuse. Il n'est pas nécessaire de connaître les valeurs de tous les attributs. Seules les valeurs des attributs qui apparaissent dans le chemin correspondant sont exigées et il est garanti que ces valeurs seront utiles pour la classification. On peut éventuellement s'arrêter à n'importe quel niveau et avoir une réponse probabiliste qui correspond à la distribution des exemples associés au nœud en question.

Les inconvénients des arbres de décision sont les suivants :

1. La technique d'arbre est moins appropriée pour la prédiction de classe à valeurs continues (arbre de régression).
2. Le coût de calcul pour la construction des arbres de décision est relativement élevé. À chaque nœud, il faut évaluer tous les attributs pour choisir le plus adéquat. Quand on veut combiner des attributs (arbre de décision oblique [111] et arbre multivariables [29]), la recherche des poids optimaux est difficile. Jusqu'à maintenant, l'utilisation des arbres obliques n'est pas très répandue.
3. Les arbres de décision sont moins performants lorsqu'on a relativement peu d'exemples et plusieurs classes.
4. Les arbres de décision découpent l'espace de données en hypercubes rectangulaires. Ils travaillent moins bien si les classes ne sont pas bien séparées par de telles surfaces. En fait, les arbres fournissent une approximation en « escalier » des surfaces de séparation entre classes.
5. Un phénomène de sur-apprentissage peut se produire quand il y a trop de données. C'est plutôt un problème commun en apprentissage et non seulement pour les arbres de décision.

Dans ce chapitre, dans un premier temps, nous présentons le schéma de construction des arbres de décision. Ensuite, la sélection d'attribut et la discrétisation des attributs numériques sont respectivement présentées dans la section 2.3 et la section 2.4. Dans chacune de ces deux sections, nous introduisons des mesures de discrimination généralisées utilisées dans ces processus (section 2.3.2 et section 2.4.2). Ce sont des formules conditionnelles de l'entropie de Rényi et de l'entropie de Daróczy. Nous proposons également dans la section 2.4.3 une mesure d'équilibre pour caractériser la discrétisation d'un attribut numérique. Dans un deuxième temps, dans la section 2.5 nous étudions l'utilisation de la logique floue dans la construction des arbres de décision pour mieux prendre en compte des imprécisions et des incertitudes en proposant une taxonomie des méthodes existantes. Dans cette taxonomie, une nouvelle utilisation de certaines mesures de discrimination floue est proposée. Finalement, nous concluons le chapitre.

Pour un état de l'art sur des arbres de décision, nous suggérons aux lecteurs les travaux de synthèse de Safavian et Landgrebe [139], Murthy [110], Rokach et Maimon [138]. Mitchell [109] et Cornuéjols, Miclet et Kodratoff [40] consacrent dans leurs livres des chapitres entiers aux arbres de décision. Dans [96], une étude comparative de plusieurs algorithmes de classification, y compris des algorithmes de construction d'arbres de décision, a été réalisée.

2.2 Construction d'arbres de décision

Les principaux buts de la construction des arbres de décision sont :

1. une meilleure généralisation des exemples de la base d'apprentissage
2. une meilleure classification de nouveaux exemples
3. une structure aussi simple que possible.

On préfère souvent des arbres de décision simples. Ils sont plus compréhensibles et rendent plus rapide la phase de classification. Selon le principe du rasoir d'Occam, ils ont plus de chances d'avoir de bonnes capacités de généralisation. Breiman *et al.* [27] montrent également que la complexité d'un arbre influe sur sa performance. Entre autres, la complexité des arbres est contrôlée par les critères d'arrêt et le processus d'élagage. La complexité des arbres est, entre autres, évaluée par le nombre de nœuds, le nombre de feuilles, la hauteur des arbres (hauteur moyenne avec ou sans pondération, hauteur minimale, hauteur maximale), le nombre d'attributs utilisés. Il faut noter toutefois qu'un arbre doit être évalué par des critères spécifiques à son usage, notamment la précision, la capacité de discrimination, le coût de classification, etc. Nous en parlerons en détails dans le chapitre 3.

Le nombre d'arbres de décision qui décrivent une base d'exemples est croissant de manière exponentielle selon le nombre d'attributs K et le nombre moyen de valeurs possibles par attributs $\overline{m}$. Selon [40], le nombre d'arbres possibles est :

$$\sum_{i=0}^{K-1}(K-i)^{\overline{m}^i}$$

La construction d'un arbre optimal, au sens où il classe parfaitement la base d'apprentissage et où il minimise le nombre moyen de tests nécessaires pour un exemple inconnu, est un problème NP-complet. Il faut donc chercher à construire un arbre *quasi-meilleur* de manière heuristique.

Dans la suite de la section, après une brève description des stratégies de construction d'arbres de décision, le schéma TDIDT est détaillé. Ensuite, nous expliquons en quelques mots l'utilisation des arbres.

2.2.1 Stratégies de construction d'arbres de décision

Safavian et Landgrebe [139] divisent les méthodes de construction d'arbres de décision en quatre catégories :

1. Bottom-Up [93] : selon cette approche, à chaque étape, grâce à une mesure de distance, qui est calculée sur les exemples de la base d'apprentissage, les deux groupes dont la distance entre eux est la plus petite sont fusionnés pour avoir un nouveau groupe. La fusion continue avec un nouvel ensemble de groupes et se termine lorsqu'on obtient un seul groupe qui est à la racine de l'arbre. L'arbre ainsi construit est un arbre binaire. Plus un partitionnement est proche de la racine, plus les deux groupes sont discriminants. Cette approche a des caractéristiques en commun avec le regroupement non-supervisé.
2. Top-Down : cette approche consiste à construire un arbre depuis sa racine vers ses feuilles en partitionnant successivement la base d'apprentissage. C'est la stratégie la plus utilisée sous le nom « Induction descendante d'arbres de décision » (*Top Down Induction of Decision Tree* (TDIDT)). La figure 2.1 décrit le principe de la stratégie TDIDT. Dans ce chapitre, seule cette stratégie est examinée dans la suite.
3. Hybride : cette approche, proposée par Kim et Landgrebe (voir [83] et aussi [139]), consiste à utiliser un processus bottom-up pour diriger et aider un processus top-down. Le processus bottom-up fournit des informations sur des groupes au processus top-down. En les exploitant, le processus top-down partitionne la base d'apprentissage. Ce partitionnement n'est pas forcément identique au partitionnement par le processus bottom-up. On procède de la même manière avec des sous-bases d'exemples jusqu'à ce que tous les exemples associés au nœud considéré appartiennent à une même classe.
4. *Growing Pruning* [27] : cette approche consiste à développer un arbre jusqu'à la taille maximale (les exemples associés à une feuille appartient à une seule classe) puis élaguer les branches. Cela permet d'éviter certaines difficultés du choix de critère d'arrêt.

2.2.2 Schéma TDIDT

La plupart des algorithmes d'induction d'arbres de décision font partie de cette catégorie. On peut citer entre autres : ID3 (*Interactive Dichotomizer version 3*) [122],

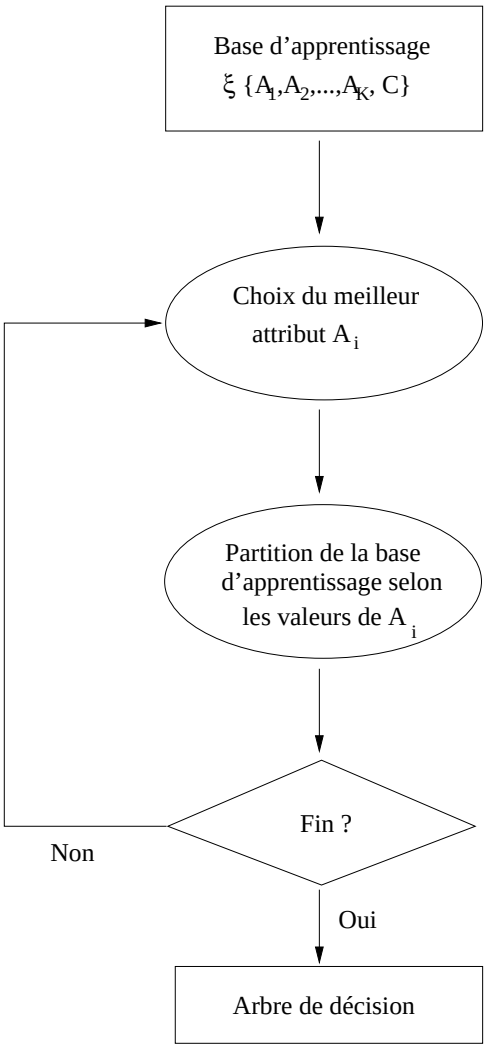

FIG. 2.1 – Construction d'arbres de décision par la stratégie TDIDT

CART [27], C4.5 [124], etc. L'arbre est construit depuis sa racine vers ses feuilles en partitionnant successivement la base d'apprentissage. Chaque nœud est associé à un ensemble d'exemples, en particulier la racine est associée à la base entière. À chaque itération, on cherche à partitionner la base associée à un nœud selon les valeurs d'un attribut choisi. Tous les exemples ayant la même valeur pour l'attribut choisi sont regroupés dans un même nœud fils. Le processus s'arrête lorsque des critères d'arrêt sont vérifiés à toutes les feuilles.

Ce schéma d'induction d'arbres de décision a été initialement étudié pour le cas de données symboliques. Les algorithmes originels éprouvent des difficultés lorsqu'ils sont appliqués à des données numériques ou floues. La première difficulté est que le nombre de valeurs possibles pour un attribut est très grand. Cela conduit à des arbres ayant beaucoup de branches. Les données numériques sont aussi ordonnées et ainsi la proximité entre valeurs doit être prise en compte. Il n'est donc pas adéquat

de les traiter telles qu'elles sont. Aussi, le plus souvent les méthodes sont généralisées pour les données numériques en y insérant une phase de discrétisation qui permet de transformer les données numériques en données symboliques. Plus récemment, cette méthode a été étendue pour des bases décrites par des attributs flous.

Soit un nœud $\mathcal{N}$ contenant un ensemble d'exemples ξ.

1. Condition d'arrêt : Si l'ensemble associé au nœud $\mathcal{N}$ satisfait des critères d'arrêt, alors le nœud est une feuille. Celle-ci est alors étiquetée par la classe majoritaire.
2. Sinon, faire les étapes suivantes :
   (a) i. Discrétiser tous les attributs numériques pour pouvoir les représenter et les traiter de la même manière que des attributs symboliques.
   ii. Choisir un attribut et l'affecter au nœud courant.
   (b) Partitionner les exemples de $\mathcal{N}$ en sous-ensembles $\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}$ selon les valeurs v_1, v_2, .., v_m de l'attribut choisi.
   (c) Créer de nouveaux nœuds pour chaque sous-ensemble ξ_i non vide de la partition. Les nouveaux nœuds sont ajoutés comme les fils du nœud $\mathcal{N}$.
   (d) Appliquer récursivement la procédure sur les nouveaux nœuds.

Suivant ce schéma, plusieurs méthodes de construction d'arbres de décision ont été proposées. Elles se différencient selon les techniques appliquées à chaque étape. Dans la suite de cette partie, nous détaillons les principales variantes.

Le choix d'un attribut pour l'affecter au nœud courant et pour partitionner la base d'apprentissage est le cœur de l'algorithme. En principe, l'attribut est choisi par une heuristique. C'est celui qui est le plus discriminant, c'est-à-dire qui maximise une mesure de discrimination. Cela permet de manière heuristique d'obtenir un arbre performant. Les deux sections suivantes se consacrent entièrement à la sélection du meilleur attribut et à la discrétisation des attributs numériques.

Étant donné l'attribut choisi, le partitionnement de la base d'apprentissage s'appuie sur ses valeurs. Dans le cas d'un attribut symbolique, chaque sous-base est constituée des exemples ayant la même valeur pour l'attribut en question. Dans le cas d'un attribut numérique discrétisé par des coupures précises, chaque sous-base correspond à un intervalle dans son domaine. Ainsi chaque exemple n'appartient qu'à une seule partie. Il existe toutefois des variantes de ce principe. Par exemple, un nœud peut éventuellement contenir des exemples dont la valeur pour un attribut fait partie d'un groupe de quelques valeurs possibles.

Classiquement, à chaque nœud l'algorithme s'arrête quand, soit tous les exemples associés au nœud appartiennent à une seule classe, soit le gain d'information apporté par chacun des attributs est nul. Cela signifie qu'il n'y a plus d'intérêt à partitionner la sous-base en question.

Les algorithmes de construction d'arbres de décision peuvent être complétés par des techniques d'élagage. En relâchant ces critères, le partitionnement s'arrête quand toutes les sous-bases d'apprentissage associées aux feuilles sont relativement homogènes. L'entropie sert alors dans la condition d'arrêt pour évaluer l'homogénéité.

Dans le cas où le seuil d'entropie est strictement positif, l'algorithme s'arrête si l'entropie de l'ensemble de tous les éléments est inférieure au seuil ε fixé par avance, ainsi l'algorithme est capable de tolérer des données bruitées.

$$I(\xi) \leq \varepsilon$$

où I est l'entropie de l'ensemble des exemples considéré. Dans le cas où ce seuil vaut 0, on revient au cas classique : l'algorithme s'arrête si tous les éléments de l'ensemble associé au nœud ont la même classe.

Le partitionnement peut également s'arrêter quand le gain d'information est petit :

$$\forall A \ : \ I(\xi) - I(\xi|A) < \varepsilon$$

Il est aussi possible d'arrêter la construction en fonction d'autres critères que la discrimination relative aux classes, par exemple la taille des sous-bases. Les critères d'arrêt influent bien sur la taille d'un arbre et sur sa capacité de généralisation. Des discussions plus détaillées sur des méthodes d'élagage se trouvent, entre autres, dans [61, 28].

2.2.3 Utilisation d'arbres de décision

Dans le cas le plus classique, le processus de construction s'arrête quand tous les exemples associés à une feuille appartiennent à une même classe. Lors de la classification, un exemple n'arrive qu'à une seule feuille, donc sa classe est l'étiquette de la feuille.

Le problème devient plus compliqué quand on ajoute des critères d'arrêt concernant le cardinal ou l'entropie de l'ensemble des exemples associés à la feuille comme nous les avons mentionnés précédemment. Cet ensemble n'est donc pas toujours homogène. Cependant, un exemple arrive encore à une seule feuille lors de sa classification. La classe de l'exemple est la classe majoritaire de la feuille. On peut cependant exploiter la distribution des exemples dans cet ensemble pour obtenir une réponse probabiliste sur la classe de l'exemple [123].

Certains travaux proposent l'usage de plusieurs arbres construits à partir des variantes de la base d'apprentissage initiale (boosting, bagging, transformation en plusieurs problèmes de classification binaire). D'ailleurs, à partir d'une même base d'apprentissage, plusieurs arbres peuvent être construits par différentes techniques. Ces arbres peuvent également être combinés. Dans ces cas, une stratégie de vote est nécessaire. En disposant de plusieurs mesures de discrimination, nous pensons à utiliser plusieurs arbres construits par différentes mesures. Cette technique peut être combinée avec d'autres comme des techniques floues et des techniques de bagging et boosting, par exemple.

2.3 Sélection du meilleur attribut

La sélection du meilleur attribut pour classer des exemples est essentielle dans la construction des arbres de décision. La différence entre des méthodes réside souvent

dans cette étape. Dans cette section, nous établissons d'abord quelques méthodes principales de sélection d'attribut. Ensuite, nous proposons l'utilisation des mesures de discrimination générales dans cette étape. Cette section se termine par un série d'expériences afin de caractériser ces nouvelles mesures.

2.3.1 Méthodes de sélection

Il n'existe pas de moyen efficace pour trouver un arbre *vraiment optimal* au sens où il classe parfaitement la base d'apprentissage et où il minimise le nombre moyen de tests nécessaires pour un exemple inconnu. De manière heuristique, à chaque étape on cherche à partitionner la base selon l'attribut le plus discriminant afin de minimiser la taille de l'arbre. Le choix repose sur le principe suivant : l'attribut choisi doit permettre de réduire la plus possible l'incertitude dans laquelle on se trouve lors de l'identification des classes des exemples. Autrement dit, la question posée sur la valeur de l'attribut permet de caractériser le plus possible la classe des exemples.

On distingue souvent deux catégories de critères de choix [102, 125]. La première catégorie contient des critères provenant de la théorie de l'information. Ils s'appuient généralement sur des mesures d'entropie telle que l'entropie de Shannon, l'entropie de Daróczy, l'entropie de Rényi, etc. La deuxième correspond à des critères variés, y compris des critères statistiques. Dans la bibliographie [139, 110, 138, 109, 40] plusieurs méthodes sont recensées. Quelques méthodes usuelles sont brièvement décrites ci-dessous.

L'utilisation de l'entropie de Shannon [141] a été introduite dans la théorie des questionnaires [118, 119] et a été reprise par Quinlan dans ID3 [122] pour choisir le meilleur attribut. Cela sert de base à de nombreuses méthodes. Le principe est de maximiser le gain d'information apporté par la question portant sur l'attribut en question ou bien rendre les sous-bases issues des valeurs de l'attribut les plus homogènes possible. Le gain d'information est la différence entre l'entropie de Shannon de la base *avant* la question et *après* la réponse obtenue sur la valeur de l'attribut. Cette heuristique vise à minimiser la taille de l'arbre.

Soit ξ une base d'exemples décomposée en n classes C_i et notons $P(C_i)$ (pour simplifier, notons P_i) la probabilité de la classe C_i ($1 \leq i \leq n$ et $\sum_{i=1}^{n} P_i = 1$). L'entropie de Shannon de ξ est définie par :

$$I_S(\xi) = I_S(P_1, P_2, .., P_n) = -\sum_{i=1}^{n} P_i \log P_i$$

Supposons que ξ soit partitionnée en m sous-ensembles $\xi_{v_1}, \xi_{v_2}, .., \xi_{v_m}$ correspondant à des valeurs $v_1, v_2, .., v_m$ de l'attribut A et ayant chacun pour entropies respectives $I(\xi_{v_1}), I(\xi_{v_2}), .., I(\xi_{v_m})$.

L'entropie de la base conditionnée par les valeurs de l'attribut A, $I(\xi|A)$, est définie comme la somme pondérée des entropies $I(\xi_{v_j})$:

$$I(\xi|A) = \sum_{j=1}^{m} P(v_j) I(\xi_{v_j}) \tag{2.1}$$

Le gain d'information apporté par la question sur la valeur de A est alors :

$$\triangle I(A, \xi) = I(\xi) - I(\xi|A)$$

Le meilleur attribut pour partitionner la base lors de la construction d'un arbre de décision est celui qui maximise le gain d'information.

Dans leur méthode CART [27], Breiman *et al.* ont proposé d'utiliser une mesure d'impureté basée sur le test statistique de Gini :

$$I_G(\xi) = I_G(P_1, P_2, .., P_n) = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} P_i P_j$$

Le *gain ratio* a été proposé par Quinlan [124] pour éliminer le biais des mesures précédentes qui favorisent les attributs ayant plusieurs valeurs. Cela conduit souvent à des arbres de grande taille. La gain ratio est défini comme :

$$I_Q(\xi|A) = \frac{I_S(\xi|A)}{I_S(A)}$$

où

$$I_S(A) = -\sum_{j=1}^{m} P(v_j) \log P(v_j)$$

Pour résoudre le même problème, López de Mántaras [52] a proposé une mesure basée sur un calcul de distance entre les partitions. Il utilise la formule ci-dessous au lieu du gain d'information dans l'algorithme ID3 :

$$I_L(\xi|A) = \frac{I_S(\xi|A) + I_S(A|C)}{I_S(A \wedge C)}$$

où

$$I_S(A|C) = -\sum_{i=1}^{n} P_i \sum_{j=1}^{m} P(v_j|C_i) \log P(v_j|C_i)$$

et

$$I_S(A \wedge C) = -\sum_{i=1}^{n} \sum_{j=1}^{m} P(v_j \wedge C_i) \log P(v_j \wedge C_i)$$

Dans la méthode SIPINA[1][166], les mesures d'entropie de Shannon, de Gini et de Daróczy sont utilisées. Cependant, à côté des processus de partitionnement, un processus de regroupement est proposé pour fusionner les ensembles. Ce processus conduit à une meilleure résistance à la fragmentation des données. L'application de

[1]http ://eric.univ-lyon2.fr/∼ricco/sipina.html

cette méthode, qui combine ces deux processus, n'est plus un arbre mais un graphe latticiel.

Pour une méthode de construction d'arbres de décision plus adaptée à des problèmes dont les classes sont déséquilibrées, Zighed *et al.* [100, 168] ont proposé une mesure d'entropie asymétrique où la distribution $(\theta_1, \theta_2, .., \theta_n)$ pour laquelle la mesure d'entropie est maximale est choisie par les utilisateurs en fonction de l'application (cf. section 1.3.2).

$$I_{asym}(P_1, P_2, .., P_n) = -\sum_{i=1}^{n} \frac{P_i(1-P_i)}{(-2\theta_i+1)P_i + \theta_i^2}$$

Ils ont, par exemple, appliqué cette mesure à la détection de cancer du poumon [101]. La formule conditionnelle qu'ils utilisent est la formule de Type 1 selon la notation adoptée dans ce mémoire.

Il existe également des méthodes de construction d'arbres de décision qui n'utilisent pas de mesures d'entropie. Entre autres, on peut citer la distance de Kolmogorov-Smirnov utilisée dans [152] ou la méthode de Van de Merckt [53]. La distance de Kolmogorov-Smirnov est définie dans le cas à deux classes comme :

$$K_x(C_1, C_2) = \max_x(|F_{C_1}(x) - F_{C_2}(x)|)$$

où x est une variable sur le domaine d'un attribut et $F_{C_i}(x)$ est la fonction de distribution de probabilité de la classe C_i sur ce domaine. Le critère selon lequel la base est partitionnée s'appuie sur la comparaison entre la valeur de l'attribut en question et x.

Dans [53], une mesure de similarité est incorporée dans la mesure de discrimination. La mesure de similarité est une mesure de contraste entre des partitions engendrées par les classes, de l'ensemble des valeurs de l'attribut numérique A. Dans le cas à deux classes C_1 et C_2, la mesure de contraste est définie comme :

$$I_{contraste}(\xi|A) = \frac{|C_1|.|C_2|}{|C_1 \cup C_2|}(m_1 - m_2)^2$$

où m_1 et m_2 sont respectivement la moyenne des valeurs de A pour les exemples de la classe C_1 et de la classe C_2 ; $|C_1|$ et $|C_2|$ sont respectivement les cardinaux de l'ensemble des exemples possédant les classes C_1 et C_2. La mesure de discrimination est alors définie comme :

$$I_V(\xi|A) = \frac{I_{contraste}(\xi|A)}{I_S(\xi|A)}$$

Pour terminer, il faut aussi citer les méthodes statistiques qui font appel à des tests statistiques pour évaluer la corrélation entre un attribut et la classe afin de choisir l'attribut le plus adéquat. Entre autres, la méthode CHAID (*CHi-square Automatic Interaction Detection*) propose d'utiliser le test du χ^2 pour évaluer la dépendance entre un attribut et la classe. Les lecteurs intéressés par ces méthodes pourront se référer à [127].

Un certain nombre de mesures sont également étudiées dans [159, 160, 87]. Entre autres, [160] et [87] ont analysé le biais des mesures dans la sélection des attributs afin de connaître le comportement de chacune face à des différents types d'attributs.

2.3.2 Mesures de discrimination en sélection d'attribut

Comme nous l'avons expliqué ci-dessus, l'entropie de Shannon sert très couramment dans les méthodes de construction d'arbres de décision. Cela peut se comprendre car elle possède plusieurs propriétés intéressantes.

En théorie de l'information, il existe d'autres mesures d'entropie. Dans la section 1.2.1.2 de telles mesures sont recensées : l'entropie de Daróczy, l'entropie de Rényi, la R-norme entropie, etc. Chacune d'entre elles est caractérisée par un ensemble de propriétés. Ces mesures ne sont généralement pas étudiées et utilisées en apprentissage inductif. Néanmoins, certaines formes spéciales de ces mesures sont utilisées, par exemple, la formule de Gini, un cas particulier de l'entropie de Daróczy, a été utilisée dans l'algorithme CART [27]. L'entropie de Shannon peut être considérée comme un cas particulier de l'entropie de Daróczy, l'entropie de Rényi, la R-norme entropie quand le coefficient β tend vers 1.

Dans l'état de l'art, des formules conditionnelles sont généralement une somme pondérée des mesures d'entropie des sous-ensembles. Si la somme la plus utilisée est pondérée par les probabilités selon l'équation (2.1), les autres formules existantes sont pondérées par une fonction des probabilités (cf. les équations (1.11), (1.12), (1.13) page 21). Les pondérations de Type 2 et de Type 3 dépendent d'un coefficient positif β.

Nous proposons dans cette thèse d'étudier l'utilisation de ces mesures dans l'induction d'arbres de décision. Ce sont des mesures plus générales que celle de Shannon et qui n'ont pas encore été utilisées dans un tel processus. Plus concrètement, l'entropie d'un ensemble d'exemples ξ ou ξ_{v_j} $(j = 1, .., m)$ peut être évaluée par une des mesures qui satisfait les propriétés du niveau $\mathcal{G}$ du modèle hiérarchique présenté dans le chapitre 1, entre autres, celle de Rényi (équation (1.2)), celle de Daróczy (équation (1.3)), la R-norme entropie (équation (1.4)). L'entropie conditionnelle est une agrégation des entropies qui satisfait le niveau $\mathcal{H}$ du modèle hiérarchique, en particulier, les formules conditionnelles de Type 1, de Type 2, de Type 3, et de Type 4. Grâce au modèle hiérarchique, ces mesures sont validées comme mesures de discrimination.

L'introduction de ces nouvelles mesures dans la construction d'arbres de décision offre des alternatives dans le choix des mesures. Les mesures différentes conduisent à des arbres ayant des propriétés différentes et les utilisateurs peuvent éventuellement sélectionner la mesure qui produit des arbres convenant le mieux à leurs propres problèmes, par exemple, des arbres équilibrés ou des arbres de petite taille. Cela nécessite une étude sur le comportement de chaque entropie dans les processus où elles peuvent éventuellement intervenir comme mesure de discrimination, notamment dans la sélection de l'attribut. Certains travaux sont effectués sur cette question, on peut citer, entre autres, les travaux de Breiman [26] et ceux de Marsala [106].

Breiman cherche à caractériser la partition idéale qui partitionne la base initiale en un nombre fixe de parties et qui rend toutes les parties les plus homogènes possible. L'homogénéité est évaluée par une mesure telle que l'entropie de Shannon ou l'indice de Gini. Quant à lui, Marsala caractérise les composants $\mathcal{FGH}$ de différentes mesures de construction d'arbres de décision flous telles l'entropie de Shannon, l'indice de Gini et la mesure d'ambiguïté de Yuan et Shaw.

2.3.3 Expérimentations

Des expérimentations ont été menées avec plusieurs bases de données pour valider l'utilisation de différentes entropies introduites et validées théoriquement dans le chapitre 1. Ce sont les mesures d'entropie de Daróczy (Type 1, 2, 3) et les mesures d'entropie de Rényi (Type 1, 2, 3). Le système DTGen (présenté dans l'annexe) a été utilisé pour construire des arbres de décision avec différentes entropies et leurs formules conditionnelles associées.

Le protocole d'expérimentation est décrit comme suit. Il sera repris plus tard dans d'autres expérimentations.

Soit une base de données. Dans la première étape, une sélection d'exemples est réalisée pour partitionner cette base en deux parties, l'une sert de base d'apprentissage et l'autre sert de base de test :

1. Partitionner des exemples de la base initiale par leurs classes. On obtient des ensembles d'exemples par classe.
2. Choisir aléatoirement à partir de chaque ensemble 50% des exemples.
3. Regrouper tous les exemples choisis dans l'étape précédente pour former la base d'apprentissage. Ainsi, la base d'apprentissage contient 50% des exemples de chaque classe.
4. Regrouper tous les restes pour former la base de test.

Dans la deuxième étape, DTGen construit un arbre de décision à partir de la base d'apprentissage et l'utilise ensuite pour classer les exemples de la base de test. Avant la sélection du meilleur attribut, les attributs numériques sont discrétisés par la méthode qui minimise l'entropie de Shannon que nous évoquons dans la section 2.4. Le taux de bonnes classifications et les autres indices d'évaluation sont calculés, en particulier la profondeur et le nombre de feuilles. La profondeur maximale, la profondeur minimale correspondent respectivement au nombre maximal et au nombre minimal de questions à poser pour identifier la classe d'un exemple. La profondeur moyenne est la longueur moyenne de la racine aux feuilles. Une autre méthode d'agrégation considérée est de pondérer la hauteur moyenne par la probabilité a priori qu'un exemple suive chaque chemin. Cette probabilité est estimée par la base d'apprentissage. Elle est définie comme le nombre moyen de questions nécessaires quand l'arbre est utilisé pour classer les exemples de la base d'apprentissage. Dans le cas où la base d'apprentissage a la même distribution des classes que la base de test, ce nombre moyen de questions est valable pour la base de test.

L'expérimentation décrite ci-dessus est répétée plusieurs fois (8 fois pour notre expérimentation) sur une même base d'exemples initiale. À la fin, les résultats de tous les tests sur la base sont agrégés pour obtenir le résultat final sur la base en question.

Base de données	#exemples	#attributs	#classes	Distribution	% classe maj.
Iris	150	4	3	$3 \times 33.3\%$	33.3
Balance scale	625	4	3	46.8% + 46.8% + 7.4%	46.8
E. coli	336	7	8	42.6% + 22.9% + 15.5% + ...	42.6
Glass identification	214	10	7	35.5% + 32.7% + 13.5% + ...	35.5
Ionosphere	351	34	2	64.2% + 35.8%	64.2
Liver-disorders	345	6	2	58.0% + 42.0%	58.0
Pima Indians diabetes	768	8	2	65.1% + 34.9%	65.1
Wine recognition	178	13	3	39.9%+ 33.1% + 27%	39.9
Waveform	300	21	3	$3 \times 33.3\%$	33.3

Tab. 2.1 – Description des bases de données d'UCI utilisées

Les expérimentations effectuées dans les différentes parties de la thèse ont été menées avec des bases d'UCI [113], à l'exception de la base « Waveform » obtenue par un générateur automatique [27]. Quelques caractéristiques de ces bases, en particulier le nombre d'exemples, le nombre d'attributs, la distribution des classes et la proportion (en pourcentage) des exemples de la classe majoritaire sont décrites dans le tableau 2.1. Ces caractéristiques donnent une idée sur la difficulté des bases différentes.

Les figures de 2.2 à 2.7 montrent le taux moyen de bonnes classifications, le nombre de feuilles et la profondeur obtenus sur toutes les bases en faisant varier le coefficient β de 0 à 50. Les valeurs sont choisies de manière représentative. L'entropie de Shannon ne dépend pas de la valeur de β, sur les figures, elle est donc représentée par une ligne horizontale. Sur les courbes, chaque point représente un indice de plusieurs bases moyennées en fonction de l'entropie et du β lui correspondant. Dans les figures, les légendes « Rényi i » , « Daroczy i » ($i = 1, 2, 3$) correspondent respectivement à l'entropie conditionnelle de Rényi de Type i et de l'entropie conditionnelle de Daróczy de Type i.

Globalement, les résultats montrent que les taux de bonnes classifications obtenus par les entropies conditionnelles de Rényi et celles de Daróczy sont légèrement différents du résultat obtenu avec l'entropie conditionnelle de Shannon. L'écart maximal est généralement inférieur à 2%. Dans les cas extrêmes quand β s'approche de 0 ou $\beta = 50$, cet écart monte jusqu'à environ 4%. Avec certaines valeurs β entre 2,5 et 6, l'entropie de Daróczy de Type 1 donne un taux de bonnes classifications légèrement

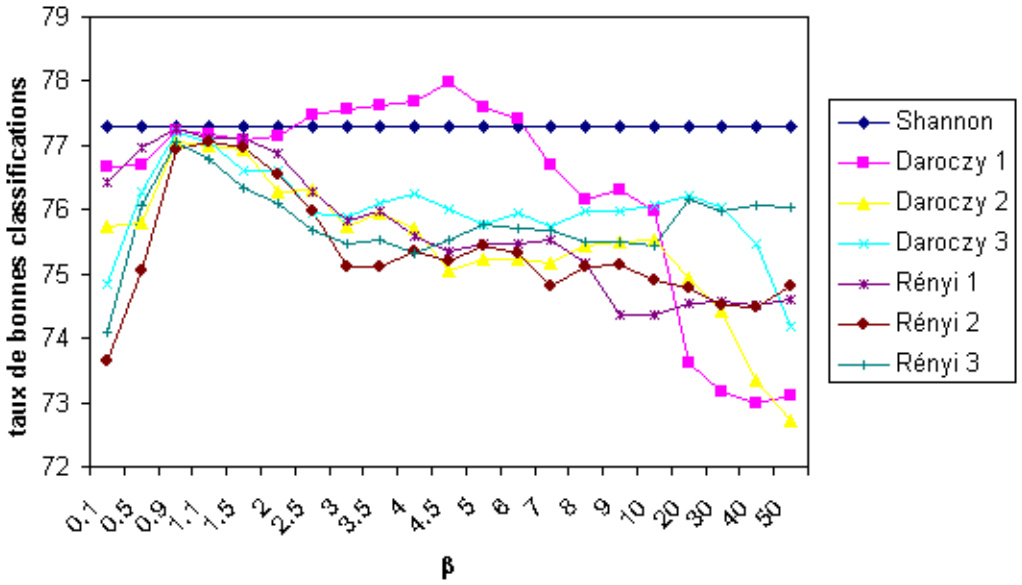

FIG. 2.2 – Taux de bonnes classifications moyenné sur différentes bases, $\beta \in (0, 50]$

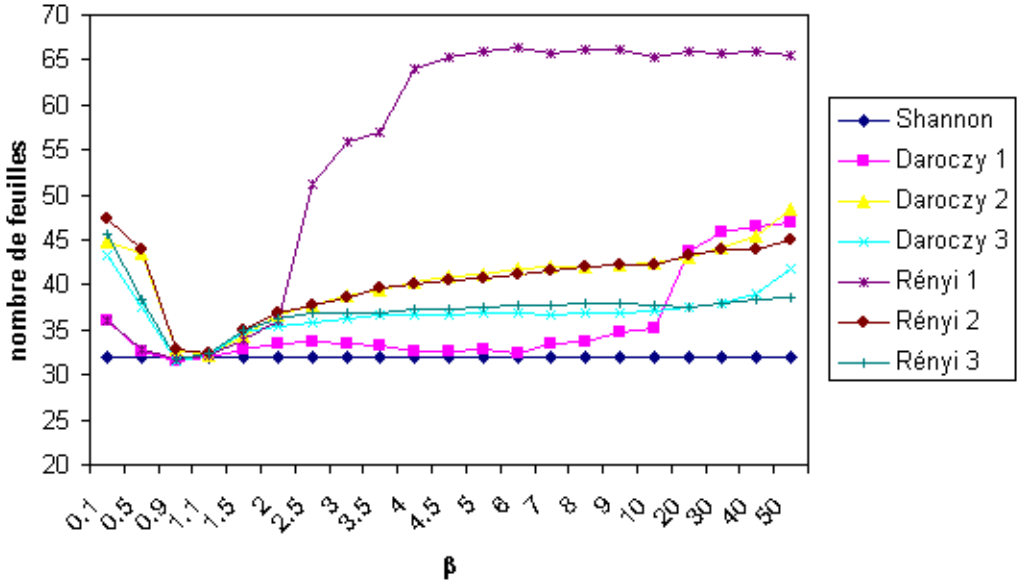

FIG. 2.3 – Nombre de feuilles moyenné sur différentes bases, $\beta \in (0, 50]$

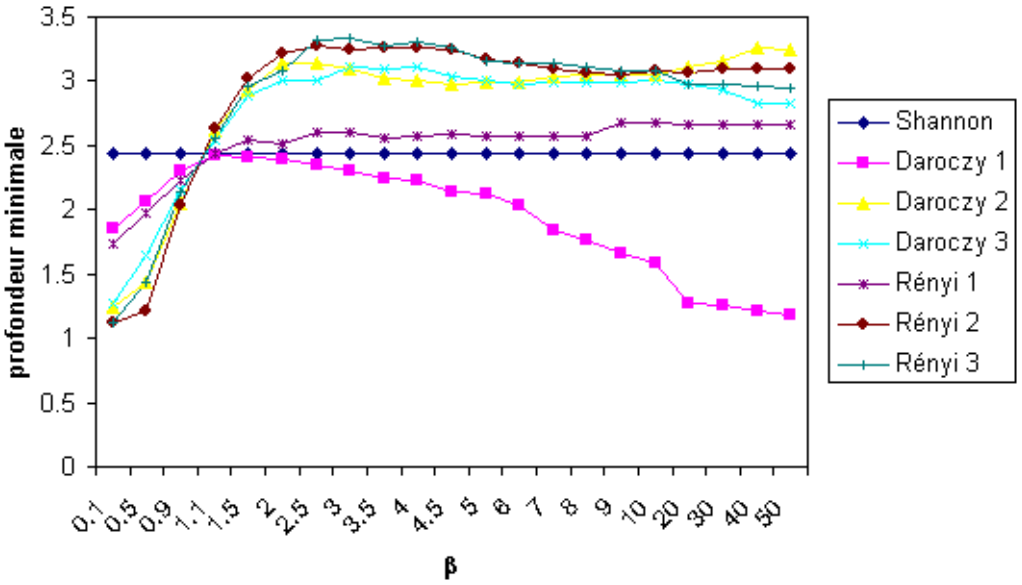

FIG. 2.4 – Profondeur minimale moyennée sur différentes bases, $\beta \in (0, 50]$

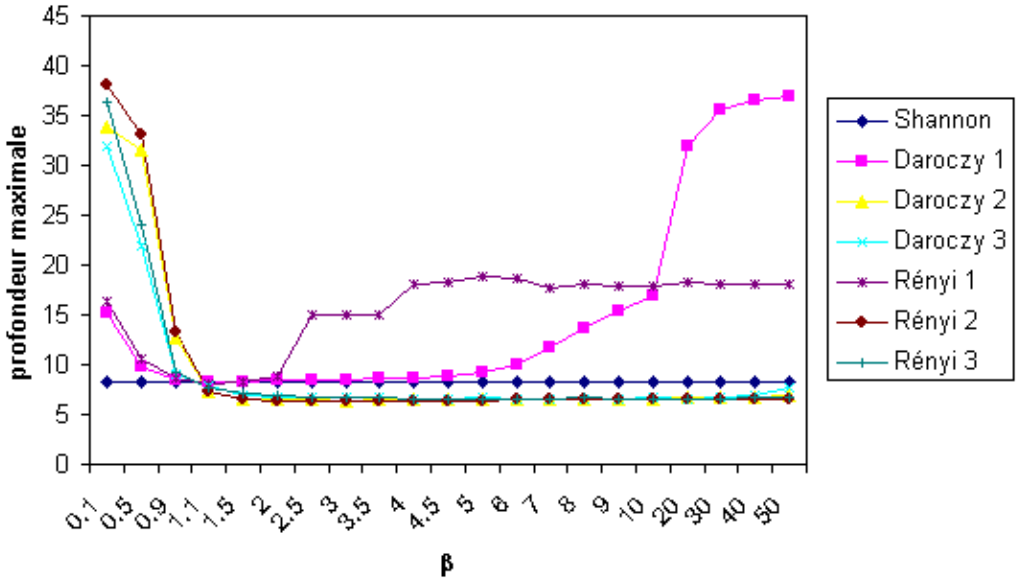

FIG. 2.5 – Profondeur maximale moyennée sur différentes bases, $\beta \in (0, 50]$

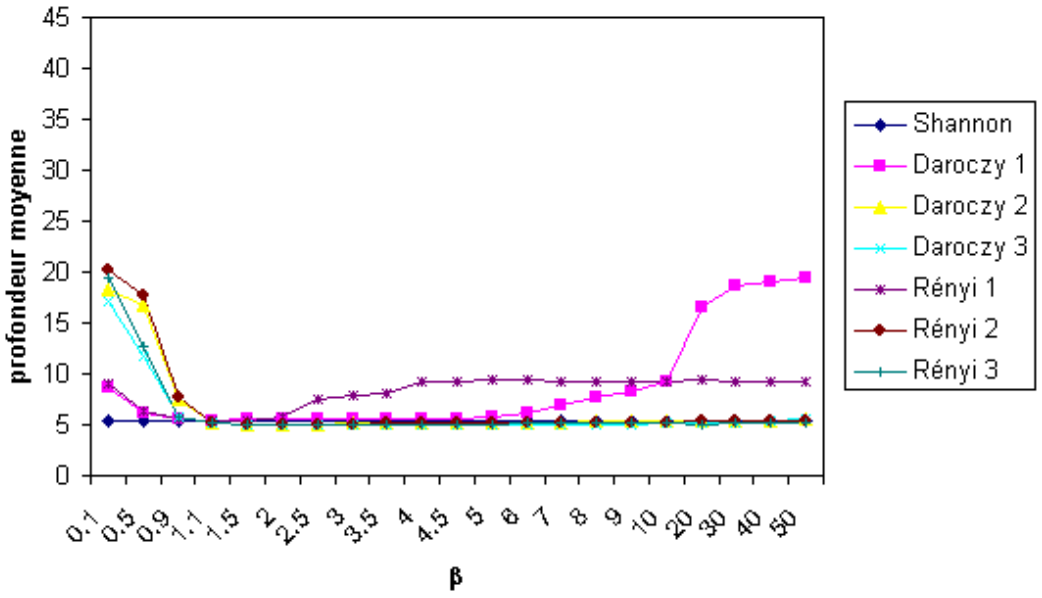

FIG. 2.6 – Profondeur moyenne des arbres moyennée sur différentes bases, $\beta \in (0, 50]$

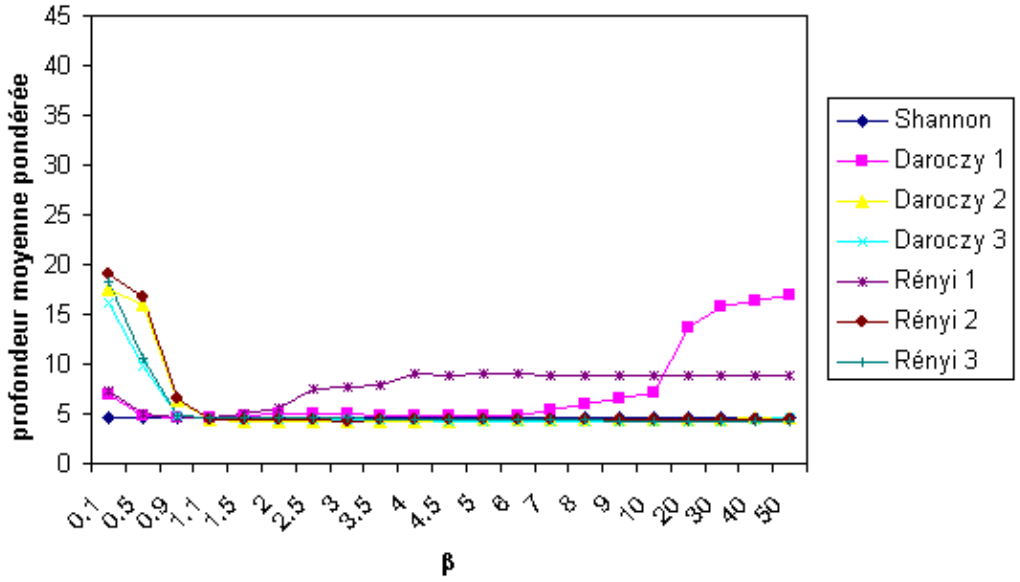

FIG. 2.7 – Profondeur moyenne pondérée par les nombres d'exemples correspondant à chaque feuille, moyennée sur différentes bases, $\beta \in (0, 50]$

meilleur que celui de l'entropie conditionnelle de Shannon. Dans les autres cas, la mesure de Shannon conduit à des résultats légèrement supérieurs.

Sur la figure 2.2, on constate une tendance très nette : quand β varie de 0 à 1, le taux de bonnes classifications augmente pour toutes les mesures. Lorsque β dépasse 1, le taux de bonnes classifications a tendance de se dégrader légèrement sauf dans le cas de l'entropie conditionnelle de Daróczy de Type 1. Avec cette dernière, le taux de bonnes classifications augmente jusqu'à ce que $\beta = 5$, puis il commence à diminuer. La diminution est relativement nette quand β est grand.

Les arbres obtenus avec des petites valeurs de β ont généralement plus de feuilles (figure 2.3). Le nombre de feuilles diminue quand β tend vers 1. La diminution la plus forte correspond à l'entropie conditionnelle de Rényi de Type 2. Quand $\beta > 1$ augmente, l'entropie conditionnelle de Rényi de Type 2 et l'entropie conditionnelle de Daróczy de Type 2 possèdent le même comportement : elles entraînent une augmentation du nombre de feuilles. Dans l'intervalle [1, 5], quand β augmente, l'entropie conditionnelle de Rényi de Type 1 entraîne une augmentation forte du nombre de feuilles. Au delà de cet intervalle, ce nombre devient assez stable. L'entropie conditionnelle de Daróczy de Type 1 n'entraîne pas d'augmentation significative jusqu'à ce que $\beta = 10$. Ensuite, avec cette mesure le nombre de feuilles augmente avec β. Les comportements des mesures de Type 3 sont similaires : ils sont assez stables après une légère augmentation quand β est un peu supérieur à 1.

Par contre, la profondeur minimale devient faible dans le cas où β s'approche de 0 (figure 2.4). Elle augmente selon β et devient stable quand $\beta > 3$ excepté pour les entropies conditionnelles de Type 1. Si l'entropie de Rényi de Type 1 devient stable un peu plus tôt que les autres, l'augmentation de β dans la formule de l'entropie conditionnelle de Daróczy de Type 1 entraîne une diminution de la profondeur minimale. Cette diminution est relativement significative et la profondeur moyenne tend vers 1.

La variation de la profondeur moyenne (figure 2.6 et figure 2.7) et maximale (figure 2.5) est différente de celle de la profondeur minimale. Ces indices de profondeurs ont le même comportement. Elles décroissent assez rapidement quand β varie de 0 à 1. Plus tard, elles deviennent plutôt stables sauf pour les entropies conditionnelles de Type 1. Quand β augmente à partir de 1, on observe une augmentation de la profondeur des arbres lorsque l'entropie de Rényi de Type 1 est utilisée. Ensuite, à partir de $\beta = 4.5$ avec la même mesure, la profondeur ne change plus beaucoup. Lorsque β est grand, les arbres construits par l'entropie de Daróczy de Type 1 deviennent relativement profonds. Les arbres sont ainsi très déséquilibrés dans ce cas car la profondeur minimale est faible avec la même mesure.

Pour caractériser de manière plus fine ces comportements, nous faisons varier le coefficient β de 0 à 10 avec un pas plus petit : 0.1. Les figures 2.8 et 2.9 montrent le taux moyen de bonnes classifications et la profondeur moyenne obtenus sur un ensemble réduit de 5 bases : « Iris » , « E. coli » , « Wine » « Pima Indians Diabetes » et « Waveform » . Pour souligner l'apport des mesures de discrimination, une expérimentation avec une sélection aléatoire du meilleur attribut à chaque itération a été menée. S'il est aléatoirement choisi, le taux de bonnes classifications (75.15%)

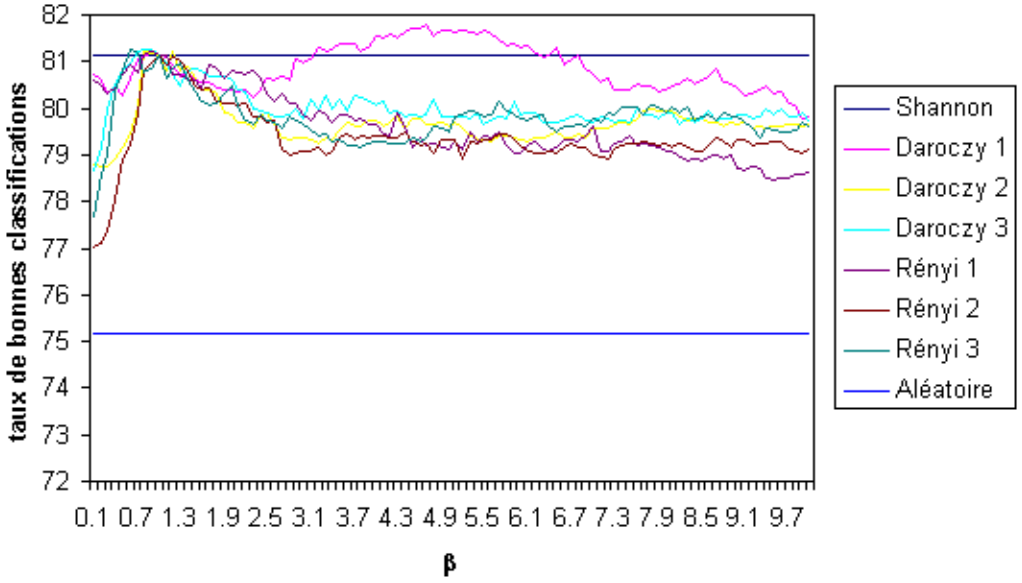

FIG. 2.8 – Taux de bonnes classifications moyenné sur 5 bases, $\beta \in (0, 10]$

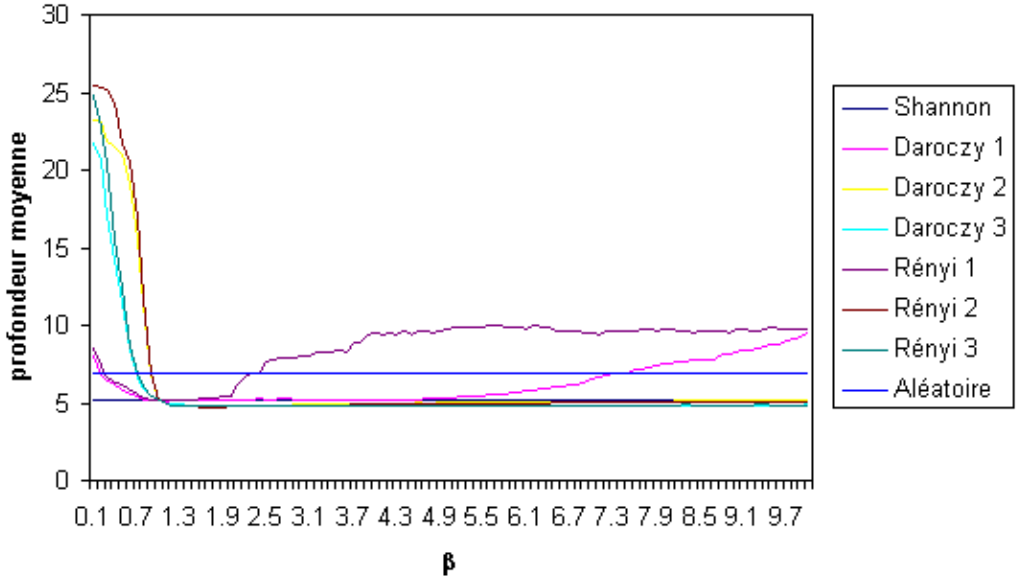

FIG. 2.9 – Profondeur moyenne des arbres moyennée sur 5 bases, $\beta \in (0, 10]$

est toujours significativement plus mauvais qu'avec d'autres méthodes. Les courbes de ces figures s'accordent bien avec toutes les remarques faites dans les paragraphes ci-dessus.

2.3.4 Discussion

Nous avons introduit dans cette section l'utilisation des mesures de discrimination dans la sélection du meilleur attribut. D'une part, comme nous l'avons montré dans le chapitre 1, ces mesures sont qualifiées comme des mesures de discrimination par le modèle hiérarchique. D'autre part, l'étude expérimentale menée dans cette section permet de caractériser ces mesures empiriquement. Ces deux études, l'une complétant l'autre, justifient l'utilisation de ces mesures.

Ces résultats montrent les différences entre des entropies conditionnelles utilisées. Ces différences résident principalement dans la taille et la forme des arbres.

Ceci est accord avec le résultat obtenu par [128] quand ils ont effectué des expérimentations similaires avec quelques entropies conditionnelles de Daróczy de Type 1 ($\beta = 1, 1.5, 2, 3, 5$). Différant de ce que nous avons fait, ils ont utilisé des données artificielles générées d'après le concept *M-of-N*[2] afin de faciliter le contrôle de la complexité des données. Sur ces 4 valeurs de β, les taux de bonnes classifications qu'ils ont obtenus semblent cohérents avec ce que nous présentons dans la figure 2.2.

Le résultat obtenu dépend non seulement des entropies utilisées mais aussi de la manière de les combiner et de la valeur de β. Il s'avère qu'il n'est pas très intéressant d'utiliser des valeurs trop petites ou trop grandes pour β. Nous proposons de renforcer les expérimentations pour obtenir des résultats plus fiables. Premièrement, nous souhaitons mettre en place un protocole d'expérimentation plus rigoureux. Entre autres, la manière de traiter les résultats est à raffiner en appliquant strictement des règles statistiques. Deuxièmement, plus de bases de données doivent être utilisées, en particulier des bases ayant des données symboliques. Enfin, il serait intéressant d'évaluer également les arbres par d'autres critères. D'autres méthodes d'agrégation des entropies pour avoir une formule conditionnelle sont également à expérimenter.

2.4 Discrétisation des attributs numériques

Les attributs numériques ont des caractéristiques différentes par rapport aux attributs symboliques. Ils prennent leurs valeurs dans un ensemble dont le nombre d'individus est très grand. C'est souvent l'ensemble des nombres réels ou des entiers. Ses valeurs sont donc ordonnées.

Cependant, certains algorithmes ne traitent que des données symboliques ou ils ne manipulent efficacement que cette catégorie de données. L'algorithme de construction d'arbres de décision a été initialement conçu pour traiter des valeurs symboliques, par exemple l'algorithme ID3 [122]. Face à des valeurs numériques, ces al-

[2]Soit N variables booléennes indépendantes. Le concept *M-of-N* est la variable qui est *vraie* si et seulement si au moins M parmi N variables sont *vraies*.

gorithmes les considèrent telles qu'elles sont : les valeurs numériques sont traitées comme des valeurs symboliques. Les techniques de sélection du meilleur attribut favorisent souvent des attributs ayant un grand nombre de valeurs. En conséquence, les attributs numériques sont choisis en premier. On obtient ainsi des arbres de très grande taille. La capacité de généralisation et la performance des arbres sont diminuées parce que les valeurs numériques sont trop précises.

Le traitement d'une valeur numérique comme une valeur symbolique ne tient pas compte de la proximité entre des valeurs numériques. Or deux valeurs proches portent souvent des significations similaires, alors qu'on les traite directement comme des valeurs symboliques distinctes, elles apparaîtront totalement différentes. De plus, les données peuvent être bruitées : la proximité entre les valeurs n'est pas négligeable et il faut la prendre en compte.

Un regroupement adéquat des valeurs numériques permet de regrouper des valeurs similaires dans un même intervalle pour leur appliquer un traitement unique. Le biais provenant des données bruités est ainsi réduit. La sémantique des données est mieux exploitée. Cela renforce l'interprétabilité des règles représentées par l'arbre de décision obtenu.

C'est pourquoi une technique usuelle consiste à insérer une étape supplémentaire dans le processus de construction d'arbres de décision afin de discrétiser les attributs numériques.

Dans cette section, nous abordons la discrétisation automatique de données dans le contexte de la construction d'arbres de décision. D'abord, des caractéristiques principales de la discrétisation automatique sont brièvement présentées. Ensuite l'utilisation des mesures de discrimination dans la discrétisation est proposée. Afin de caractériser ces mesures dans la discrétisation, nous définissons une mesure d'équilibre. Une expérience est menée pour étudier le comportement de ces nouvelles mesures.

2.4.1 Description générale

La phase de discrétisation a pour but de découper un domaine numérique en un certain nombre d'intervalles. Les attributs discrétisés seront considérés ensuite comme des attributs symboliques. À chaque nœud, au lieu de poser une question qui porte sur la valeur précise de l'attribut choisi, une question sur l'intervalle dans lequel se trouve la valeur considérée est posée. Chaque point de coupure apparaissant dans la discrétisation peut être considéré comme un test. Un intervalle est associé à un seul test s'il a une extrémité à l'infini, du type $v < c$ ou $v > c$. Il est associé à deux tests s'il est limité par deux points : $a < x < b$.

Pour une formulation formelle du problème, voir [98, 169].

Dans certains problèmes simples, la discrétisation peut se faire manuellement. Les données sont regroupées par un expert selon leur sémantique. Évidemment, cette méthode n'est pas applicable aux données complexes, notamment quand la sémantique de données n'est pas explicite ou quand les données sont volumineuses. La discrétisation interactive est également possible. La connaissance experte est intégrée dans la discrétisation mais la vitesse du processus n'est souvent pas assurée.

Plusieurs méthodes de discrétisation sont disponibles (voir par exemple [98, 126]). Elles sont caractérisées par un nombre de propriétés. Dans la construction des arbres de décision en présence d'attributs numériques, les attributs discrétisés doivent être disponibles avant la sélection du meilleur attribut. Il est donc possible que l'attribut soit discrétisé une fois pour toutes avant d'appliquer l'algorithme d'induction classique. Ainsi, une fois les points de coupure trouvés, dans toutes les itérations de l'algorithme de construction d'arbres qui suivent, le même regroupement est utilisé. L'inconvénient de cette stratégie est qu'elle exclut le contexte dans lequel la discrétisation intervient à chaque itération. Il s'avère que la sémantique d'une coupure dépend du contexte qui est caractérisé par l'ensemble des questions posées sur des attributs pour conduire au nœud où les exemples sont localisés. Une stratégie plus convenable est de discrétiser les attributs à chaque itération, notamment juste avant l'évaluation des attributs pour choisir le meilleur.

Les méthodes de discrétisation sont principalement caractérisées suivant 2 axes différents, selon qu'elles appliquent la stratégie top-down ou bottom-up et selon qu'elles se placent dans un mode supervisé ou non-supervisé.

La stratégie top-down cherche à décomposer au fur et à mesure le domaine numérique en plusieurs intervalles. Au début, il y a un seul intervalle pour toutes les valeurs numériques. La méthode dans C4.5 [124] qui minimise une entropie conditionnelle de Shannon fait partie de ce groupe de méthodes. On peut citer aussi des méthodes naïves comme la méthode qui divise des valeurs en partitions de même longueur ou de même nombre d'exemples, la méthode Zeta [72].

La stratégie bottom-up cherche à fusionner des intervalles élémentaires dont chacun ne contient qu'une seule valeur ou un ensemble de valeurs correspondant à des exemples d'une même classe. Les méthodes MDLPC [169, 11], Chi2 [99], FUSINTER [169] font partie des méthodes bottom-up.

Une courte description des modes supervisé vs. non-supervisé des méthodes se trouve dans [169] et une étude plus détaillée est présentée dans [57]. Les méthodes supervisées sont caractérisées par leur utilisation des classes des exemples. Pour les méthodes non-supervisées, on ne s'intéresse qu'aux valeurs d'attributs et non à la classe des exemples. On revient dans ce cas au problème de regroupement (*clustering*) dans l'espace d'une dimension. On peut citer dans le cadre des méthodes non-supervisées : la discrétisation en intervalles de même longueur (*equal interval width method*), la discrétisation en intervalles ayant un même nombre de valeurs (*equal frequency method*), la discrétisation aléatoire (*random discretize*) [7]. Dans le cadre de l'étude des arbres de décision, il n'est pas intéressant d'aborder le mode non-supervisé car le but recherché dans ce contexte est de trouver le rapport entre les valeurs d'attribut et la classe des exemples.

2.4.2 Méthodes de discrétisation basées sur une entropie

Parmi les méthodes qui suivent la stratégie top-down, on utilise souvent une mesure dérivant d'une mesure d'entropie (*entropy-based* et *purity-based methods*), par exemple, les méthodes CART [27], FUSINTER [169], etc. L'entropie de Shannon

est couramment utilisée. Nous proposons d'ajouter dans les choix possibles d'autres formules d'entropie existantes telles que l'entropie de Rényi, l'entropie de Daróczy et la R-norme entropie.

Supposons que les valeurs de l'attribut A soient ordonnées et que le domaine de A se discrétise en h intervalles disjoints deux à deux d_1, d_2, .., d_h. Cette discrétisation est notée par D. L'entropie de l'ensemble des exemples dont la valeur de $A \in d_j$ est $I(\xi|A \in d_j)$ qui exprime le désordre de la sous-base selon les classes. $I(\xi|A \in d_j)$ est calculée selon une formule d'entropie choisie.

On évalue ensuite l'entropie conditionnelle de la base pour la discrétisation D, qui est une somme pondérée des $I(\xi|A \in d_j)$:

$$I(\xi|D) = \sum_{j=1}^{h} w_j I(\xi|A \in d_j)$$

où w_j est le poids associé à l'intervalle d_j et dépend souvent du nombre d'exemples dont la valeur pour A est dans l'intervalle d_j. Parmi l'ensemble des discrétisations possibles, celle qui maximise le gain d'information, défini comme la différence $I(\xi) - I(\xi|D)$, est choisie. La discrétisation d'un attribut revient donc à trouver $D_{meilleur}$:

$$D_{meilleur} = \arg\max_{D}(I(\xi) - I(\xi|D))$$

L'algorithme de discrétisation décrit ci-dessus tente de représenter un attribut numérique de manière la plus adéquate par rapport aux classes. Plus les valeurs symboliques obtenues sont cohérentes par rapport aux classes, meilleure est la discrétisation.

Le résultat de la discrétisation dépend donc de la mesure d'entropie utilisée [23]. Nous caractériserons dans la section suivante ces mesures comme des mesures utilisées dans la discrétisation.

Il est intéressant de remarquer que le problème de discrétisation est similaire, d'un certain point de vue, au problème de sélection du meilleur attribut. Considérons une discrétisation possible D de l'attribut A en h intervalles d_1, d_2, .., d_h. On crée un *pseudo-attribut* symbolique A_D issu de la discrétisation D, qui a h valeurs dont chacune correspond à un intervalle parmi d_1, d_2, .., d_h. Par abus de notation, ces valeurs sont désignées également par d_1, d_2, .., d_h. La discrétisation est maintenant représentée par le pseudo-attribut A_D.

L'algorithme de discrétisation effectue en effet une sélection du meilleur attribut parmi l'ensemble des *pseudo-attributs* possibles selon un critère s'appuyant sur la notion d'entropie. L'attribut choisi est celui qui maximise le gain d'information. Ainsi on revient donc au problème de sélection d'attributs. Cela justifie l'utilisation d'une mesure de discrimination pour la discrétisation.

Cependant, dans plusieurs cas, le nombre de *pseudo-attributs* (qui est le nombre de solutions de discrétisations possibles) peut être très grand. Cela rend difficile l'application directe du processus de sélection du meilleur attribut. La recherche d'une manière adéquate pour limiter l'espace des solutions avant de lancer l'algorithme de sélection du meilleur attribut est donc nécessaire.

Comme les mesures d'entropie cherchent à décomposer le domaine en intervalles dont chacun ne contient que des valeurs des exemples d'une seule classe, elles favorisent la discrétisation en un grand nombre d'intervalles. Le cas limite est que chaque intervalle ne contient qu'une seule valeur. Cela n'apporte aucune signification. Il faut donc des techniques pour limiter le biais de ce phénomène. La plupart des algorithmes font appel à des critères d'arrêt qui imposent un certain nombre de contraintes sur l'ensemble des valeurs numériques. Elles reposent généralement sur une mesure d'homogénéité des exemples correspondant à un intervalle : si ces exemples sont suffisamment homogènes, on ne découpe plus l'intervalle. Une autre mesure est de fixer le nombre maximal d'intervalles pour un attribut. La discrétisation en deux est largement utilisée. La discrétisation peut également s'arrêter quand les exemples dans un intervalle ne sont pas suffisamment nombreux. Un ensemble de critères d'arrêt se trouve dans [34].

Concernant l'entropie de Shannon, [64, 65] ont démontré que les coupures trouvées à l'aide de l'entropie de Shannon se trouvent toujours aux frontières entre deux classes. Cela conduit à une réduction significative de l'espace de recherche des coupures, et en conséquence à une réduction de temps de calcul. Les autres mesures d'information ne possèdent pas les mêmes propriétés. Considérons l'exemple d'une base de données qui se compose de 6 exemples qui se répartissent en deux classes C_1 et C_2 et qui sont décrits par un attribut A (tableau 2.2).

ξ	e_1	e_2	e_3	e_4	e_5	e_6
A	10	11	12	13	14	15
classe	C_2	C_1	C_2	C_2	C_1	C_2

Tab. 2.2 – Exemple des valeurs d'un attribut

Les calculs montrent qu'avec l'entropie de Rényi de Type 1 ($\beta = 20$) ou l'entropie de Rényi de Type 2 ($\beta = 3$) la coupure trouvée est 12.5. C'est la valeur qui sépare deux exemples e_3 et e_4 de classe C_2. La coupure trouvée avec l'entropie de Shannon est 13.5.

2.4.3 Évaluation de discrétisation et mesure d'équilibre

L'évaluation de la qualité d'une discrétisation s'appuie sur plusieurs critères comprenant à la fois des critères objectifs et des critères subjectifs :

1. L'homogénéité des classes des exemples dont les valeurs de l'attribut à discrétiser se trouvent dans un même intervalle (évalué par la mesure CAIM par exemple [91]) : il est préférable que ces exemples appartiennent à une seule classe.
2. Le nombre d'intervalles : un petit nombre d'intervalles est préféré. Cela permet de simplifier les données et en particulier de réduire la taille (le nombre de feuilles, la profondeur) de l'arbre obtenu.

3. La performance du modèle de classification obtenu : le résultat de la discrétisation sert ensuite au processus d'apprentissage. La performance du modèle de classification obtenu constitue ainsi un indice raisonnable de performance de la discrétisation. Cette approche est plutôt pragmatique
4. Critère subjectif : la discrétisation doit être adéquate, c'est-à-dire représenter et caractériser le mieux possible la nature des données. Elle doit conserver l'influence de l'attribut sur la classe de l'exemple. On évalue également l'interprétabilité et la sémantique de la discrétisation. Ce type de critères non objectifs est utilisé par certains auteurs [169].

En fonction des problèmes à résoudre, un ou quelques critères particuliers sont plus intéressants que d'autres. Dans certaines applications réelles, des arbres équilibrés sont préférés car des nombres similaires de tests sont nécessaires pour identifier la classe des exemples. Tandis que dans certains autres, une réponse rapide sur une classe particulière est préférée. En particulier dans les problèmes médicaux où la classe des maladies graves doit être identifiée aussi tôt que possible. Évidemment, la discrétisation influence la forme de l'arbre obtenu. Nous souhaitons alors étudier dans la suite le comportement des mesures d'entropie dans la discrétisation. Pour cela, nous nous proposons d'étudier l'équilibre d'une discrétisation binaire. Cette étude permet de considérer et de choisir la mesure qui favorise un type particulier d'arbres de décision donnés : équilibré, non-équilibré. Pour cela, une mesure d'équilibre est introduite.

Considérons le cas à deux classes. Nous proposons de définir la mesure d'équilibre E d'une discrétisation D comme suit :

$$E(D) = \max\left(\frac{n_1}{n_2}, \frac{n_2}{n_1}\right)$$

où n_1 et n_2 sont respectivement le nombre de valeurs dans l'intervalle 1 (à gauche) et l'intervalle 2 (à droite) du point de coupure.

On a : $E(D) \geq 1$. Si $n_1 \geq n_2$, E est une fonction croissante selon $\frac{n_1}{n_2}$. Quand $n_1+n_2 = n$ est fixé, E est une fonction croissante selon n_1-n_2. Plus une discrétisation a une mesure d'équilibre proche de 1, plus les nombres d'exemples dans les deux parties sont similaires. Il est probable que cela conduit à un arbre de décision équilibré au sens de la profondeur. En général, une discrétisation ayant une faible mesure d'équilibre est préférée.

Une expérimentation, qui a été menée avec cette mesure d'équilibre, est présentée dans la section suivante.

2.4.4 Expérimentations

L'expérimentation menée dans cette section a pour but d'illustrer l'utilisation de la mesure d'équilibre d'une discrétisation binaire présentée ci-dessus et de montrer comment différentes mesures de discrimination se comportent dans un processus de discrétisation. Les mesures considérées sont les entropies conditionnelles de Rényi,

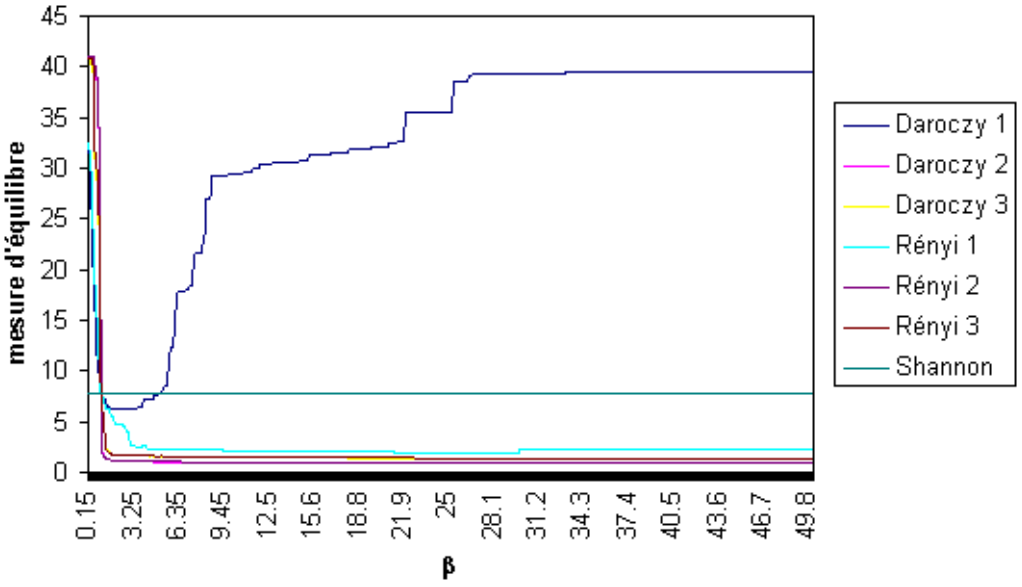

FIG. 2.10 – Mesures d'équilibre moyennes des partitions avec les entropies conditionnelles en grande échelle, $\beta \in (0, 50]$

celles de Daróczy et l'entropie conditionnelle de Shannon. La base de données « Waveform » [27] a été utilisée. Elle se compose de 3 classes, chacune contenant 100 exemples. Chaque exemple est décrit par 21 attributs numériques. À partir de la base initiale, à chaque fois, une classe est éliminée pour obtenir 3 bases, chacune ne contenant que 200 exemples, répartis en seulement 2 classes. Les attributs de chacune des 3 bases sont discrétisés. Au total, 63 discrétisations ont été réalisées (3 bases, 21 attributs chacune).

A chaque fois, les entropies conditionnelles de Rényi, celles de Daróczy et celle de Shannon avec différentes valeurs du coefficient β sont utilisées pour identifier le point de coupure de l'attribut numérique en question. Une fois le point de coupure trouvé, la mesure d'équilibre de la discrétisation est calculée. Enfin les statistiques telles que la valeur moyenne, la valeur maximale, la valeur minimale, et la valeur médiane des mesures d'équilibre selon chacune des entropies conditionnelles sont moyennées sur l'ensemble des discrétisations.

Les figures 2.10 et 2.11 présentent les résultats obtenus. Les courbes avec β variant de 0 à 50 sont dessinées dans la figure 2.10. Dans cette figure, la ligne horizontale d'ordonnée 7.85 correspond à l'entropie conditionnelle de Shannon qui ne dépend pas de β. Pour faciliter la visualisation, la figure 2.11 présente le même résultat mais sur une petite échelle de l'ordonnée. Les courbes correspondant à l'entropie conditionnelle de Shannon et celle de Daróczy dépassent le cadre de cette figure.

A l'exception de l'entropie conditionnelle de Daróczy de Type 1, les mesures d'équilibre correspondant à chaque entropie conditionnelle sont décroissantes selon

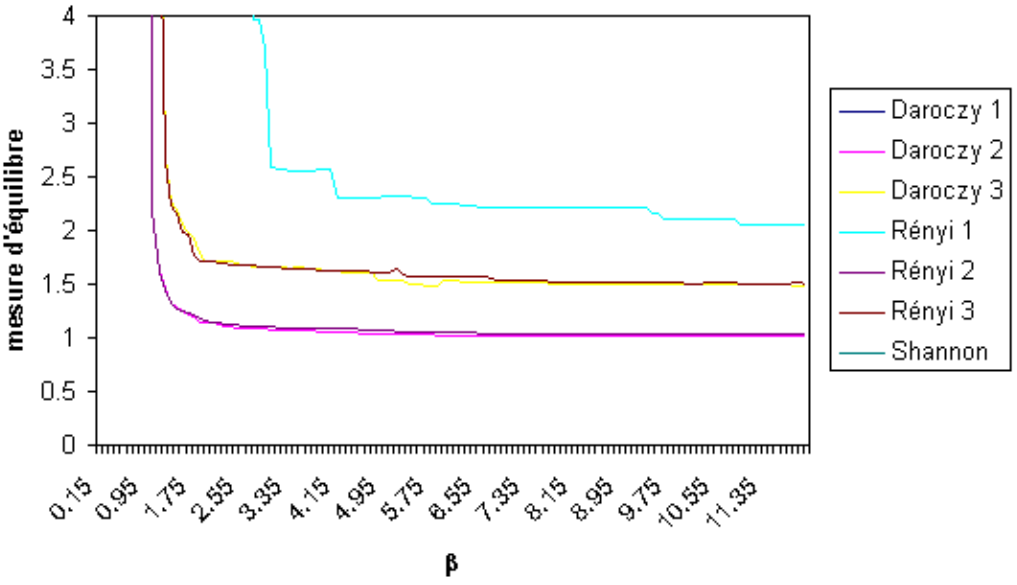

FIG. 2.11 – Mesures d'équilibre moyennes des partitions avec les entropies conditionnelles en petite échelle, $\beta \in (0, 12]$

β. La décroissance est très rapide lorsque β est proche de 0. Lors du passage à la valeur 1 de β, il y a un changement très fort pour les mesures de Type 2 : la mesure d'équilibre passe de 28 pour $\beta = 0.95$ à 2.2 pour $\beta = 1.05$. Pour le Type 2 et de Type 3, les mesures de même type ont un comportement très similaire. Avec les entropies de Type 2, lorsque $\beta > 2.5$, les partitions tendent vers un équilibre parfait (le degré d'équilibre tend vers 1) où le point de coupure découpe le domaine d'attribut en deux intervalles dont chacun contient le même nombre de valeurs. Tandis qu'avec les mêmes valeurs de β, les partitions issues des entropies conditionnelles de Type 3 ont une mesure d'équilibre proche de 1.5.

Lorsque β s'approche de 0, les partitions sont très déséquilibrées. Les points de coupure se trouvent souvent proches des extrémités du domaine d'attributs, ainsi seuls quelques exemples à une extrémité sont regroupés. La stratégie est plutôt d'obtenir un petit intervalle à l'extrémité du domaine qui contient des valeurs d'attributs des exemples d'une même classe. Ces remarques s'accordent bien avec le résultat décrit dans la figure 2.4, où les mesures entropies conditionnelles correspondant à la petite valeur de β entraînent la création rapide de la première feuille.

Le comportement des entropies de Type 2 peut être interprété par leur pondération :

$$I(\xi|D) = P^\beta(A \in d_1) I_\beta(\xi|A \in d_1) + P^\beta(A \in d_2) I_\beta(\xi|A \in d_2)$$

où $P^\beta(A \in d_1) + P^\beta(A \in d_2) = 1$. Pour minimiser cette quantité, quand β est assez grand, il favorise $P^\beta(A \in d_1) \simeq P^\beta(A \in d_2)$ et quand β est proche de 0, il favorise $\min(P^\beta(A \in d_1), P^\beta(A \in d_2)) = 0$.

Parmi les entropies conditionnelles de Type 1, l'entropie conditionnelle de Daróczy engendre des partitions très déséquilibrées, tandis que l'entropie conditionnelle de Rényi semble engendrer des partitions plus équilibrées. La partition la plus équilibrée avec l'entropie conditionnelle de Daróczy de Type 1 correspond à des valeurs de β aux alentours de 1. Au delà de cet intervalle, la mesure augmente assez rapidement. Cela est cohérent avec la figure 2.4, où la grande valeur de β pour l'entropie conditionnelle de Daróczy de Type 1 entraîne des arbres dont la profondeur minimale est petite.

2.4.5 Discussion

L'utilisation des différentes mesures de discrimination dans la discrétisation des attributs numériques est justifiée. Une discrétisation est caractérisée par sa capacité de discrimination des classes. Nous avons montré que, d'un certain point de vue, un problème de discrétisation peut être considéré comme équivalent à un problème de sélection du meilleur attribut. Ainsi, les mesures de discrimination sont capables de servir à un processus de discrétisation.

L'expérimentation menée dans cette section est encore très réduite. Pourtant, elle permet de relever quelques comportements des mesures utilisées. Pour un résultat plus fiable, nous envisageons d'élargir l'expérimentation sur plusieurs attributs numériques. L'étude du rapport entre le résultat de discrétisation et la forme d'arbre doit être approfondie. Cela permet de comprendre l'influence de la phase de discrétisation sur l'induction d'arbres. Par exemple, quelles sont les mesures qui rendent les arbres plus petits ? Quelles sont les mesures qui entraînent des arbres ayant un taux de bonnes classifications élevé ? Pour cela, nous envisageons de caractériser une discrétisation autrement que par l'équilibre.

2.5 Utilisation des sous-ensembles flous dans la construction des arbres

2.5.1 Introduction

Depuis une quinzaine d'années, les techniques issues de la théorie des sous-ensembles flous [163] ont été considérées pour la construction des arbres de décision flous. D'une part, cela répond à un besoin de manipuler des données floues, qui viennent de différents domaines, notamment ceux qui concernent des processus cognitifs. Une valeur floue d'un attribut est représentée par un sous-ensemble flou, l'appartenance d'un exemple à une classe n'est pas absolue, la séparation entre les classes n'est pas nette. Ce genre de données apparaît de plus en plus. D'autre part, l'introduction de ces techniques permet également de renforcer certaines propriétés des méthodes classiques même sur les données conventionnelles : traitement plus souple des données, réduction de la sensibilité au bruit et augmentation de la précision de classification. Comme l'esprit humain manipule plus aisément des concepts

imprécis, un arbre de décision flou est souvent plus compréhensible, notamment quand les sous-ensembles flous sont décrits par des expressions linguistiques. Par exemple, dans un contexte précis, les prix d'un même appareil photo numérique dans 2 magasins différents sont de 329.90 euros et 331 euros. Ils sont différents mais il n'est pas très juste de dire que le prix de 331 euros est élevé et que l'autre n'est pas élevé. Cette situation peut éventuellement arriver si une coupure précise est utilisée et si elle se situe entre les deux prix. Les données méritent donc d'être manipulées de manière plus souple. Si on les traite comme des valeurs symboliques ou comme des valeurs numériques avec une coupure précise, ces deux valeurs conduisent les exemples dans des feuilles différentes.

Pour remédier à ces problèmes, des fonctions d'appartenance sont introduites dans les arbres de décisions flous pour caractériser l'appartenance d'un exemple à un nœud ou celle d'une valeur précise à une modalité floue associée à un arc. Dans certains cas, même les exemples à l'entrée de l'algorithme sont associés à des degrés d'appartenance. Pendant le partitionnement de la base d'apprentissage, un exemple peut être dirigé vers plusieurs nœuds. Il appartient ainsi à plusieurs sous-ensembles flous associés à ces nœuds avec certains degrés d'appartenance. Dans chaque feuille, la présence de plusieurs classes est possible et donc chaque feuille peut être étiquetée par plus d'une classe. L'identification des fonctions d'appartenance devient donc un problème majeur.

En considérant un arbre comme une représentation d'un ensemble de règles de classification, dans le cas classique, lors de la classification, une seule règle est appliquée pour identifier la classe d'un exemple. Cependant, il est possible que plusieurs règles floues dans l'ensemble des règles correspondant à un arbre de décision flou soient activées afin de classer un exemple. Un exemple arrive ainsi à plusieurs feuilles. Les degrés d'appartenance sont déterminés par plusieurs méthodes à partir du chemin que l'exemple parcourt de la racine vers les feuilles. Une agrégation et éventuellement une étape de défuzzification sont donc indispensables pour aboutir à un résultat final exploitable. Cette agrégation doit tenir compte à la fois des degrés d'appartenance de l'exemple aux feuilles et des répartitions des exemples dans chaque feuille. Par défaut, la classe à laquelle l'exemple appartient avec le degré d'appartenance le plus grand lui est affectée.

En référence au schéma de construction d'arbres TDIDT présenté au début de ce chapitre (section 2.2.2), en présence de données floues ou de données fuzzifiées, des techniques dédiées à leur traitement peuvent intervenir dans les étapes :

1. Sélection du meilleur attribut
2. Discrétisation des attributs numériques : une coupure floue peut être utilisée au lieu d'une coupure précise. Cette étape est souvent importante pour la conception des fonctions d'appartenance.
3. Partitionnement récursif de la base associée au nœud courant selon les valeurs floues de l'attribut choisi. Un exemple peut appartenir à plusieurs ensembles avec un certain degré d'appartenance.
4. Vérification des critères d'arrêt

Toutefois, on n'exclut pas l'existence des variantes de schéma TDIDT ainsi que des schémas complètement différents. Dans les autres schémas, les techniques floues peuvent également intervenir. Nous nous limitons dans cette thèse à l'étude du schéma TDIDT.

Dans l'état de l'art, plusieurs méthodes ont été proposées pour construire des arbres de décision flous. Nous recensons dans ce chapitre les méthodes principales, avec l'accent sur les méthodes récentes, et proposons une taxonomie des méthodes. Cette taxonomie vise à caractériser les méthodes en s'appuyant sur la stratégie de partitionnement récursif de la base d'apprentissage et la stratégie de conception de fonctions d'appartenance. Nous introduisons ensuite l'utilisation des mesures de discrimination floues, telles que les entropies conditionnelles de Rényi, celles de Daróczy décrites dans le chapitre 1 dans la construction des arbres de décision flous.

2.5.2 Typologie

Dans quelques travaux, les méthodes de construction d'arbres de décision flous sont caractérisées. Marsala [102] a insisté sur le rôle primordial des mesures utilisées pour la sélection de l'attribut le plus discriminant. Ainsi, les méthodes de construction d'arbres en présence de données floues sont classées selon la mesure utilisée, notamment l'entropie floue de Shannon ou une autre mesure.

L'entropie floue a été présentée dans le chapitre 1 (formule (1.18), page 40). Parmi les principales méthodes utilisant cette entropie, on peut citer les travaux de Ramdani [129, 130] et ceux de Weber [158] avec la méthode nommée Fuzzy ID3 sur laquelle de nombreux autres travaux sont d'ailleurs basés, entre autres, ceux de Umano *et al.* [150], Janikow avec le système FID [78], etc. La différence entre ces méthodes se situe dans l'optimisation des partitions floues ou dans l'utilisation de l'arbre pendant la classification [102].

La méthode de Cios et Sztandera [37] s'appuie sur des mesures floues comme la mesure floue de De Luca et Termini ou l'entropie floue de Kosko. Pour la catégorie de méthodes sans utilisation de l'entropie de Shannon, on peut citer, entre autres, les travaux de Bothorel [17] et de Boyen [24]. Bothorel a utilisé une mesure de contraste entre deux sous-ensembles flous. La mesure de Kolmogorov-Smirov a été utilisée par Boyen pour leurs applications dans le domaine de sécurité des réseaux électriques.

Dans une autre optique, Chiang [36] divise également les méthodes de construction des règles floues par arbres de décision en deux mais selon un autre critère dépendant du moment où la fuzzification est effectuée. Si la fuzzification s'effectue avant ou pendant la construction (*pre-fuzzification*), il faut tenir compte des attributs flous pour la sélection du meilleur attribut et pour le partitionnement de la base d'apprentissage. La plupart des méthodes appartiennent à cette catégorie : la méthode de Weber [158], celle de Janikow [78], la méthode SAFI de Ramdani [130], la méthode de Umano *et al.* [150], Yuan et Shaw [162], etc. Certains [154, 79] supposent que les données à traiter sont floues ou qu'elles sont fuzzifiées a priori par un humain. D'autres utilisent une méthode automatique de génération des fonctions d'appartenance, par exemple par une discrétisation automatique des données numé-

riques. La discrétisation peut se faire en une seule fois [158] avant la construction des arbres ou en plusieurs fois, à la volée, pendant la construction de l'arbre [117]. Cette dernière méthode semble plus judicieuse car elle peut prendre en compte le contexte actuel du processus d'induction, par exemple la valeur d'autres attributs.

À l'opposé, certains construisent un arbre classique, puis cherchent à générer des règles floues à partir de l'arbre classique (*post-fuzzification*). L'utilisation de règles floues dans ce cas permet de généraliser les règles et renforcer leur robustesse. Cios et Sztandera [37] ont proposé de représenter la frontière floue entre des valeurs numériques par un neurone artificiel ayant une fonction d'activation sigmoïde. Un réseau de neurones est également utilisé dans le travaux de [24]. De leur côté, Crockett *et al.* [41] utilisent un algorithme génétique pour construire des règles floues à partir d'un arbre de décision classique construit par C4.5. Nous nous intéressons peu à cette catégorie de méthodes car nous souhaitons obtenir tout de suite, après l'induction, un arbre de décision flou.

Dans la suite, nous proposons une nouvelle taxonomie des méthodes qui s'inspire et généralise des taxonomies décrites ci-dessus. Notre taxonomie se fonde sur deux critères principaux :

1. La méthode de sélection du meilleur attribut
2. La stratégie d'identification des fonctions d'appartenance.

Notons également que, à part ces deux critères que nous jugeons les plus importants ci-dessus les méthodes de construction d'arbres de décision flou peuvent éventuellement être caractérisées par les propriétés suivantes :

1. Propriétés des données que la méthode est capable de les traiter (précises, imprécises et/ou incertaines, numériques, symboliques, floues)
2. Méthode d'élagage utilisée
3. Mécanisme d'inférence
4. Incrémentalité
5. Propriétés des arbres produits (binaires, forêt d'arbres, etc)

Nous décrivons par la suite ces deux critères principaux.

2.5.2.1 Premier critère : méthode de sélection du meilleur attribut

C'est un critère similaire à celui de [104]. Il y a plusieurs méthodes de sélection de l'attribut. Plusieurs d'entre elles se basent sur la méthode ID3 classique en utilisant l'entropie floue au lieu de l'entropie de Shannon. À côté des méthodes utilisant l'entropie floue citées ci-dessous, on peut citer encore des travaux plus récents [165, 79, 81, 94, 75, 117, 36, 85, 149, 41]. Nous analysons ci-dessous plutôt des méthodes publiées depuis une dizaine années.

Janikow et ses co-auteurs développent et maintiennent un système, nommé FID, depuis 1996 [79]. Dans la première version, les attributs sont discrétisés manuellement et leurs valeurs sont représentées par des sous-ensembles flous. Plusieurs améliorations ont été ajoutées dans le système initial afin de renforcer sa qualité : des

méthodes de discrétisation automatique des données numériques, le traitement des attributs ayant des valeurs manquantes, des méthodes d'élagage, etc [81]. FID peut traiter des attributs ayant à la fois des valeurs numériques et des valeurs symboliques. L'utilisation des forêts d'arbres est également introduite [80]. Ici, l'utilisation d'une forêt signifie que des tests alternatifs à chaque nœud d'un arbre sont possibles. Donc, au lieu de choisir un seul attribut, ils en choisissent plusieurs. L'arbre est ensuite développé pour toutes les valeurs des attributs choisis. L'arbre obtenu est en fait un arbre à 3 dimensions. Entre autres, l'introduction des tests alternatifs permet de mieux s'adapter à des valeurs manquantes de l'exemple à classer. Si une valeur d'un attribut est manquante, on peut éventuellement utiliser la valeur d'un autre attribut du même exemple. Le principe utilisé dans FID est repris dans plusieurs travaux ultérieurs.

Lee *et al.* [94] utilisent également la mesure d'entropie floue et proposent d'associer à chaque arc plusieurs valeurs symboliques d'un attribut au lieu d'une seule valeur. Pour cela, le concept de *classwise element set* est introduit dans le partitionnement de l'espace selon un attribut symbolique. Dans leur méthode, les données numériques sont représentées par des nombres flous, des intervalles flous ou des valeurs précises. Les classes sont nominales et associées à un degré de confiance.

De leur côté, après avoir analysé la complexité du problème d'optimisation d'arbres de décision flous, Wang *et al.* [155] proposent un processus de fusion des branches afin de réduire la taille des arbres. À chaque nœud, la fusion des sous-ensembles flous correspondant à des valeurs différentes du meilleur attribut est effectuée à la suite de sa sélection. Seuls les sous-ensembles ayant des propriétés spécifiques communes sont choisis pour fusionner. Ainsi, un arc de l'arbre peut être étiqueté par plusieurs valeurs de l'attribut choisi et la taille de l'arbre obtenu devient plus faible.

Récemment, Bartczuk et Rutkowska [12] présentent une nouvelle version de l'algorithme Fuzzy-ID3. La différence essentielle par rapport à la version originale est que plusieurs attributs et leurs valeurs peuvent être affectés à une seule feuille. L'idée est de retenir, à chaque nœud, les valeurs des attributs qui entraînent directement la classe des exemples et de créer des feuilles avec les exemples correspondants. Ensuite, les exemples restants sont partitionnés selon un attribut sélectionné à l'aide d'une mesure d'entropie. La prise en compte de plusieurs attributs et leurs valeurs dans une feuille permet de réduire significativement la taille de l'arbre.

La méthode proposée par Pedrycz et Sosnowski [116] fait partie d'une approche différente. Il s'agit une approche hybride qui combine la stratégie top-down et la stratégie bottom-up. Un regroupement flou (*fuzzy clustering*) des exemples selon la stratégie bottom-up est effectué avant chaque partitionnement de la base. Cela permet de mieux exploiter la similarité entre des exemples. Le partitionnement s'effectue sur les partitions trouvées par l'algorithme de regroupement. Le partitionnement n'est donc pas linéaire et il se base sur plusieurs attributs. La frontière est tracée de manière floue dans l'espace des exemples.

Tsang *et al.* [149] proposent une présélection des attributs flous avant la construction des arbres de décision flous. Leur méthode de sélection d'un sous-ensemble d'attributs s'intitule OFFSS (*Optimal Fuzzy-valued Feature Subset Selection*). Ainsi, seul

un sous-ensemble d'attributs est considéré par l'algorithme de construction d'arbres. Cela conduit non seulement à la réduction de temps de calcul dans la phase de construction mais aussi à la simplicité de l'arbre et à l'augmentation de la précision de la classification en ne prenant en compte que des attributs intéressants.

Une version normalisée de mesure d'information a été proposée par Wang et Borgelt dans [154, 156]. Ils ont remarqué que le gain d'information apparaissant dans l'algorithme Fuzzy ID3 peut éventuellement être négatif. Ce n'est pas le cas dans la méthode ID3 classique. Ce phénomène est la conséquence de deux faits. Le premier est que la somme des degrés d'appartenance d'une valeur numérique à tous les sous-ensembles flous correspondant à l'attribut n'est pas toujours 1, sachant qu'il y a des chevauchements entre des sous-ensembles flous. Le second vient de l'utilisation des opérateurs de type t-norme et t-conorme dans la définition de la probabilité et de la probabilité conditionnelle floue. Cette valeur négative engendre des problèmes techniques dans la sélection du meilleur attribut. Pour remédier à ce problème, au lieu de choisir les opérateurs de type t-norme et t-conorme adéquats et d'imposer des contraintes spécifiques dans la construction des sous-ensembles flous comme nous l'avons fait (cf. section 1.4.2, page 35 et également [47, 44]), ils proposent de normaliser les probabilités floues pour que leur somme soit 1 :

$$\sum_{i=1}^{n} P^*(C_i) = 1 \quad \text{et} \quad \sum_{i=1}^{n} P^*(C_i|v_j) = 1$$

où v_j est la valeur d'un attribut.

Wang *et al.* décrivent dans leur article [157] une heuristique de recherche de l'attribut le plus important pour déterminer la classe d'un exemple. Le degré d'importance d'un attribut est défini par le nombre d'exemples incohérents créés lors de la suppression de l'attribut en question. Un exemple est considéré incohérent par rapport à un autre exemple si les deux coïncident sur tous les attributs sauf sur l'attribut de classe. Cette mesure est généralisée pour mesurer le degré d'importance d'un attribut flou. L'attribut ayant le plus grand degré est sélectionné pour partitionner la base d'apprentissage. Une étude comparative assez intéressante entre la méthode Fuzzy ID3, celle de Yuan et Shaw [162] et cette méthode de Wang *et al.* est également présentée. La conclusion sur les différents aspects est [3] :

1. Applicabilité : Fuzzy ID3 $\geq$ Yuan et Shaw, et Wang
2. Complexité : Fuzzy ID3 $\leq$ Yuan et Shaw $\leq$ Wang
3. Interprétabilité : Yuan et Shaw $\leq$ Wang $\leq$ Fuzzy ID3
4. Précision : Wang $\leq$ Fuzzy ID3 $\leq$ Yuan et Shaw
5. Traitement d'ambiguïté de classification : Yuan et Shaw $\geq$ Fuzzy ID3 et Wang
6. Robustesse : Wang $\leq$ Fuzzy ID3, et Yuan et Shaw.

La méthode *Soft Decision Trees (SDT)* proposée par Olaru et Wehenkel [115], cherche à minimiser une fonction d'erreur afin de sélectionner un attribut et de

[3] Pour alléger l'écriture, on note ici « $X \geq Y$ » pour « l'algorithme X est meilleur que l'algorithme Y selon cet aspect » .

déterminer des paramètres des fonctions d'appartenance. Les feuilles ne sont pas associées à une classe discrète mais à une valeur numérique comme pour les arbres de régression. Les arbres obtenus par cette méthode sont binaires. Chaque arbre caractérise l'appartenance d'un exemple à une classe. Dans le cas de problème à plus de deux classes, plusieurs arbres sont utilisés : chacun s'occupe de distinguer une classe contre l'union des toutes les autres. Dans leur méthode, le domaine d'un attribut est discrétisé en deux intervalles par une coupure floue qui est caractérisée par un seuil précis α et un étalement β. Leur algorithme cherche un attribut dont la discrétisation permet de minimiser l'erreur carrée :

$$\text{erreur}_\xi = \sum_{e \in \xi} \mu_\xi(e)[\mu_c(e) - \hat{\mu_c}(e)]^2$$

où ξ est l'ensemble flou des exemples associés à un nœud, $\mu_c(e)$ est le vrai degré d'appartenance de e à c, et $\hat{\mu_c}(e)$ est le degré d'appartenance de l'exemple e à la classe c prévu par l'arbre à ce nœud.

Pour la construction des arbres flous binaires similaires à des arbres étudiés par Olaru et Wehenkel, Boyen et Wehenkel [24] ont proposé l'utilisation d'une version normalisée de la mesure de Kolmogorov-Smirnov.

Parmi les méthodes qui n'utilisent pas l'entropie floue, la plus citée est la méthode de Yuan et Shaw [162]. Cette méthode utilise comme mesure de discrimination une mesure d'ambiguïté de classification. Nous avons examiné cette mesure dans le chapitre précédent (cf. section 1.5.4, page 43). Ha et Zhang [70] proposent une approche différente pour calculer la mesure d'ambiguïté de classification. Dans les travaux de Yuan et Shaw, la mesure d'ambiguïté de classification d'un attribut est la moyenne pondérée de chacune des valeurs de l'attribut. Selon Ha et Zhang la mesure d'ambiguïté de classification d'un attribut est obtenue en calculant l'ambiguïté d'une distribution de possibilité normalisée issue de l'attribut. Il est à noter que l'équivalence de ces méthodes de calcul a été montrée dans [105].

2.5.2.2 Second critère : stratégie d'identification de fonctions d'appartenance

Les fonctions d'appartenance ont un impact important durant l'induction et l'utilisation des arbres de décision. Il s'agit généralement de caractériser comment une valeur appartient à un ensemble flou défini sur son domaine. Cet ensemble définit une modalité. Pour un attribut numérique discrétisé, il s'agit d'un intervalle flou. Ensuite, il faut caractériser comment un exemple appartient à l'ensemble flou des exemples associés à une feuille. Lors de la classification, les fonctions d'appartenance participent à la décision sur l'influence de chaque règle sur le résultat agrégé.

La stratégie d'identification des fonctions d'appartenance s'appuie sur deux points principaux :

1. Quand détermine-t-on les fonctions d'appartenance (avant, pendant ou après la construction de l'arbre) ?
2. Comment détermine-t-on les fonctions d'appartenance ?

La première question est celle posée dans la travail de Chiang [36] que nous avons évoqué au début de la section 2.5.2. Nous analysons la seconde.

Certaines méthodes supposent que des données floues sont disponibles a priori [79]. Après avoir sélectionné un attribut à l'aide d'une mesure d'entropie floue, les exemples sont partitionnés selon les valeurs de l'attribut. Supposons que le degré d'appartenance d'un exemple e à l'ensemble des exemples $\mathcal{X}$ correspondant au nœud $\mathcal{N}$ en question soit $\mu_{\mathcal{X}}(e)$. À ce nœud, l'attribut flou choisi est A. Le fils $\mathcal{N}_j$ de $\mathcal{N}$, correspondant à la valeur floue v_j de A, est associé à un sous-ensemble flou d'exemples $\mathcal{X}_j$ caractérisé par une fonction d'appartenance. Cette dernière est déterminée par la combinaison entre l'appartenance de e à $\mathcal{X}$ et la satisfaction de la valeur $e(A)$ de l'exemple e pour A relativement à v_j :

$$\mu_{\mathcal{X}_j}(e) = T(s_{v_j}(e(A)), \mu_{\mathcal{X}}(e))$$

où $s_{v_j}(e(A))$ est le degré de satisfaction de $e(A)$ relativement à v_j et T est une t-norme. Dans la pratique, le *produit* est couramment utilisé. Si $e(A)$ est une valeur précise, $s_{v_j}(e(A))$ devient $\mu_{v_j}(e(A))$.

L'existence de données floues n'est pas toujours le cas et on a également besoin de construire des arbres de décision flous à partir de données précises, notamment des données numériques. En général, l'identification des sous-ensembles flous réside dans le processus de discrétisation floue. Au lieu d'affecter une valeur à un seul intervalle comme dans le cas de la discrétisation non floue, une valeur appartient à un intervalle avec un certain degré d'appartenance et elle peut appartenir à plusieurs intervalles. Cela permet par la suite une manipulation plus souple des valeurs numériques, et ainsi éviter un basculement brutal, surtout pour des valeurs proches de la coupure. Plusieurs travaux montrent que la discrétisation floue renforce les performances du système [102], en particulier sa robustesse.

Les fonctions d'appartenance peuvent être déterminées manuellement par les utilisateurs ou par des experts [154, 130] à partir des données numériques. On cherche alors à découper le domaine de l'attribut selon la sémantique de l'attribut. Par exemple, le domaine des notes d'un examen peut être découpé en des intervalles flous correspondant à des mentions : excellent, très bien, bien, assez bien, passable, échec. Comme dans le cas classique, la discrétisation floue manuelle peut se faire de manière interactive : l'utilisateur est chargé de déterminer la zone floue à chaque étape, avant la sélection du meilleur attribut. Cependant, la discrétisation manuelle ne fonctionne plus quand on possède une grande quantité de données ou quand la sémantique des données est difficile à comprendre. Par d'ailleurs, la dépendance à un être humain entraîne des problèmes humains : erreur ou incohérence des experts, conflits entre des opinions. La discrétisation automatique est donc souvent nécessaire.

Parmi les méthodes de discrétisation automatique, une famille de méthodes usuelles consiste à appliquer d'abord un processus traditionnel pour trouver des coupures précises, puis assouplir des points de coupure en créant un étalement (zone floue) autour de chacun des points de coupure précis. Un état de l'art des méthodes de discrétisation classiques est établie dans [98]. L'utilisation de cette zone d'étalement est justifiée par le manque des connaissances lorsqu'on échantillonne un certain

nombre d'exemples pour la base d'apprentissage. Plus l'attribut est vague, plus l'étalement doit être important. Différentes méthodes ont été proposées pour déterminer cette zone [77] :

1. utilisation de la valeur maximale inférieure et de la valeur minimale supérieure à la coupure pour définir la zone floue [165]. Une variante de cette méthode est que l'on détermine l'étalement en fonction de la distance entre ces deux valeurs.

2. utilisation d'un intervalle de longueur fixée autour de la coupure, par exemple 10% de la longueur du domaine de l'attribut.

Dans tous les cas, la longueur de l'étalement est souvent choisie de manière expérimentale. Si la zone floue est trop faible, l'arbre pourra être identique à celui obtenu par la méthode classique correspondante car, pendant le partitionnement, aucun exemple de la base d'apprentissage ne se trouvera dans cette zone sensible. En conséquence, chaque exemple n'appartient qu'à une seule partition et le degré d'appartenance est 1. Cependant, chaque arc de l'arbre flou est associé à une fonction d'appartenance et finalement le résultat en classification peut être différent pour de nouveaux exemples.

Les fonctions d'appartenance correspondant à la coupure floue sont souvent linéaires par morceaux. La plupart entre d'elles sont trapézoïdales ou triangulaires, voir par exemple [165, 117, 115]. Ces formes sont largement utilisées pour des raisons de simplicité. Cependant des fonctions plus complexes peuvent être considérées, par exemple la forme gaussienne [35].

Olaru et Wehenkel [115] identifient d'abord un attribut et sa coupure précise (d'étalement nul) par la fonction d'erreur. Dans une second étape, l'étalement est optimisé.

Dans les travaux de Peng et Flash [117], l'entropie floue intervient dans la sélection de coupure floue à partir de données numériques précises. Les coupures sont positionnées autour des frontières des classes différentes. La coupure retenue est celle qui rend les plus homogènes les ensembles des exemples dans les deux intervalles qu'elle sépare. Dans le cas de la discrétisation binaire, qui est d'ailleurs très utilisée en raison de sa simplicité, un seul point de coupure est nécessaire à chaque étape. Une discrétisation en plus de deux intervalles suit le même principe.

Considérons un ensemble flou ξ d'exemples ayant un attribut A à discrétiser en deux intervalles flous v_1 et v_2 (figure 2.12). Supposons qu'une coupure floue soit caractérisée par un point de coupure c et un étalement δ. La forme de la fonction d'appartenance est trapézoïdale.

Les fonctions d'appartenance d'une valeur x à v_1 et à v_2 sont définies comme suit :

$$\mu_{v_1}(x) = \begin{cases} 1 & \text{si } x \leq c - \delta \\ \frac{c+\delta-x}{2\delta} & \text{si } c - \delta < x < c + \delta \\ 0 & \text{si } x \geq c + \delta \end{cases}$$

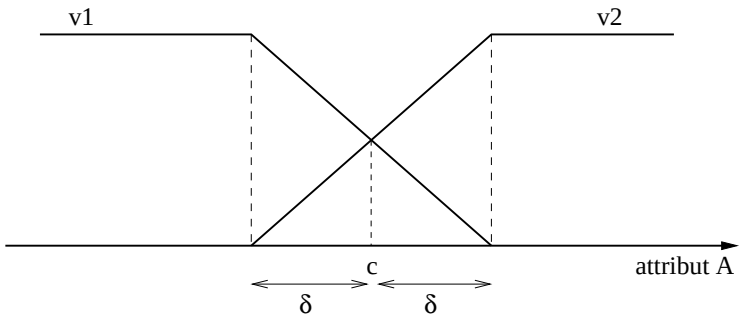

FIG. 2.12 – Coupure floue

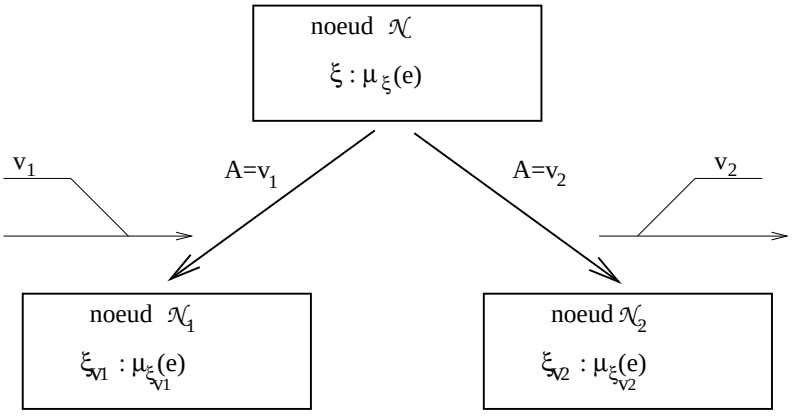

FIG. 2.13 – Degré d'appartenance à un nœud

et

$$\mu_{v_2}(x) = \begin{cases} 0 & \text{si } x \leq c - \delta \\ \frac{x-c+\delta}{2\delta} & \text{si } c - \delta < x < c + \delta \\ 1 & \text{si } x \geq c + \delta \end{cases}$$

On a :

$$\forall x \in \mathbb{R} \ : \ \mu_{v_1}(x) + \mu_{v_2}(x) = 1$$

Ces deux valeurs floues constituent une partition binaire floue du domaine de l'attribut. Il s'agit d'une partition normale. La fonction d'appartenance de ξ_{v_j} ($i = 1$ ou $i = 2$) est définie par :

$$\mu_{\xi_{v_j}}(e) = \mu_\xi(e).\mu_{v_j}(e(A))$$

où $e(A)$ est la valeur pour A de e.

Supposons que $I^*(\xi_{v_j})$ soit l'entropie floue de l'ensemble des exemples issus de la modalité v_j.

Après le découpage, l'entropie floue de ξ conditionnée par le découpage est :

$$\frac{|\xi_{v_1}|}{|\xi|} I^*(\xi_{v_1}) + \frac{|\xi_{v_2}|}{|\xi|} I^*(\xi_{v_2})$$

La coupure qui minimise cette entropie est sélectionnée. Plus l'attribut est incertain, plus l'étalement flou doit être important.

Une fois l'attribut choisi, la base ξ est partitionnée en ξ_{v_1} et ξ_{v_2}.

De leur côté, Lee *et al.* [94] proposent de construire des partitions floues de forme trapézoïdale pendant l'induction des arbres en utilisant une mesure de gain d'information. Avant de choisir une coupure floue, les valeurs numériques sont décrites sous la forme de nombres flous, d'intervalles flous ou simplement d'une valeur précise. Les extrémités du support de l'ensemble flou à créer et les extrémités de son noyau sont déterminées à partir des extrémités de supports de valeurs existantes d'attributs flous. Il doit évaluer toutes les partitions possibles pour trouver celle qui maximise le gain d'information.

L'algorithme *Fuzzy c-means* peut intervenir pour construire des sous-ensembles flous sur le domaine d'un attribut numérique [36]. Ainsi, on n'utilise pas d'information de classe dans la construction des sous-ensembles flous. Pedrycz [116] utilise également une technique de regroupement flou pour identifier les fonctions d'appartenance. Cependant, le regroupement est effectué dans l'espace des exemples et non dans l'espace des valeurs numériques d'un attribut. Par conséquent, dans la méthode de Pedrycz, aucun choix d'attributs n'est demandé et le partitionnement s'appuie sur plusieurs attributs.

Dans [103], une méthode originale pour déterminer des sous-ensembles flous a été développée en s'inspirant de la morphologie mathématique. L'auteur cherche à déterminer les noyaux homogènes pour l'attribut en question et ensuite à étendre ces noyaux pour définir des sous-ensembles flous. Cette technique est automatique, simple à mettre en œuvre, elle tient compte de la répartition des classes et est proche du comportement humain sur la même tâche. Elle est capable de tenir compte de la présence de classes floues.

Dans quelques travaux récents [85, 142, 35], des algorithmes génétiques sont employés pour optimiser les fonctions d'appartenance. Ils procèdent de la façon suivante :

1. produire une première population de fonctions d'appartenance. Ces fonctions sont trapézoïdales dans [85] ou gaussiennes dans [35].
2. construire des arbres de décision flous en utilisant les fonctions d'appartenance. L'algorithme FID [79] est utilisé. Cependant, un autre algorithme d'induction d'arbres flous peut intervenir.
3. évaluer l'ensemble des fonctions d'appartenance par la qualité de l'arbre de décision flou obtenu avec ces fonctions.
4. si la condition d'arrêt est satisfaite alors s'arrêter
5. sinon, générer une nouvelle population de fonctions d'appartenance en appliquant les mutations génétiques, et aller à (2).

Ainsi, les algorithmes génétiques sont utilisés pour produire un ensemble de fonctions d'appartenance appropriées au sens où elles permettent de produire un meilleur arbre de décision flou.

Toujours basé sur un algorithme génétique, Crockett *et al.* [41] construisent d'abord un arbre de décision classique par l'algorithme C4.5 puis cherchent à rendre floues les frontières précises présentes dans les nœuds. Il s'agit ici une approche de post-fuzzification. Pour justifier le choix de la post-fuzzification, ils argumentent que la génération des règles floues directement à partir des données floues cause des redondances. Des règles sont éventuellement générées avec des sous-ensembles flous similaires et ainsi la base des règles se complique. Cela affaiblit l'interprétabilité des règles.

2.5.3 Entropie conditionnelle floue en construction des arbres flous

2.5.3.1 Sélection du meilleur attribut

Dans le chapitre précédent, un modèle hiérarchique pour les mesures de discrimination floues a été présenté. Il permet de caractériser les mesures de discrimination floues. Plusieurs mesures floues utilisées dans un processus de construction d'arbres de décision ont été validées par ce modèle, en particulier l'entropie de Shannon floue ou la mesure d'ambiguïté de classe de Yuan et Shaw. Ces dernières sont couramment utilisées dans la construction d'arbres de décision. D'autres mesures d'entropie floue sont également disponibles et validées dans le cadre du modèle : les entropies conditionnelles floues de Rényi, celles de Daróczy, les formules conditionnelles de la R-norme entropie floue. Nous généralisons l'utilisation de l'entropie de Shannon en introduisant ces mesures d'entropie conditionnelle floues pour construire des arbres de décision.

Nous utilisons ici quelques notations principales que nous avons introduites dans le chapitre précédent (cf. page 13).

1. ξ l'ensemble des exemples (l'appartenance d'un exemple à une classe peut être partielle).
2. $v_1, v_2, .., v_m$ des valeurs associées à un attribut. Ce sont des sous-ensembles flous définis sur le domaine de l'attribut A. Ils sont généralement issus d'un processus de discrétisation. En général, ils constituent une partition normale du domaine. Nous présentons dans la partie suivante de cette section la méthode pour les identifier.
3. ξ_{C_i} le sous-ensemble flou des exemples appartenant à la classe C_i avec un degré strictement positif :
$$\forall e \in \xi \ : \ \sum_{i=1}^{n} \mu_{\xi_{C_i}}(e) = \mu_{\xi}(e)$$
4. ξ_{v_j} le sous-ensemble flou de tous les exemples dont la valeur pour A appartient

à v_j avec un degré strictement positif :

$$\forall e \in \xi \ : \ \sum_{j=1}^{m} \mu_{\xi_{v_j}}(e) = \mu_{\xi}(e)$$

5. $P^*(C_i)$ la probabilité de la classe floue C_i dans ξ :

$$P^*(C_i) = \frac{|\xi_{C_i}|}{|\xi|}$$

6. $P^*(v_j)$ la probabilité qu'un exemple de ξ prenne la valeur v_j pour valeur d'un attribut A :

$$P^*(v_j) = \frac{|\xi_{v_j}|}{|\xi|}$$

7. $P^*(C_i|v_j)$ la probabilité qu'un exemple ayant v_j pour valeur de A appartienne à C_i :

$$P^*(C_i|v_j) = \frac{|\xi_{C_i} \cap \xi_{v_j}|}{|\xi_{v_j}|}$$

où la t-norme utilisée pour définir l'intersection est le produit.

8. $I^*(\xi)$ l'entropie floue de l'ensemble des exemples ξ. Elle est définie par une fonction qui satisfait le niveau $\mathcal{G}^*$ du modèle $\mathcal{FGH}^*$, comme par exemple l'entropie de Rényi, l'entropie de Daróczy et la R-norme entropie.

9. $I^*(\xi|A)$ l'entropie de l'ensemble des exemples conditionnée par l'attribut A. Elle est définie par une fonction qui satisfait le niveau $\mathcal{H}^*$ du modèle $\mathcal{FGH}^*$. C'est une somme pondérée des mesures d'entropie floues $I^*(\xi_{v_j})$:

$$I^*(\xi|A) = \sum_{j=1}^{m} w(v_j) I^*(\xi_{v_j})$$

où w_j est le poids de la modalité v_j. En particulier il est caractérisé dans les formules des entropies conditionnelles de Type 1, 2, 3, 4 (formules (1.10)-(1.13) page 21).

10. $G*(\xi|A)$ le gain d'information floue :

$$G^*(\xi|A) = I^*(\xi) - I^*(\xi|A)$$

L'attribut choisi est celui qui maximise le gain d'information floue :

$$A_{meilleur} = \arg\max_{A} G^*(\xi|A)$$

2.5.3.2 Discrétisation floue

Nous évoquons ensuite la discrétisation de l'attribut numérique A pour obtenir des sous-ensembles flous $v_1, v_2, ..., v_m$ sur son domaine. En utilisant les mesures de discrimination, deux stratégies sont possibles :

1. Les points de coupure précis sont identifiés à l'aide d'une mesure d'entropie classique (cf. section 2.4.2). Ces points de coupure sont rendus flous par un étalement. La largeur de l'étalement est un paramètre de l'algorithme. Cette stratégie est relativement simple à mettre en œuvre. La forme des sous-ensembles est généralement trapézoïdale.
2. Le seconde stratégie est utilisée dans le travail de Peng et Flash [117] décrit précédemment (page 83). Cependant, au lieu de l'entropie floue de Shannon, plusieurs variantes sont possibles. En particulier, les mesures de discrimination floues abordées ci-dessus peuvent être utilisées. D'abord, un ensemble de discrétisations floues potentielles est identifié par une stratégie limitant le nombre de discrétisations floues à évaluer. Chaque discrétisation constitue une partition floue du domaine de l'attribut en question. La forme des sous-ensembles flous est trapézoïdale, triangulaire ou gaussienne pour simplifier le calcul. Une mesure de discrimination floue intervient dans l'évaluation des discrétisations pour choisir celle qui discrimine le plus possible des exemples de différentes classes. C'est une généralisation de la méthode présentée dans la section 2.4.2.

Des mesures de discrimination floues sont implémentées dans la plateforme de construction d'arbres de décision DTGen décrit en annexe.

2.5.4 Expérimentations

Dans cette section, des expérimentations ont été menées pour mettre en évidence la qualité des arbres construits à l'aide de mesures validées par le modèle $\mathcal{FGH}^*$ et ceux construits à l'aide d'autres mesures. Les expérimentations sont menées toujours avec les bases de données répertoriées sur le site de l'UCI (décrites dans le tableau 2.1 - page 61), à l'exception de la base « Waveform » obtenue par un générateur automatique [27]. Le protocole d'expérimentation choisi est identique à celui décrit dans la section 2.3.3.

Les expérimentations sont menées avec l'entropie conditionnelle de Shannon et l'entropie conditionnelle d'événements flous qui sont validées par le modèle hiérarchique. On les qualifiera de *bonnes* mesures. D'autre part, on introduit l'utilisation de la mesure suivante pour illustrer notre comparaison. Cette mesure est considérée comme *mauvaise* car elle ne satisfait pas les propriétés imposées au niveau $\mathcal{G}^*$ du modèle $\mathcal{FGH}^*$:

$$I_b^*(\xi) = I_b^*(P_1^*, P_2^*, .., P_n^*) = -\sum_{i=1}^{n} \log P_i^* \tag{2.2}$$

Le tableau 2.3 décrit les taux de bonnes classifications obtenus sur différentes

bases. Sur chaque base, le meilleur taux de bonnes classifications est mis en gras et le pire taux est mis en italique.

Il montre que l'utilisation de *bonnes* mesures de discrimination conduit à de meilleurs arbres. Les taux de bonnes classifications par l'entropie conditionnelle de Shannon sont toujours meilleurs que ceux produits avec la mesure décrite par l'équation (2.2). Dans la plupart des cas (7/9), l'entropie conditionnelle floue conduit à des arbres de décision plus performants que ceux obtenus avec la mesure décrite par l'équation (2.2) ou avec l'entropie de Shannon.

Les expérimentations montrent aussi que la taille des arbres de décision construits par l'entropie conditionnelle de Shannon et par l'entropie conditionnelle floue est plus petite que celle des arbres de décision construits par la mesure décrite par l'équation (2.2).

Certes, cette expérimentation est à renforcer, sur un plus grand nombre de bases et de mesures. Nous prévoyons également de développer plus avant notre méthode de comparaison des résultats.

Base de données	Ent. de Shannon	Ent. floue	"Mauvaise" mesure
Iris	**96.00** ± 1.63	*95.16* ± 3.05	95.50 ± 2.20
Balance scale	76.52 ± 2.02	**77.56**± 3.09	*74.63* ± 1.58
E. coli	78.08 ± 2.13	**78.16** ± 2.14	*74.39* ± 1.47
Glass identification	65.11 ± 3.62	**68.45** ± 6.12	*61.54* ± 3.59
Ionosphere	87.21 ± 2.55	**87.57**± 1.89	*79.21*± 2.66
Liver-disorders	65.18 ± 3.43	**66.06**± 2.82	*60.46*± 3.51
Pima Indians diabetes	69.37 ± 1.20	**69.79** ± 1.25	*65.13*±1.80
Wine recognition	**91.47** ± 3.45	*90.63*± 1.95	91.33 ± 1.79
Waveform	66.61 ± 2.76	**66.86**±4.22	*64.32*±2.98

TAB. 2.3 – Taux de bonnes classifications moyens et écarts types par différentes mesures (%)

2.5.5 Résumé

Dans cette section, nous avons proposé une nouvelle taxonomie de méthodes de construction d'arbres de décision flous. Cette taxonomie repose sur la méthode de sélection du meilleur attribut et la stratégie d'identification de fonctions d'appartenance.

Nous avons ensuite introduit l'utilisation des mesures de discrimination floues dans la recherche des coupures floues pour un attribut numérique. Les mêmes mesures sont proposées pour la sélection des attributs dont les valeurs sont floues ou issues d'une discrétisation par des coupures floues. Cette proposition est justifiée par la validation, dans le chapitre 1 de ces mesures par un modèle hiérarchique pour des mesures de discrimination floues.

2.6 Conclusion

L'utilisation de l'entropie de Shannon dans la construction des arbres de décision est très fréquente. Cette mesure intervient dans différentes étapes, notamment dans la sélection du meilleur attribut et dans la discrétisation des attributs numériques. L'extension de l'entropie de Shannon pour les événements flous sert également de base à plusieurs algorithmes pour le même but lorsqu'on souhaite prendre en compte l'incertitude et l'imprécision. À côté de ces mesures, plusieurs autres sont recensées.

Nous avons introduit des mesures plus générales que l'entropie de Shannon et l'entropie de Shannon floue dans ce processus. Ce sont des mesures de discrimination classiques et floues, en particulier des entropies conditionnelles de Daróczy, de Rényi et leurs extensions floues. Ces mesures possèdent un certain nombre de caractéristiques définies par le modèle hiérarchique. Elles sont également caractérisées de manière expérimentale. Notre proposition fournit des choix alternatifs des mesures afin d'obtenir des solutions plus adaptées à des problèmes spécifiques.

Dans ce chapitre, une nouvelle taxonomie des méthodes d'induction d'arbres de décision flous est également proposée. Elle est caractérisée par le critère de sélection du meilleur attribut et la stratégie d'identification de sous-ensembles flous.

Finalement, une implémentation de ces mesures est intégrée dans une plateforme d'expérimentation s'intitulée DTGen. Ce logiciel sert à une série d'expérimentations sur plusieurs bases de données, y compris des bases issues des applications réelles que nous présentons dans le chapitre 5.

Chapitre 3

Mesures de discrimination et évaluation de classifieurs

L'évaluation de la performance de classifieurs est une tâche nécessaire mais difficile en apprentissage automatique. Elle permet de comparer des méthodes de classifications entre elles et de comparer les classifieurs. Grâce à l'évaluation, nous pouvons prendre des décisions concernant le choix des méthodes et des classifieurs. Plusieurs critères ont été proposés et utilisés dans un tel processus. Chaque critère mesure une ou plusieurs facettes des classifieurs. Les critères nous aident également à caractériser les méthodes de classification et ainsi mieux comprendre les comportements des méthodes vis-à-vis des données. Cela est nécessaire car il s'avère que dans les travaux existants il n'existe aucune méthode qui soit la meilleure pour tous les problèmes [140]. C'est-à-dire que si un algorithme est plus efficace pour un problème particulier, il est de niveau inférieur pour d'autres problèmes.

Cependant, la plupart des mesures existantes ne prennent pas en compte les caractéristiques du problème telles que la qualité des données disponibles et notamment la distribution des classes. Elles ne considèrent que le résultat de classification obtenu. Cela cause des biais dans l'évaluation et dans l'interprétation des résultats, en particulier pour la comparaison des algorithmes sur des bases de données différentes. Dans ce chapitre, nous justifions l'utilisation de mesures de discrimination comme une alternative pour évaluer des classifieurs. Un des avantages de ce type de mesure est la prise en compte des caractéristiques des données. La justification se base principalement sur le modèle hiérarchique pour les mesures de discrimination qui a été introduit et utilisé dans l'induction d'arbres de décision.

3.1 Introduction

Les techniques d'apprentissage inductif deviennent de plus en plus populaires dans les recherches scientifiques et industrielles. L'ensemble des méthodes d'apprentissage ne cesse de s'agrandir. Naturellement l'évaluation et la comparaison des méthodes entre elles doivent être étudiées. Cela correspond à un besoin réel de la recherche et de l'industrie. Or, la qualité d'une méthode de classification, en particulier

le résultat de classification, est un concept difficile à définir.

Les critères d'évaluation doivent être établis en fonction des propriétés du problème à résoudre. Certaines applications demandent que le classifieur construit donne la plus grande précision possible tandis que certaines préfèrent obtenir la réponse le plus tôt possible (décision médicale par exemple). Dans plusieurs applications, certaines classes sont plus importantes que d'autres donc il est souhaitable que la classification soit la plus juste possible pour ces classes. Dans la suite, nous décrivons le processus d'évaluation et les problématiques de l'évaluation.

3.1.1 Processus de classification

Le processus de résolution d'un problème de classification n'est pas la même dans la réalité que dans l'expérimentation. Dans l'expérimentation on connaît préalablement les classes de tous les exemples. Évidemment on ne les utilise pas dans la construction du modèle de classification, mais on peut les utiliser dans l'évaluation des modèles obtenus. Tandis que dans la réalité, on ne connaît pas a priori les classes des exemples même dans la phase d'évaluation.

Dans la *réalité*, un problème de classification doit être résolu en suivant quatre étapes :

1. *Acquisition de données :* Les données de différentes sources sont acquises : expérience, mesure, données dans les stocks, etc. On espère souvent que les données acquises représentent bien l'univers des données possibles. Des pré-traitements sont utilisés éventuellement si nécessaire. Parmi ces pré-traitements, on peut citer le traitement des valeurs manquantes abordé dans le chapitre 4 de cette thèse.
2. *Construction d'un modèle de classification :* À partir des données et éventuellement des expériences et de connaissances acquises dans l'étape précédente, on construit un modèle de classification (classifieur) par un processus d'apprentissage automatique. Le classifieur est une façon de modéliser les données disponibles. En apprentissage, on suppose souvent que les exemples ayant des caractéristiques communes se trouvent dans une même classe. On essaie d'exploiter le plus d'informations possibles liées à la détermination de la classe qui servira plus tard à classifier de nouveaux exemples. Le modèle peut être considéré en tant que conteneur des informations synthétiques exploitées par le processus d'apprentissage au travers de toutes les données disponibles. Le but est de mettre dans le conteneur le plus possible d'informations utiles. Ainsi, il faut faire face à différentes difficultés : des bruits qui dégradent les données, des incertitudes, des imprécisions et des indisponibilités de données, etc.
3. *Utilisation du modèle de classification :* Dans cette étape, le modèle déduit dans la première étape est utilisé pour déterminer les classes auxquelles appartiennent de nouveaux exemples. Grâce à des informations descriptives d'un exemple, le modèle de classification le classifie dans la classe dans laquelle il existe des exemples ayant les mêmes caractéristiques que lui. Il n'est pas obligatoire que toutes les informations sur le nouvel exemple soient utilisées pour

déterminer sa classe. Seules les informations discriminantes, intéressantes pour le modèle, sont employées.

4. *Évaluation du modèle de classification :* Dans cette étape, le modèle et la méthode de construction du modèle sont évalués en utilisant différents critères. Selon le problème et la méthode abordée, on cherche à utiliser les critères adéquats. Il peut arriver qu'on ne connaisse pas la vraie classe des exemples. C'est pour cette raison que l'évaluation directe est parfois impossible. Dans ces cas, il faut éventuellement évaluer par des critères indirects qui estiment, par exemple, la conséquence d'utiliser les résultats de classification pour un but quelconque.

Dans l'*expérimentation*, on dispose souvent d'une base de données qui contient des exemples et on connaît a priori la classe à laquelle appartient chaque exemple. Cela constitue la différence essentielle entre des expérimentations et des problèmes réels. Pour évaluer une méthode de classification, on partitionne aléatoirement, selon un protocole bien précisé, la base initiale en deux : l'une sert de base d'apprentissage et l'autre sert de base de test. La base d'apprentissage est exploitée par un algorithme pour construire un modèle de classification. Ce modèle sert ensuite à classifier des exemples de la base de test. Enfin, la performance du classifieur est habituellement évaluée par les résultats de classification sur cette base de test. Les classes réelles d'exemples sont comparées avec les classes prédites par le classifieur.

Pour bien comprendre les méthodes et les résultats ainsi que pour faciliter les analyses des résultats, le protocole d'expérimentation doit être bien formulé. Le protocole d'expérimentation influence significativement le résultat obtenu. Il décrit précisément et clairement des conditions et le déroulement d'une expérience. Le but est que l'expérience puisse être reproduite à l'identique. Cela permet aussi de comparer les méthodes entre elles. Dans la pratique, on utilise souvent la validation croisée et ses variantes. Pour réduire le biais du résultat, dans n'importe quel protocole, aucune information des exemples dans la base de test ne peut être utilisée dans la construction du modèle de classification. Dans la majorité des cas, l'expérimentation doit être répétée plusieurs fois.

3.1.2 Problématique de l'évaluation et de la comparaison

Plusieurs critères ont été proposés et utilisés dans l'évaluation d'un processus de classification. Chaque critère mesure une ou plusieurs facettes des classifieurs et est lié à une caractéristique souhaitée des classifieurs. Les critères nous aident à caractériser des méthodes de classification et ainsi à mieux comprendre les comportements des classifieurs. L'ordre d'importance des critères est différent de l'un à l'autre. Évidemment, les problèmes à résoudre sont très divers et aucun critère ne peut donc satisfaire tous les besoins. Ce n'est donc pas étonnant qu'on propose de plus en plus de critères pour évaluer différentes facettes d'un problème. Or, avec plusieurs critères, déterminer si un classifieur est meilleur qu'un autre, même sur les mêmes bases de test, n'est pas évident. Une combinaison des critères semble être une stratégie d'évaluation favorable. Cette combinaison doit être construite en fonction

de la finalité souhaitée. L'agrégation des critères constitue une tâche nécessitant une grande investigation. La problématique devient plus difficile si les tests sont effectués sur des bases différentes.

D'abord, toutes les mesures d'évaluation sont conçues sous un certain nombre d'hypothèses. Par exemple, une hypothèse souvent imposée est que tous les exemples soient d'égale importance, et que la base d'apprentissage et la base de test soient distribuées de la même manière. Les bonnes classifications de n'importe quel exemple sont appréciées de façon identique et les mauvaises classifications des exemples sont jugées également de façon identique. Autrement dit, on présume que le coût d'erreur de toutes les classifications est identique. Mais ces hypothèses ne sont pas valides pour tous les problèmes.

Ensuite, un critère peut être biaisé par des caractéristiques des problèmes, notamment celles de la base de données. Entre autres, ces caractéristiques incluent des mesures simples, des mesures statistiques, des mesures basées sur la théorie de l'information [67]. Les mesures simples sont le nombre d'exemples, le nombre d'attributs, le nombre de classes, la proportion d'attributs binaires, l'erreur quantifiée par le coût. Les mesures statistiques incluent le rapport entre les écart-types (*standard deviation ratio*), la valeur moyenne de corrélation (*mean value of correlation*), le coefficient de *skewness* qui mesure le degré d'asymétrie de la distribution, le coefficient de *kurtosis* qui mesure le degré d'écrasement de la distribution. Les mesures issues de la théorie de l'information incluent l'entropie des classes, l'entropie des attributs, l'entropie mutuelle moyenne entre la classe et les attributs, ainsi que le rapport signal/bruit.

Parmi les caractéristiques, la proportion des classes dans la base de données influe fortement sur le résultat comme l'ont remarqué plusieurs auteurs [88, 121]. Dans certaines applications les classes ont parfois des fréquences très différentes. Par exemple, dans le commerce électronique : 99% de visiteurs n'achètent rien et seulement 1% d'entre eux achète quelque chose ; dans la sécurité, 99.99% de personnes ne sont pas des terroristes ; etc. Aussi, un classifieur naïf qui classifie tous les exemples dans la classe majoritaire peut avoir un taux très élevé de bonnes classifications : 99% dans le problème du commerce électronique, 99.99% dans le problème de la sécurité, mais sera alors plutôt inutile.

Il est évident que si la base initiale est *facile* (c'est-à-dire que des exemples se séparent bien dans l'espace ou que la classe majoritaire contient une grosse partie de la base par exemple) on obtient facilement un taux élevé de bonnes classifications. Aussi il serait préférable de prendre en compte la nature de la base lorsqu'on évalue le résultat des tests. Cela permet de faire ressortir la contribution du modèle dans la détermination de la classe de l'exemple.

Pour valider les méthodes, on utilise fréquemment des bases conventionnelles (bases de l'UCI [113] par exemple) espérant qu'une méthode qui fonctionne bien sur ces bases fonctionne aussi bien sur d'autres données. Dans [121], les auteurs ont pris un exemple pour montrer que la distribution existante dans la base de l'UCI est parfois loin de la distribution réelle. Cependant, dans la réalité nous avons besoin de comparer les méthodes testées sur les bases réelles différentes. Quand on évalue

une méthode, chacun utilise parfois sa propre base, soit par intérêt particulier, ou tout simplement parce que les travaux ont une finalité vers une application spécifique, donc une base spécifique. Cela cause des difficultés dans la comparaison des différentes approches proposées. Dans certain travaux, on normalise [67] les caractéristiques des bases de données et les résultats de tests pour les mettre aux mêmes échelles. La normalisation peut tenir compte des caractéristiques des problèmes à confronter. Par exemple, le résultat obtenu est normalisé avec le meilleur résultat connu ou par celui de la méthode qui classifie tous les exemples dans la classe majoritaire.

Dans ce chapitre, nous souhaitons aborder le problème sous un autre angle et proposons une nouvelle technique pour évaluer la performance du modèle de classification, qui est capable de relâcher quelques contraintes d'utilisation et permet de caractériser un classifieur par sa capacité de discrimination des exemples en tenant compte de caractéristiques des bases de données. Il s'agit également d'une mesure normalisée par l'entropie de la base de test par rapport aux classes.

En classification, il apparaît deux partitions de la base de test. La première partition est naturelle : les exemples sont regroupés selon leurs classes. La deuxième partition est celle qui générée par le classifieur. Dans les sections qui suivent, nous proposons de considérer l'adéquation entre ces deux partitions. Cela permet d'évaluer la capacité de discrimination des modèles de classification par rapport des classes et de comparer des classifieurs entre eux. L'idée initiale a été introduite pour l'induction d'arbres de décision qui aide à choisir le meilleur attribut, selon lequel on partitionne une base d'apprentissage : l'attribut choisi est le plus discriminant par rapport à des classes. Il nous fournit un critère supplémentaire possédant certaines caractéristiques intéressantes dans l'évaluation de classifieurs. Entre autres, il permet d'éliminer le biais sur la distribution des classes en évaluant la différence entre l'impureté de la base avant et après la classification.

3.2 Critères d'évaluation des modèles de classification

3.2.1 Taxonomies et notations

3.2.1.1 Taxonomies

Plusieurs critères d'évaluation sont établis dans la littérature, on peut citer entre autres certaines catégories principales :

1. Critères basés sur la précision, comme par exemple : taux de bonnes classifications, précision, rappel, F-mesure, critères basés sur une courbe ROC, analyses du tableau de confusion. Ce sont les critères principaux et les plus couramment utilisés.
2. Critères basés sur l'entropie, comme par exemple l'entropie croisée, le critère proposé par Kononenko et Bratko [88], la mesure de divergence dirigée, la

mesure de récompense d'information, le gain d'entropie [48, 18], la mesure d'évaluation proposée par Ben Amor *et al.* [10].

3. Complexité du classifieur (par exemple longueur maximale, nombre de nœuds, nombre de feuilles dans le cas d'un arbre de décision).
4. Interprétabilité du modèle de classification. Ce critère est assez subjectif. Les techniques d'arbres de décision sont réputées pour leur interprétabilité. Un classifieur simple est souvent plus interprétable.
5. Vitesse : à la fois le temps nécessaire pour la construction du classifieur et pour classer un exemple.
6. Robustesse : la sensibilité de la méthode par rapport à des modifications mineures de la base d'apprentissage. Cette capacité permet de résister au bruit présent dans les données.
7. Capacité de passage à l'échelle.

Parmi ces critères, les 5 premiers concernent les modèles de classification. Les 3 derniers concernent les méthodes de construction de modèles. Le cinquième critère concerne à la fois les méthodes de construction de modèles et les modèles eux-mêmes.

En statistique, pour l'évaluation d'un modèle, on s'intéresse plutôt aux 3 premiers critères. Entre autres, le critère d'information bayésien BIC (*Bayesian Information Criterion*) et le critère d'information AIC d'Akaike (*Akaike Information Criterion*) sont souvent utilisés mais plutôt pour l'évaluation de la capacité de description des données d'un modèle statistique. Ces critères sont une combinaison de la qualité d'ajustement (estimée par une mesure statistique comme le χ^2 sur la base d'apprentissage) et la complexité d'un modèle. Ainsi, ils permettent d'arbitrer entre complexité et qualité d'ajustement dans la sélection de modèles sachant que, en général, il existe un compromis entre la complexité et la qualité du modèle statistique. En particulier, Ritschard et Zighed [136, 137] ont proposé d'adapter ces mesures au cas des arbres de décision. Ritschard [135] a remarqué que l'arbre correspondant à un BIC minimum assure en moyenne le meilleur taux de bonnes classifications.

Une liste complète des mesures avec leurs descriptions, ainsi que l'étude empirique de ces mesures se trouvent, entre autres, dans les travaux de Caruana [31, 32, 33].

Dans [33], les mesures d'évaluation sont classées en 3 catégories selon la manière d'interpréter des résultats obtenus, sachant que cette taxonomie ne couvre pas les mêmes critères que la précédente :

1. Mesures liées à un seuil *(threshold metric)* : pour cette catégorie de mesures, on ne s'intéresse qu'à savoir si la valeur donnée par le classifieur est inférieure ou supérieure à un seuil fixé. Il n'est pas important de savoir si cette valeur est proche du seuil ou pas. Toutes les mesures qui sont calculées après avoir comparé la valeur donnée par le classifieur et le seuil appartiennent à cette catégorie, y compris la plupart des mesures recensées ici : la précision, lift mesure, taux de vrais positifs, le coefficient de corrélation,...
2. Mesures liées à un ordonnancement *(rank metrics)* : dans le cas à deux classes, positive et négative, on s'intéresse à savoir comment les cas positifs sont ordonnés avant les cas négatifs mais pas directement à la valeur donnée par le

classifieur. L'AUC (*Area Under the Curve*) et la précision moyenne sont dans cette catégorie. Cette catégorie est largement utilisée en recherche d'information.

3. Mesures liées à des probabilités : ces mesures prennent en compte telles quelles les valeurs numériques fournies par un classifieur. Ces valeurs sont interprétées comme les probabilités que l'exemple appartienne à une classe. L'erreur carrée et l'entropie croisée sont dans cette catégorie.

Pour chaque catégorie de classifieurs, un ensemble de critères propres est défini. Ces critères sont spécifiques à la catégorie en question et servent à comparer les classifieurs de la même catégorie. Par exemple, dans [63] les critères suivants pour les arbres de décision sont établis :

1. Taux de bonnes classifications sur les nouveaux exemples (à maximiser)
2. Nombre de règles (nombre de feuilles) (à minimiser)
3. Nombre de nœuds (à minimiser)
4. Nombre de pré-conditions dans les règles (à minimiser)
5. Nombre moyen d'exemples supportés par une règle (à maximiser)
6. Nombre moyen de tests par exemple (à minimiser).

Parmi ces critères, le dernier influence directement la vitesse de classement. Les deuxième, troisième et quatrième sont des mesures sur la taille des arbres. Les rapports entre ces critères ont été étudiés. Dans plusieurs cas, l'interprétabilité et la sensibilité basées sur une analyse géométrique des arbres de décision sont considérées [8]. Pour des arbres de décision flous, on peut aussi évaluer le volume de l'espace flou et le gain de performance par rapport à des méthodes classiques.

3.2.1.2 Notations

Dans ce chapitre, on utilise les notations suivantes. Soit $\xi_T = \{e_1, e_2, ..., e_N\}$ l'ensemble des exemples qui forment la base de test. Supposons qu'un exemple e appartienne à la classe $e(C)$ de l'ensemble des classes $\mathcal{C} = \{C_1, C_2, .., C_n\}$. Si $n = 2$, pour simplifier, on parle de classe positive (C_2) et de classe négative (C_1). Ces deux classes sont étiquetées respectivement par 1 et 0 si on préfère des valeurs numériques pour les classes.

Dans le cas de classification probabiliste, chaque exemple est associé à une distribution de probabilité $(P_{C_1}(e), P_{C_2}(e), .., P_{C_n}(e))$, dans laquelle $P_{C_i}(e)$ est la probabilité que l'exemple e appartienne à la classe C_i. C'est la cible du classifieur. La distribution de probabilité pour un exemple e qui appartient à une classe unique $e(C)$ est sous la forme $(0, .., 1, .., 0)$, où 1 correspond à la classe $e(C)$.

Supposons que, a priori, la probabilité qu'un exemple e appartienne à la classe C de $\mathcal{C}$ soit $P_e(C)$. Aussi, a priori, la distribution originale de probabilité est :

$$(P_e(C_1), P_e(C_2), .., P_e(C_n))$$

Cette distribution de probabilité est estimée à partir des connaissances a priori (sans regarder la base de test) comme la fréquence de chaque classe dans la base d'apprentissage.

Supposons que, a posteriori, le classifieur retourne la probabilité $P'_e(C)$ qu'un exemple e soit dans la classe C. Aussi, a posteriori, la distribution prédite de probabilité est $(P'_e(C_1), P'_e(C_2), .., P'_e(C_n))$.

Jusqu'ici, on a alors trois distributions de probabilité $(P_{C_1}(e), P_{C_2}(e), .., P_{C_n}(e))$, $(P_e(C_1), P_e(C_2), .., P_e(C_n))$ et $(P'_e(C_1), P'_e(C_2), .., P'_e(C_n))$. Pour avoir une idée sur la précision de la classification (c'est-à-dire la cohérence entre la distribution prédite et la distribution originale), il suffit de comparer la première et la troisième distributions. Il est souhaitable qu'elles soient identiques. Pour évaluer la contribution de l'algorithme d'apprentissage sur l'identification de la classe des exemples, il faut prendre en compte la deuxième distribution.

Dans le cas à deux classes, un classifieur donne une valeur numérique entre 0 (classe négative ou classe C_1) et 1 (classe positive ou classe C_2). Cette valeur est la probabilité, selon le classifieur, que cet exemple soit dans la classe positive, autrement dit c'est $P'_e(C_2)$. On a :

$$P'_e(C_1) = 1 - P'_e(C_2)$$

Grâce à ce lien, on parle donc souvent de $P'_e(C_2)$ au lieu de $(P'_e(C_1), P'_e(C_2))$.

Avec une distribution de probabilité comme résultat de classification, si l'on souhaite obtenir une seule classe, on doit choisir la classe la plus probable. Dans ce choix, on peut éventuellement intégrer d'autres facteurs tels que le coût d'erreur. Par exemple, dans le cas à deux classes, on fixe un seuil s. Si $P'_e(C_2) \geq s$ alors la classe prédite est positive ; sinon la classe prédite est négative. s est choisi selon le problème posé. Plus s est grand, plus on est prudent (car le coût d'erreur est plus important) pour décider si un exemple appartient à la classe positive. Dans plusieurs cas où les coûts d'erreur sont identiques ou inconnus pour tous les exemples, s est fixé à 0.5.

Classe réelle \ Classe prédite	C_1	C_2	...	C_n
C_1	N_{11}	N_{12}	...	N_{1n}
C_2	N_{21}	N_{22}	...	N_{2n}
...	...	...	...	...
C_n	N_{n1}	N_{n2}	...	N_{nn}

TAB. 3.1 – Matrice de confusion

On considère le cas le plus simple où chaque exemple appartient à une seule classe et le résultat fourni par le classifieur est traité de manière à ce qu'on obtienne une seule classe. Dans ce cas, la matrice de confusion contient des informations sur la classification réelle et la classification prédite faite par un modèle de classification. Il s'agit d'une table de contingence confrontant les classes prédites (colonnes) et les

classes désirées (lignes) pour les exemples de la base de test. Dans le tableau 3.1, N_{ij} est le nombre d'exemples de la classe C_i classés dans la classe C_j. La performance d'un modèle de classification est généralement évaluée en se basant sur les informations figurant dans cette matrice. En réduisant cette matrice possédant n^2 nombres à une seule valeur numérique pour évaluer la performance d'un classifieur, on perd la richesse d'information donnée par la matrice. Mais cela est nécessaire, car d'une part cela donne une information plus synthétique et plus visuelle, et d'autre part cela sert à la comparaison entre les classifieurs. Il est donc nécessaire de caractériser les mesures d'évaluation et de concevoir un ensemble de mesures qui s'adaptent au mieux au problème de classification considéré.

Dans la suite, nous établissons un état de l'art des mesures d'évaluation. D'abord, un survol rapide des mesures d'évaluation principales est présenté. Ensuite, nous nous concentrons sur des critères provenant de la théorie de l'information.

3.2.2 Mesures non basées sur la théorie de l'information

Le taux de bonnes classifications p est un des critères les plus utilisés. Il est défini comme le rapport entre le nombre d'exemples correctement classifiés et le nombre total d'exemples.

$$p = \frac{\sum_{i=1}^{n} N_{ii}}{\sum_{i=1}^{n} \sum_{j=1}^{n} N_{ij}}$$

On parle aussi de taux d'erreur *err* qui est défini par :

$$\text{err} = 1 - p$$

L'objectif des méthodes de construction de modèles de classification est de maximiser ce taux de bonnes classifications, c'est-à-dire de minimiser le taux d'erreur. L'utilisation du taux de bonnes classifications se fait toujours sous la condition implicite que la distribution des classes est connue a priori. On suppose aussi en utilisant cet indice que le coût de mauvaise classification est égal pour tous les exemples. Cela rend impossible l'utilisation du taux de bonnes classifications dans certaines catégories de problèmes. Par exemple, dans une application médicale, une mauvaise classification d'un patient malade coûterait beaucoup plus qu'une mauvaise classification d'une personne bien portante. Une analyse plus détaillée sur le taux de bonnes classifications se trouve dans [121, 88].

Une mesure déduite du taux d'erreur est le pourcentage de réduction d'erreur. Cela évalue combien le taux d'erreur est réduit par un processus d'apprentissage. Par exemple, dans l'application de commerce électronique introduite en 3.1.2, si le taux de bonnes classifications est de 99.5%, cela veut dire que le taux d'erreur est réduit de 1% (sans apprentissage) à 0.5% (avec apprentissage). Aussi, il y a 50% de réduction pour le taux d'erreur. Le même principe est applicable non seulement avec le taux d'erreur mais aussi pour d'autres mesures.

Dans l'état de l'art, on considère souvent le cas à deux classes, car il s'agit d'un cas plus simple à étudier. Dans la réalité, ce cas apparaît plus souvent que d'autres,

comme par exemple : l'utilisateur novice ou expérimenté dans la classification des utilisateurs par les traces d'interaction homme-machine (section 5.1), la situation de crise ou la situation normale dans la prédiction de risque, le courriel non sollicité ou messages sollicités dans le filtrage des messages non sollicités, etc. Dans la recherche de documents, il y a deux classes de documents : pertinent ou non-pertinent. Dans la suite, pour simplifier, nous ne présentons que des mesures pour deux classes. En général, ces mesures peuvent être étendues pour les cas à plusieurs classes et pour des classifieurs qui donnent une distribution de probabilité comme résultat.

Classe prédite / Classe réelle	négative	positive
négative	N_{11}	N_{12}
positive	N_{21}	N_{22}

TAB. 3.2 – Matrice de confusion pour deux classes

Quelques mesures déduites de la matrice de confusion (cf. tableau 3.2) sont présentées ci-dessous.

Le taux de vrais positifs (rappel ou sensibilité) exprime la probabilité de bien classer un exemple de la classe positive :

$$R = \frac{N_{22}}{N_{21} + N_{22}} \tag{3.1}$$

Le taux de vrais négatifs (spécificité) exprime la probabilité de bien classer un exemple de la classe négative :

$$\frac{N_{11}}{N_{11} + N_{12}}$$

Le taux de faux positifs exprime la probabilité de mal classer un exemple de la classe négative :

$$\frac{N_{12}}{N_{11} + N_{12}}$$

Le taux de faux négatifs exprime la probabilité de mal classer un exemple de la classe positive :

$$\frac{N_{21}}{N_{21} + N_{22}}$$

Les 4 valeurs ci-dessus sont en fait des probabilités conditionnées par les classes réelles des exemples : le taux de vrais positifs et le taux de faux négatifs sont estimés sur les exemples de la classe positive ; le taux de vrais négatifs et le taux de faux

positifs sont estimés sur les exemples de la classe négative. Les deux premières sont à maximiser et les deux dernières sont à minimiser.

Les 4 valeurs ci-dessous sont également des probabilités conditionnées mais par les classes prédites des exemples.

La précision (valeur prédite positive) exprime la probabilité de bien classer un exemple dans la classe positive :

$$P = \frac{N_{22}}{N_{12} + N_{22}} \tag{3.2}$$

La valeur prédite négative exprime la probabilité de bien classer un exemple dans la classe négative :

$$\frac{N_{11}}{N_{11} + N_{21}} \tag{3.3}$$

La probabilité qu'un exemple ayant été classé dans la classe négative soit réellement dans la classe positive (*prediction -conditionned fallout*) :

$$\frac{N_{21}}{N_{11} + N_{21}} \tag{3.4}$$

La probabilité qu'un exemple ayant été classé dans la classe positive est réellement dans la classe négative (*negative - conditioned miss*) :

$$\frac{N_{12}}{N_{12} + N_{22}} \tag{3.5}$$

Parmi les 4 mesures ci-dessus, les deux premières (la précision et la valeur prédite négative) sont à maximiser et les deux dernières, (3.4) et (3.5) sont à minimiser.

La précision (3.2) et le rappel (3.1) sont très utilisés en recherche d'information. La F-mesure est la moyenne harmonique pondérée du rappel et de la précision. La moyenne est équilibrée quand $\beta = 1$.

$$F = \frac{(\beta^2 + 1)PR}{R + \beta^2 P}$$

Le rapport de cotes (*odds-ratio*), au sens général, est une grandeur statistique permettant de comparer deux facteurs de risques différents dans une population. Il est largement utilisé dans le domaine médical. Il est défini comme suit[1] :

$$\frac{\frac{N_{11}}{N_{12}}}{\frac{N_{21}}{N_{22}}} = \frac{N_{11}N_{22}}{N_{12}N_{21}}$$

Dans notre contexte, il faut le maximiser.

[1] http ://rocr.bioinf.mpi-sb.mpg.de/ROCR.pdf

Le coefficient Kappa mesure une corrélation entre deux variables statistiques. En apprentissage, il évalue l'accord entre la distribution naturelle et la distribution issue d'un classifieur en prenant en compte l'accord par « chance » . Il est considéré comme une amélioration du taux de bonnes classifications. Comme la base contient $(N_{11} + N_{12})$ exemples de la classe négative et $(N_{22} + N_{21})$ exemples de la classe positive, quand on classifie $N_{11} + N_{21}$ (resp. $(N_{12} + N_{22})$) exemples dans la classe négative (resp. positive) de façon aléatoire, l'espérance du nombre d'exemples correctement classifiés dans la classe négative (resp. positive) est :

$$\frac{(N_{11} + N_{12})}{N}(N_{11} + N_{21})$$

resp.

$$\frac{(N_{21} + N_{22})}{N}(N_{12} + N_{22})$$

L'espérance du nombre d'exemples bien classifiés par « chance » est donc :

$$\frac{(N_{22} + N_{21})(N_{22} + N_{12}) + (N_{11} + N_{12})(N_{11} + N_{21})}{N}$$

La contribution effective du classifieur est donc :

$$N_{11} + N_{22} - \frac{(N_{22} + N_{21})(N_{22} + N_{12}) + (N_{11} + N_{12})(N_{11} + N_{21})}{N}$$

Le coefficient Kappa, introduit par J. Cohen [38], évalue un « taux de bonnes classifications » et exclut la partie des bonnes classifications par « chance » .

$$Ka = \frac{(N_{22} + N_{11}) - \frac{(N_{22} + N_{21})(N_{22} + N_{12}) + (N_{11} + N_{12})(N_{11} + N_{21})}{N}}{N - \frac{(N_{22} + N_{21})(N_{22} + N_{12}) + (N_{11} + N_{12})(N_{11} + N_{21})}{N}}$$

Ce coefficient est toujours inférieur ou égal à 1 et on souhaite le maximiser. Un coefficient Kappa négatif signifie que le classifieur est pire qu'un classifieur aléatoire. La définition de cette mesure peut s'étendre facilement aux cas à plusieurs classes. Ce coefficient peut aussi servir dans la sélection d'attribut [95].

Le coefficient de corrélation[2],[3], aussi appelé le coefficient de corrélation de rang partiel de Kendall (*Kendall partial rank correlation*) est une mesure non-paramétrique de corrélation partielle :

$$Ke = \frac{N_{22}N_{11} - N_{12}N_{21}}{\sqrt{(N_{22} + N_{21})(N_{11} + N_{12})(N_{12} + N_{22})(N_{11} + N_{21})}}$$

Ce coefficient prend ses valeurs entre -1 et 1. Une classification parfaite correspond à 1 et la classification aléatoire correspond à 0. Une valeur négative signifie une classification pire que la classification aléatoire.

[2]http ://www.utdallas.edu/∼herve/Abdi-KendallCorrelation2007-pretty.pdf
[3]http ://www.cons-dev.org/elearning/stat/stat7/st7.html

Si le coût d'erreur (lors des mauvaises classifications) et le bénéfice (lors des bonnes classifications) sont différents classe par classe, le classifieur doit maximiser le bénéfice :

$$\frac{N_{11}B(n,n) - N_{12}C(n,p) - N_{21}C(p,n) + N_{22}B(p,p)}{N_{11} + N_{12} + N_{21} + N_{22}}$$

où $B(n,n)$ et $B(p,p)$ sont respectivement les bénéfices quand des exemples des classes négative et positive sont bien classifiés ; $C(n,p)$ et $C(p,n)$ sont respectivement les coûts pour des exemples de la classe négative et positive qui sont mal classifiés. Dans plusieurs problèmes, on utilise $B(n,n) = B(p,p) = 0$, et on vise simplement à minimiser le coût causé par l'erreur.

La *Lift value* [74] est préférée dans l'analyse marketing. Elle mesure combien de fois le classifieur en question est meilleur par rapport à un classifieur aléatoire sur un sous-ensemble de données :

$$LIFT = \frac{\text{taux de vrais positifs}}{\text{taux d'exemples classifiés dans la classe positive}}$$

On calcule souvent cette mesure sur un sous-ensemble de données dans la base de test. Cette mesure est intéressante en marketing, par exemple quand on souhaite envoyer une publicité à un nombre limité de personnes d'une population dont 5% seront intéressés. Si on l'envoie à un ensemble de personnes choisies aléatoirement, seuls 5% d'entre eux sont intéressés. Mais si on l'envoie à un groupe de personnes présélectionnées par un classifieur, on peut espérer que le pourcentage de personnes intéressées sera plus élevé. Dans ce cas, on ne s'intéresse pas aux autres personnes de la population.

L'erreur carrée (*root mean squared error - RMSE*) est plutôt utilisée dans la régression, où la classe est une valeur numérique. Cette mesure est applicable en classification binaire où les classes sont 0 et 1. C'est une des mesures liées à des probabilités. Elle est définie par :

$$RMSE = \sqrt{\frac{1}{N}\sum(\text{Prédite}(C) - \text{Réelle}(C))^2}$$

Quand les résultats sortis par un classifieur sont des nombres réels de [0,1], on peut trier les exemples selon l'ordre décroissant des valeurs correspondantes. Plus un exemple se trouve au début de la liste, plus il est probable qu'il appartienne à la classe positive. Les mesures liées à un ordonnancement sont définies en s'appuyant sur cette liste triée.

La précision moyenne est une mesure liée à un ordonnancement. C'est la moyenne des précisions (taux de bonnes classifications) dont chacune est calculée sur les m premiers exemples ($m = 1, 2, .., n$ avec n le nombre d'exemples) de la liste mentionnée ci-dessus.

Le seuil de rentabilité ou point mort (*break even point - BEP*) est le point où la précision et le rappel sont égaux. La précision moyenne et le BEP sont à maximiser.

Avec la précision moyenne, ce sont des mesures populaires en recherche d'information plutôt qu'en classification.

Dans la dernière décennie, la courbe ROC (*Receiver Operating Characteristic*) a souvent été employée dans la communauté d'apprentissage automatique [25, 62, 121] pour l'évaluation des performances de classifieurs. Elle joue aussi un rôle important dans le domaine médical. La courbe ROC est une mesure de performance de classification à 2 dimensions. Elle représente le compromis entre les bénéfices (vrais positifs) en ordonnée et les pertes (faux positifs) en abscisse. Chaque classifieur discret renvoie seulement une classe de décision pour un exemple, il produit donc un point (taux de faux positifs, taux de vrais positifs) dans l'espace ROC. Quand un classifieur ne renvoie pas seulement une classe de décision mais un score ou une probabilité, un seuil peut être utilisé pour décider la classe d'un exemple. Avec un seuil, on peut déterminer un point dans l'espace ROC. En faisant varier le seuil de $-\infty$ à $+\infty$, on obtient une courbe dans l'espace ROC, la courbe ROC.

On estime que l'utilisation de cette courbe est mieux justifiée statistiquement par rapport aux autres mesures. À partir d'une courbe ROC, l'aire sous la courbe (AUC) peut être calculée. Une propriété intéressante de la courbe ROC est son insensibilité à la distribution des classes. Dans plusieurs cas, comme la base de règles, où un classifieur produit seulement une classe par exemple, l'utilisation d'un tel classifieur sur une base de test ne produit qu'une matrice de confusion et un seul point sur la courbe ROC est donc déterminé. Dans ces cas, pour utiliser les courbes ROC, les classifieurs devraient être modifiés pour produire des valeurs intermédiaires à chaque exemple plutôt que juste une seule classe. D'ailleurs, l'analyse de courbes ROC ne convient pas au choix de classifieur. Pour conclure qu'un classifieur est meilleur qu'un autre, il doit être meilleur que l'autre dans la totalité de l'espace ROC [120] c'est-à-dire qu'il doit avoir un taux de vrais positifs plus élevé et un taux de faux positifs moins élevé que l'autre.

Il existe une technique [120] dérivée de la courbe ROC qui propose d'utiliser l'enveloppe convexe (*ROC convex hull*) de la courbe ROC pour comparer des classifieurs entre eux quand on prend en compte le coût d'erreur des classes et la probabilité des classes. Il y a quelques années, une autre amélioration de la courbe ROC, nommée la courbe de coût (*cost curves*) a été proposée par Drummond et Holte [58, 60]. Il s'agit d'une technique de visualisation de la performance des classifieurs en prenant en compte le coût d'erreur et la probabilité des classes dans l'espace à 2 dimensions. L'abscisse est la fonction de probabilité de coût (*probability cost function*) et l'ordonnée est le coût prédit normalisé (*normalized expected cost*). Elle permet de résoudre un certain nombre de problèmes [59] concernant notamment la comparaison entre des classifieurs, la performance moyenne, l'intervalle de confiance d'une performance, etc. Ce sont des problèmes que la courbe ROC ne peut pas résoudre.

Dans [32], l'auteur tente de proposer une combinaison des mesures. La mesure SAR combine trois mesures : erreur carrée (RMSE), taux de bonnes classifications (ACC) et ROC. Chacune de ces mesures appartient à une catégorie différente : le taux de bonnes classifications est une mesure liée à un seuil, l'erreur carrée est une mesure liée à des probabilités et le ROC est une mesure liée à un ordonnancement. Le choix

de ces trois mesures est justifié par le fait que le taux de bonnes classifications est la mesure la plus populaire parmi les mesures liées à un seuil, ROC est la meilleure mesure parmi les mesures liées à un ordonnancement, et RMSE est la meilleure mesure de sa catégorie. La mesure SAR est définie comme suit :

$$\frac{(1 - RMSE) + ACC + ROC}{3}$$

En plus des mesures universelles qui peuvent être appliquées dans plusieurs applications, il y a des mesures spécifiques à un domaine donné. Par exemple, SLQ (Slac Q-Score) a été développé pour certains problèmes de physique des particules [31]. Cependant, dans le cadre de cette thèse, on ne s'intéressera pas à cette catégorie de mesures.

La plupart des mesures citées ci-dessus, sauf le coefficient Kappa, évaluent la relation entre les classes prédites et les classes réelles des exemples d'une base de test en ne s'appuyant que sur des informations obtenues postérieurement (information conditionnelle d'un classifieur) et ne prennent pas en compte les caractéristiques du problème considéré (information antérieure). La difficulté de chaque problème n'est pas prise en compte lors de l'évaluation du résultat. Toutefois, la caractérisation d'une base de données en apprentissage est elle-même un problème majeur, voir par exemple [73]. Nous ne creusons pas cette problématique dans le cadre de cette thèse. Le résultat obtenu est ainsi biaisé par la complexité du problème. Si un problème est « facile » il y a plus de chances qu'un exemple soit correctement classé. Aussi, avec moins d'effort, on obtient un bon résultat. En regardant seulement le résultat de classification, on n'a aucune idée précise sur le succès du classifieur. Cela rend impossible de comparer les résultats obtenus avec des problèmes différents.

3.2.3 Mesures basées sur la théorie de l'information

Dans la suite de cette section, les critères basés sur la théorie de l'information pour l'évaluation de classifieurs sont présentés et formalisés sous un formalisme commun. La plupart de ces mesures sont proposées pour la classification probabiliste et elles prennent la classification classique (la cible est une classe et le classifieur affecte une classe unique à un exemple) en cas particulier. Elles évaluent habituellement la cohérence entre la distribution de probabilité prédite par le modèle de classification et la distribution de probabilité réelle pour chaque exemple puis les agrègent sur l'ensemble des exemples pour obtenir l'évaluation globale.

La mesure basée sur l'entropie croisée est décrite dans [31] pour le cas à deux classes : négative (C_1) et positive (C_2). Elle mesure combien les valeurs prédites sont proches de la valeur réelle. Dans le cas simple, pour tout exemple e on a soit $P_{C_1}(e) = 1$ et $P_{C_2}(e) = 0$, soit $P_{C_1}(e) = 0$ et $P_{C_2}(e) = 1$. Rappelons que la probabilité prédite est $P'_e(C_2)$ qui indique la probabilité que l'exemple soit dans la classe C_2 - classe positive. Évidemment, si $P_{C_1}(e) = 1$ et $P_{C_2}(e) = 0$ c'est-à-dire que l'exemple est dans la classe C_1, une petite probabilité $P'_e(C_2)$ est préférée. Dans le cas contraire, une grande probabilité $P'_e(C_2)$ est préférée. L'entropie croisée, qui doit être minimisée,

pour un exemple e est définie par :

$$\text{entropie-croisée}(e) \quad = -P_{C_1}(e) \log P'_e(C_1) - P_{C_2}(e) \log P'_e(C_2)$$

L'*entropie croisée* pour une base de test est définie comme la somme des entropies croisées de tous les exemples de la base.

$$\text{entropie-croisée}(\xi_T) = \sum_{i=1}^{N} \text{entropie-croisée}(e_i)$$

Pour rendre indépendante l'entropie croisée de la taille de la base de test, l'*entropie croisée moyenne* est définie par la somme des entropies croisées pour chaque exemple divisée par le nombre d'exemples dans la base de test.

$$\text{entropie-croisée-moyenne}(\xi_T) = \frac{\sum_{i=1}^{N} \text{entropie-croisée}(e_i)}{N}$$

La mesure de divergence dirigée de Kullback-Leibler est également utilisée. Elle mesure la distance de Kullback-Leibler entre la distribution de probabilité prédite et la distribution de probabilité réelle pour un exemple.

$$d_{KL}((P_{C_1}(e), .., P_{C_n}(e)), (P'_e(C_1), .., P'_e(C_n))) = \sum_{i=1}^{n} P_{C_i}(e) \log \frac{P_{C_i}(e)}{P'_e(C_i)}$$

Dans le cas à deux classes où la distribution de probabilité $(P_{C_1}(e), P_{C_2}(e))$ prend comme valeur l'une des deux distributions (1,0) et (0,1), la divergence dirigée se réduit à l'*entropie croisée* :

$$\begin{aligned} \text{entropie-croisée}(e) &= -P_{C_1}(e) \log P'_e(C_1) - P_{C_2}(e) \log P'_e(C_2) \\ &= P_e(C_2) \log \frac{P_e(C_2)}{P'_e(C_2)} + P_e(C_1) \log \frac{P_e(C_1)}{P'_e(C_1)} \end{aligned}$$

avec comme convention : $0 \log 0 = 0$.

Un inconvénient des mesures ci-dessus est que leurs valeurs sont infinies quand un exemple est complètement mal classifié, comme par exemple lorsque la distribution réelle est (1, 0) et la distribution prédite est (0, 1). D'ailleurs, elles ne tiennent pas compte de la distribution de probabilité a priori.

Dans [9, 10], les auteurs montrent un autre inconvénient de cette mesure. Considérons le cas non réduit à 2 classes, où chaque exemple a une seule classe réelle. Considérons un exemple e, sans perte de généralisation, on peut supposer qu'il appartient à la classe C_1. La cible est donc la distribution (1, 0, 0, ..., 0). La mesure de divergence entre cette distribution et une autre ne dépend que de sa première valeur et les autres valeurs sont totalement ignorées. Par exemple :

$$d_{KL}((1,0,0,0), (0.7, 0.1, 0.1, 0.1)) = d_{KL}((1,0,0,0), (0.7, 0.3, 0, 0))$$

Ce n'est pas une propriété intéressante car tous les éléments de la distribution devraient être pris en compte. Une fonction de mesure, intitulée IC, prenant en

paramètre une distribution de probabilité $(P'_e(C_1), .., P'_e(C_n))$ pour évaluer l'efficacité d'une classification de l'exemple e est proposée [10] et définie comme suit :

$$IC(P'_e(C_1), .., P'_e(C_n)) = P'_e(C_1) - \varepsilon \sum_{i=2}^{n} P'_e(C_i) \log P'_e(C_i)$$

où ε est suffisament petite.

Les couples $(P'_e(C_1), \sum_{i=2}^{n} P'_e(C_i) \log P'_e(C_i))$ sont ordonnés lexicographiquement.

Cependant, cette mesure n'est pas pratique à cause de la présence d'une valeur suffisament petite.

Une mesure de récompense (*information reward measure*) est proposée dans [90]. Elle est appliquée dans le cas où chaque exemple e n'appartient qu'à une seule classe $e(C)$. Dans la classification binaire, pour chaque exemple, la récompense est définie comme :

$$\text{récompense}(e) = 1 + \log P'_e(e(C))$$

Elle est de 1 si la classification est correcte $(P'_e(e(C)) = 1)$, 0 pour l'ignorance complète $(P'_e(e(C)) = 0.5)$ et elle est négative si $P'_e(e(C)) < 0.5$. Comme la distance de Kullback-Leibler, la récompense ne tient pas compte des caractéristiques des problèmes, en particulier de la distribution a priori des classes.

Par une autre approche, Kononenko et Bratko [88] ont proposé une mesure qui tient compte explicitement des probabilités a priori des classes. Cette propriété intéressante est conservée dans les mesures basées sur la capacité de discrimination qui sont présentées dans les sections suivantes. Les auteurs ont suggéré d'évaluer la quantité d'information gagnée ou perdue dans la classification de chaque exemple, puis dans la classification de tous les exemples de la base de test. A priori, la quantité d'information nécessaire pour confirmer que e est dans la classe C est : $-\log P_e(C)$. De façon analogue, la quantité d'information nécessaire pour décider correctement que e n'appartient pas à la classe C est : $-\log(1 - P_e(C))$. A posteriori, si $P'_e(e(C)) \geq P_e(e(C))$ alors la probabilité de la classe $e(C)$ change dans la « bonne direction » . On est alors en présence d'un gain d'information :

$$-\log P_e(e(C)) + \log P'_e(e(C))$$

Si $P'_e(e(C)) < P_e(e(C))$ alors la probabilité de la classe $e(C)$ change dans la « mauvaise direction » . On est alors en présence d'une perte d'information :

$$-\log(1 - P_e(e(C))) + \log(1 - P'_e(e(C)))$$

Le score final est la différence entre la quantité d'information gagnée et la quantité d'information perdue sur tous les exemples de la base de test. Il peut être normalisé en divisant par le nombre d'exemples de la base de test.

3.3 Critères basés sur des mesures de discrimination

Les algorithmes d'apprentissage essaient d'extraire le plus possible d'informations « intéressantes » de la base d'apprentissage pour construire un classifieur et l'utiliser ensuite pour classifier de nouveaux exemples. Le classifieur doit discriminer tous les exemples par leurs classes. Ainsi il induit une partition de l'ensemble des exemples. Nous proposons d'évaluer la capacité de discrimination du classifieur en analysant l'adéquation entre la partition produite par le classifieur et la partition naturelle produite par les vraies classes. Comme le critère basé sur la théorie de l'information proposé par Kononenko et Bratko [88], les critères basés sur la mesure de discrimination tiennent compte de la différence entre l'information fournie par le classifieur (a posteriori) et l'information disponible sur la base de test (a priori). Par contre, nous considérons directement la distribution de probabilité sur l'ensemble des exemples au lieu de celle de chaque individu.

Soit ξ_{TC} l'ensemble des exemples dans la base de tests ξ_T qui appartiennent à la classe C et $\xi_{TC'}$ l'ensemble des exemples dans ξ_T qui sont classifiés dans la classe C. Dans la suite de ce chapitre, pour simplifier, on enlève T dans la notation des sous-ensembles liés à la base de test.

La méthode proposée évalue l'adéquation entre deux partitions : $\{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}$ et $\{\xi_{C'_1}, \xi_{C'_2}, .., \xi_{C'_n}\}$.

Supposons que le classifieur M soit induit à partir de ξ et qu'il classifie tous les exemples de la base de test ξ_T. M étiquette chaque exemple e de ξ_T par une classe de $\mathcal{C}$. Ainsi, M introduit un nouvel attribut f_M : pour chaque exemple de ξ_T, l'attribut prend comme valeur la classe affectée à l'exemple par M (cf. tableau 3.3).

Exemple	Classe réelle	A_1	A_2	...	A_K	f_{M_1}	f_{M_2}
e_1	$e_1(C)$	v_{11} *	v_{12}	...	v_{1K}	C_{11}	C_{12}
e_2	$e_2(C)$	v_{21}	v_{22}	...	v_{2K}	C_{21}	C_{22}
...	...	...	...	...	...	...	...
e_N	$e_N(C)$	v_{N1}	v_{N1}	...	v_{NK}	C_{N1}	C_{N2}

TAB. 3.3 – Résultat de la classification par deux classifieurs : M_1 et M_2

* $v_{ik} = e_i(A_k)$ la valeur pour l'attribut A_k de l'exemple e_i.

Comme nous l'avons expliqué dans le chapitre 1 (voir aussi [107, 131]), une quantité validée par le modèle hiérarchique peut servir à mesurer la capacité de discrimination d'un attribut, en particulier le nouvel attribut f_M. Ainsi, elle peut mesurer la capacité de discrimination du classifieur M. La mesure de la capacité de discrimination de M consiste également en 3 niveaux.

Niveau $\mathcal{F}$: Le niveau $\mathcal{F}$ concerne les mesures de l'adéquation entre l'ensemble des exemples de la classe C_i et l'ensemble des exemples classifiés dans la classe C_j. Elles sont notées par : $F(\xi_{C'_j}, \xi_{C_i})$. Chacune prend sa valeur minimale quand l'ensemble des exemples classifiés dans la classe C_j est un sous-ensemble d'exemples de la classe C_i, et elle prend sa valeur maximale quand aucun exemple de la classe C_i n'est classifié dans la classe C_j.

Niveau $\mathcal{G}$: Le niveau $\mathcal{G}$ concerne les fonctions agrégeant des fonctions du niveau $\mathcal{F}$ pour mesurer la quantité d'information apportée en classifiant des exemples dans la classe C_j. Elles sont notées par : $G(\xi_{C'_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$. Chacune prend sa valeur minimale quand il existe une classe C_i satisfaisant : l'ensemble des exemples classifiés dans la classe C_j est un sous-ensemble d'ensembles des exemples de la classe C_i, autrement dit, tous les exemples classifiés dans la classe C_j n'appartiennent effectivement qu'à une seule classe C_i (par exemple dans le cas de classification idéale). Une $\mathcal{G}$-fonction prend sa valeur maximale quand les adéquations entre l'ensemble des exemples classifiés dans la classe C_j et chacun des ensembles des exemples d'une même classe sont identiques.

Niveau $\mathcal{H}$: Le niveau $\mathcal{H}$ concerne les fonctions agrégeant des fonctions du niveau $\mathcal{G}$ pour mesurer la capacité de discrimination du modèle M par rapport à des classes de $\mathcal{C}$. Elles sont notées : $H(\{\xi_{C'_1}, \xi_{C'_2}, .., \xi_{C'_n}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$.

Cette fonction mesure l'inadéquation entre les deux partitions : celle par le classifieur et celle par les classes des exemples. Plus la valeur pour la $\mathcal{H}$-fonction est petite, plus les deux partitions sont adéquates. Quand $H = 0$, les deux partitions sont identiques.

Dans la suite, nous établissons un critère d'évaluation basé sur l'entropie de Shannon, une mesure de discrimination, pour illustrer les arguments présentés ci-dessus. Notons qu'on peut évidement établir des critères basés sur d'autres mesures de discrimination que celle de Shannon.

Niveau $\mathcal{F}$:

$$F(\xi_{C'_j}, \xi_{C_i}) \quad = \quad -\log \frac{|\xi_{C_i} \cap \xi_{C'_j}|}{|\xi_{C'_j}|} = -\log P(C_i|C'_j)$$

où $p(C_i|C'_j)$ est la probabilité qu'un exemple classifié dans la classe C_j soit de la classe C_i et $|.|$ est la cardinalité d'un ensemble.

Niveau $\mathcal{G}$:

$$\begin{aligned} G(\xi_{C'_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\}) &= \sum_{i=1}^{n} \frac{|\xi_{C_i} \cap \xi_{C'_j}|}{|\xi_{C'_j}|} F(\xi_{C'_j}, \xi_{C_i}) \\ &= -\sum_{i=1}^{n} \frac{|\xi_{C_i} \cap \xi_{C'_j}|}{|\xi_{C'_j}|} \log \frac{|\xi_{C_i} \cap \xi_{C'_j}|}{|\xi_{C'_j}|} \\ &= -\sum_{i=1}^{n} P(C_i|C'_j) \log P(C_i|C'_j) \\ &= I(\xi_{C'_j}) \end{aligned}$$

C'est l'entropie par rapport aux vraies classes du sous-ensemble des exemples classifiés dans la classe C_j.

Niveau $\mathcal{H}$:

$$\begin{aligned} H(\{\xi_{C'_1}, .., \xi_{C'_n}\}, \{\xi_{C_1}, .., \xi_{C_n}\}) &= \sum_{j=1}^{n} \frac{|\xi_{C'_j}|}{|\xi_T|} G(\xi_{C'_j}, \{\xi_{C_1}, .., \xi_{C_n}\}) \\ &= \sum_{j=1}^{n} P(C'_j) I(\xi_{C'_j}) = I(\xi_T|M) \end{aligned}$$

où $P(C'_j)$ est la probabilité qu'un exemple soit classifié dans la classe C_j et $I(\xi_T|M)$ est l'entropie de la base de test conditionnée par le classifieur M.

On note : $I(\xi_T)$ l'entropie de la base de test ξ_T :

$$I(\xi_T) = I(P_1, P_2, .., P_n) = -\sum_{i=1}^{n} P_i \log P_i$$

où P_i est la probabilité qu'un exemple de ξ_T soit dans la classe C_i.

La formule suivante permet d'estimer la quantité d'information apportée par M :

$$\triangle I(M, \xi_T) = I(\xi_T) - I(\xi_T|M)$$

On a :

$$0 \leq \triangle I(M, \xi_T) \leq I(\xi_T)$$

Dans le processus d'apprentissage automatique, les algorithmes essaient de se renseigner autant que possible sur la base d'apprentissage. L'information obtenue à travers un tel processus permet de construire un classifieur. Ainsi on peut imaginer que le classifieur est un conteneur d'informations. Par conséquent, la formule ci-dessus mesure combien d'informations de la base de test sont stockées dans le classifieur. Autrement dit, elle mesure la partie de l'information nécessaire pour décrire la base de test gagnée par l'apprentissage sur les exemples de la base d'apprentissage.

Elle exprime également la différence entre l'incertitude moyenne de tous les sous-ensembles des exemples de la base de test, identifiés par le classifieur, et l'incertitude initiale de la base de test.

Le taux de gain d'information est naturellement défini comme suit :

$$\tau(M, \xi_T) = \frac{\triangle I(M, \xi_T)}{I(\xi_T)}$$

On a :

$$0\% \leq \tau(M, \xi_T) \leq 100\%$$

$\tau(M, \xi_T)$ mesure le taux de l'information de la base de test prise dans le classifieur. Évidemment, un taux élevé et un gain important d'information sont préférés.

Le taux de gain d'information décrit ci-dessus n'est pas un nouvel indice. Il est utilisé dans la transmission de données comme un indice de qualité d'un canal de transmission. Il sert également à évaluer la capacité de description d'un modèle statistique. Il est connu en statistique sous le nom de coefficient d'incertitude de Theil [148] (voir également [136, 137]). Ce coefficient caractérise la proportion de réduction de l'entropie de Shannon. À notre tour, nous exploitons la même signification pour mesurer la capacité de discrimination d'un modèle de classification plutôt que pour la capacité de description. Cette utilisation est justifiée à l'aide du modèle hiérarchique pour des mesures de discrimination. Ainsi, non seulement des mesures connues, en particulier l'entropie de Shannon et l'indice de Gini, sont envisageables mais également d'autres.

3.4 Propriétés additionnelles

Les critères basés sur des mesures de discrimination évaluent l'adéquation entre la partition naturelle des exemples et celle générée par le modèle de classification. Ils éliminent le biais sur la distribution des exemples par leurs classes. Ils constituent des critères supplémentaires à ajouter parmi les critères existants. Cependant, ils n'évaluent que la capacité de discrimination du classifieur mais n'évaluent pas l'exactitude de la classification.

Si nous considérons qu'un processus de classification est l'exécution successive d'un processus pour discriminer des exemples en un certain nombre de parties, puis d'un processus d'affectation des classes à chacune des parties, les critères proposés évaluent la première étape. Même dans le cas où l'affectation des classes n'est pas correcte (ainsi l'exactitude est faible) tandis que la discrimination est bonne, le modèle de classification n'est pas totalement inutile. Ainsi, le modèle de classification est évalué en tant que discriminateur. Ce critère n'évalue donc pas la correspondance entre les partitions et les classes.

Comme nous avons argumenté dans la section précédente, si le modèle de classification est parfait, c'est-à-dire que le taux de bonnes classifications est 100%, alors ce modèle est également parfait selon le critère de discrimination et $\tau(M, \xi_T) = 100\%$.

Dans le cas d'un classifieur qui classifie tous les exemples dans une même classe, les deux partitions sont complètement inadéquates et $\tau(M, \xi_T) = 0\%$.

Dans le cas où chaque partition générée par le modèle de classification a la même proportion de classe que dans la base de test, on a : $\tau(M, \xi_T) = 0\%$.

Comme $\tau(M, \xi_T)$ est une valeur normalisée, il peut être un indice pertinent pour comparer la performance d'un modèle de classification à travers les différentes bases de données ou pour comparer la performance de différents modèles de classification à travers une même base de données.

3.5 Extension à la classification floue

La mesure proposée peut éventuellement être étendue pour d'autres classifieurs.

Dans le cas de la classification floue, on doit évaluer le rapport entre deux partitions floues. On rencontre plus souvent le cas où chaque exemple appartient à une seule classe et le classifieur donne à chaque exemple un ensemble de degrés d'appartenance correspondant à chacune des classes. La partition par la classe est donc nette et la partition par le classifieur est floue. Dans ce cas, le modèle hiérarchique pour les mesures de discrimination floue $\mathcal{FGH}^*$ (cf. chapitre 1) peut être appliqué pour construire et valider des critères.

La fonction $F^*(\xi_{C'_j}, \xi_{C_i})$ du niveau $\mathcal{F}^*$ mesure l'adéquation entre l'ensemble flou des exemples de la classe C_i et l'ensemble flou des exemples classifiés dans la classe C_j.

La fonction $G^*(\xi_{C'_j}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$ du niveau $\mathcal{G}^*$ prenant en paramètres un sous-ensemble flou et une partition floue, est une agrégation des fonctions du niveau $\mathcal{F}^*$ pour mesurer la quantité d'information apportée en classifiant des exemples dans la classe C_j.

La fonction $H^*(\{\xi_{C'_1}, \xi_{C'_2}, .., \xi_{C'_n}\}, \{\xi_{C_1}, \xi_{C_2}, .., \xi_{C_n}\})$ du niveau $\mathcal{H}^*$ prenant en paramètres deux partitions floues, est une fonction agrégeant des fonctions du niveau $\mathcal{G}^*$ pour mesurer la capacité de discrimination du modèle M par rapport à des classes de $\mathcal{C}$.

Les propriétés de ces fonctions sont décrites dans le chapitre 1. À partir des ces fonctions, la différence entre la diversité a priori et a posteriori des exemples de la base de test peut être déduite. Cela permet d'évaluer la capacité de discrimination du modèle dans la classification floue.

3.6 Expérimentations

3.6.1 Exemple sur des données fictives

Considérons les matrices de confusion (cf. tableau 3.4) obtenues avec 3 bases de données artificielles. Chaque base contient 2 classes mais les distributions des classes sont différentes entre les 3 bases. Les taux de bonnes classifications avec ces 3 bases sont identiques : 80%. Cependant, le classifieur correspondant à la deuxième

base classifie tous les exemples dans la classe majoritaire, et ne donne donc aucune information utile. Aussi, le gain d'information dans ce cas est 0%. La première base est plus hétérogène que la troisième car 80% des exemples sont dans une même classe. Elle est donc plus difficile pour discriminer les exemples. Avec un même taux de bonnes classifications, le classifieur effectuant sur la première base est plus fort que celui effectuant sur la troisième base. L'indicateur lié à la capacité de discrimination semble cohérent avec ces remarques (cf. tableau 3.5).

	Base 1		Base 2		Base 3	
Classe prédite / Classe réelle	Pos.	Nég.	Pos.	Nég.	Pos.	Nég.
Pos.	40	10	0	20	20	10
Nég.	10	40	0	80	10	60

TAB. 3.4 – Matrices de confusion pour 3 bases de données artificielles

	Base 1	Base 2	Base 3
Taux de bonnes classifications	80.0	80.0	80.0
Entropie de la base (bit)	1	0.72	0.88
Entropie conditionnelle (bit)	0.72	0.72	0.69
Gain d'information (bit)	0.28	0.0	0.19
$\tau(\%)$	28	0	21.7

TAB. 3.5 – Résultats de classification pour 3 bases de données artificielles

3.6.2 Exemple sur des données de l'UCI

Quelques expériences ont été menées avec plusieurs bases de données artificielles et des bases de données de l'UCI [113], décrites dans le tableau 2.1 (page 61), pour illustrer et justifier les critères basés sur les mesures de discrimination. Le logiciel DTGen a été utilisé. Nous avons réalisé la même expérimentation que celle décrite dans la section 2.5.4 (page 88). Les expérimentations sont menées avec trois mesures dont deux mesures de discrimination : l'entropie conditionnelle de Shannon, l'entropie conditionnelle d'événements flous et une mesure qui n'est pas qualifiée en tant que mesure de discrimination : la mesure décrite par la formule (2.2) (page 88). Le critère d'évaluation utilisé est l'entropie de Shannon.

Le tableau 3.6 présente les résultats obtenus avec quelques bases de données de l'entrepôt de l'UCI. La deuxième colonne est l'entropie de la base de test moyennée. Les colonnes suivantes décrivent le taux de gain d'information correspondant respectivement à des méthodes de construction d'arbres utilisant l'entropie conditionnelle de Shannon, l'entropie conditionnelle d'événements flous et la mesure (2.2).

Sur chaque ligne, le meilleur taux de gain d'information est mis en gras et le pire taux est mis en italique.

Base de données	Entropie de la base (bit)	τ(%) Ent. Shannon	τ(%) Ent. floue	τ(%) Mesure (2.2, page 88)
Iris	1.58	**86.27**	*82.62*	85.02
Balance scale	1.31	39.58	**40.54**	*32.36*
E. coli	2.18	54.75	**55.18**	*49.80*
Glass identification	2.14	35.50	**37.07**	*34.64*
Ionosphere	0.94	41.49	**42.56**	*21.87*
Liver-disorders	0.98	6.07	**6.88**	*2.42*
Pima Indians diabetes	0.93	8.32	**8.62**	*3.97*
Wine recognition	1.56	**70.56**	*67.56*	70.04
Waveform	1.58	21.02	**21.36**	*20.58*

Tab. 3.6 – Résultat de classification sur les bases de données de l'UCI mesuré par le gain d'information

Avec les bases « Pima Indians diabetes » et « Liver-disorders » , les taux de gain d'information sont petits parce que la contribution des classifieurs pour discriminer les exemples est relativement petite ; ils ne font pas une différence significative par rapport à un classifieur naïf qui classifie tous les exemples dans la classe majoritaire. Le taux de bonnes classifications sur la base « E. coli » et sur la base « Balance » sont semblables mais le taux de gain d'information pour la première est nettement meilleur. Cela peut être expliqué par la plus grande complexité de la base « E. coli » par rapport à la base « Balance » (nombre de classes : 8 classes contre 3 classes, quantité d'information : 2.18 bits contre 1.31 bits). Aussi, le classifieur devrait fonctionner mieux sur la base « E. coli » pour obtenir les taux de bonnes classifications semblables. Le rapport entre les résultats sur la base « Waveform » et la base « Glass Identification » , et le rapport entre les résultats sur la base « Iris » et la base « Ionosphère » sont similaires.

Ces résultats s'accordent bien avec le résultat décrit dans le tableau 2.3 (page 89). Sur chaque base, le meilleur taux de bonnes classifications correspond bien au meilleur taux de gain d'information. Par contre, le taux de gain d'information nous permet une estimation substantielle de la capacité effective des classifieurs.

3.6.3 Caractérisation d'une base d'images

La caractérisation d'une base permet d'en estimer la complexité et, entre autres, de choisir une méthode d'exploitation adéquate. Cela facilite aussi l'analyse de l'efficacité de l'algorithme d'exploitation. Les caractéristiques simples et souvent utilisées sont des statistiques telles que le nombre de classes, le nombre d'exemples, le nombre d'attributs, la distribution des exemples. Des mesures plus sophistiquées sont également utilisées, en particuliers, l'erreur bayésienne, la complexité de la surface de séparation, la structure des classes. Quelques études sur la caratérisation de données d'apprentissage se trouvent dans [13, 73].

Dans notre travail, nous proposons d'utiliser le taux de gain d'information comme un indice caractéristique pour estimer la complexité d'une base et d'une classe. Cet indice est une fonction qui dépend de la base et d'une méthode d'exploitation.

$$\text{indice} = f(\xi, \mathcal{M})$$

où ξ est une base de données, $\mathcal{M}$ est une méthode d'exploitation. Une méthode d'exploitation est comprise dans ce contexte comme un couple formé d'un algorithme de construction d'un modèle de classification (ID3 par exemple) et un protocole d'expérimentation bien précis (validation croisée par exemple). Pour que l'indice ne dépende que de la base, il faut fixer la méthode d'exploitation à préférence une méthode simple et réprésentative d'une famille de méthode.

En particulier, la complexité d'une classe dans une base peut être estimée par τ si on réduit le problème initial à un problème à deux classes : la classe considérée contre toutes les autres.

Cette proposition a été développée dans le contexte de la recherche d'images. Nous proposons [18] de calculer un indice caractéristique pour une base d'images donnée. Cet indice est une fonction qui dépend de la base d'images, des attributs qui la caractérisent avec, entre autres, la couleur et la texture, et une méthode d'exploitation. Plus l'indice est élevé pour une base d'images, plus la base est considérée comme facile et inversement.

$$\text{indice} = f(\xi_{images}, \mathcal{A}_{images}, \mathcal{M})$$

où ξ_{images} est une base d'images, $\mathcal{A}_{images}$ est un ensemble d'attributs des images, $\mathcal{M}$ est une méthode d'exploitation.

Pour une base d'images, nous devons calculer des attributs pour chaque image et utiliser une méthode de composition de ces attributs pour produire un indice valable. Ensuite, si nous utilisons la même procédure (les mêmes attributs, la même méthode de composition) pour une nouvelle base d'images, nous espérons obtenir un nouvel indice qui permettra de comparer non seulement les deux bases d'images, mais aussi, ensuite, les résultats d'algorithmes travaillant sur ces deux bases. Pour cela, les attributs utilisés et la méthode de composition doivent être fixés et ne pas changer, sinon la comparaison des indices devient impossible.

Dans le domaine de la recherche d'images, nous avons choisi les attributs les plus fréquents pour ce domaine : la couleur et la texture. Dans les applications plus

spécifiques, telle que le traitement des images radiographiques, on peut éventuellement choisir d'autres attributs qui seraient plus adéquates. Nous avons ensuite choisi le calcul par arbres de décision comme méthode de composition de l'indice. Cette méthode est couramment utilisée car elle est simple et reproductible. D'autres méthodes de composition peuvent aussi être envisagées. Le taux de gain d'information τ est considéré comme l'indice caractéristique pour une base d'images donnée.

Une expérimentation a été menée pour montrer que dans la plupart des cas, plus cet indice est élevé, meilleur est le résultat en recherches d'images. Le lecteur est invité à consulter [18] pour les résultats détaillés.

3.7 Conclusion et perspectives

Dans ce chapitre, nous avons proposé des critères basés sur des mesures de discrimination en considérant l'adéquation entre les partitions de l'ensemble des exemples par leurs vraies classes et par le classifieur. Le taux de gain d'information décrit dans ce chapitre fournit une autre vue sur la performance d'un modèle de classification. La performance du classifieur est jugée sur sa capacité à discriminer des exemples des différentes classes. Cette mesure prend en compte la différence entre la base de test et les bases classifiées par le modèle. Elle nous donne une estimation sur la contribution effective du modèle dans la détermination des classes des exemples. Ce type de critères est justifié par le modèle hiérarchique pour les mesures de discrimination. Un ensemble d'expériences a été fait pour illustrer et justifier ces critères. Une extension de ces critères aux classifications non-classiques est décrite.

Comme un classifieur M peut être étudié en tant qu'attribut spécial f_M, cela nous suggère d'étudier la possibilité d'agréger plusieurs modèles de classification pour obtenir le meilleur modèle. Supposons que M_1, M_2, .., M_s soient induits d'une base d'apprentissage. Ces modèles de classification génèrent un ensemble d'attributs spéciaux $\{f_{M_1}, f_{M_2}, .., f_{M_s}\}$ de la base d'apprentissage. À partir de l'ensemble des attributs spéciaux, un arbre de décision peut être construit. Une validation empirique d'un tel processus d'agrégation devra être menée.

Chapitre 4

Traitement de données manquantes basé sur l'entropie

Les travaux présentés dans ce chapitre sont des travaux réalisés en commun avec Thomas Delavallade. L'état de l'art sur le traitement des données manquantes sort du cadre de cette thèse et fait partie de travaux de Thomas Delavallade [54]. Ces travaux communs ont été publiés dans [46, 55].

4.1 Motivation

Le traitement des valeurs manquantes en apprentissage automatique est un sujet qui attire de plus en plus l'intérêt des chercheurs. D'une part, plusieurs techniques d'apprentissage ne sont utilisables ou du moins ne s'avèrent efficaces que sur des données complètes. D'autre part, les résultats obtenus dépendent beaucoup de la qualité des données. Or, dans les applications réelles, la qualité des données peut ne pas être satisfaisante car souvent des données sont manquantes. La qualité des données est également altérée par la présence de données erronées, inconsistantes, difficiles (voire impossibles) à obtenir, ou simplement indisponibles aux utilisateurs. Cette problématique apparaît dans de nombreuses applications réelles : la manipulation de questionnaires avec des non-réponses, la bioinformatique [114], ainsi que la prévision de risques qui fait partie des travaux de thèse de Thomas Delavallade. Dans cette application, plus d'un quart des données sont manquantes. Ces données manquantes rendent difficile l'application des algorithmes d'apprentissage tels que les arbres de décision ou les k-plus proches voisins. Pour une manipulation efficace et valide, il faut tenir compte des données manquantes. Cela nous motive à examiner ce problème qui est commun aux statisticiens [97] et aux chercheurs travaillant dans le domaine de l'apprentissage automatique.

Nous avons décidé de nous focaliser sur les effets du traitement des données manquantes sur un problème de classification en mode supervisé pour lequel la grande majorité des classifieurs ne peut travailler qu'avec une base de données complète. Les données manquantes peuvent faire chuter les performances d'un classifieur, voire le rendre inutilisable. Nous proposons dans ce chapitre une technique pour substituer

des valeurs manquantes en appliquant la théorie de l'information.

Dans la suite, nous décrivons notre propre approche basée sur l'entropie pour remplacer des valeurs manquantes. Une série d'expérimentations est décrite pour valider et caractériser notre technique en la comparant avec les techniques existantes. Enfin, nous conclurons nos travaux. La description détaillée du traitement des données manquantes sort du cadre de cette thèse et le lecteur intéressé pourra se référer à [55, 46].

4.2 Nouvelle méthode de substitution basée sur l'entropie

Confronté à des données manquantes, plusieurs stratégies sont possibles. On peut soit développer des algorithmes qui peuvent intrinsèquement travailler en présence de données manquantes, soit réduire la base en question en éliminant des exemples possédant au moins une valeur manquante ou en éliminant des attributs contenant ce type de valeurs, soit chercher un moyen adéquat pour remplir les valeurs manquantes. La dernière stratégie constitue alors une étape de pré-traitement avant le lancement de l'algorithme principal d'analyse ou de fouille de données. Elle semble la plus utilisée et notre approche se place dans cette perspective.

Dans cette étude, nous considérons que les données manquantes surviennent de manière complètement aléatoire, ne dépendant d'aucune variable.

Notre objectif est de construire un bon classifieur à partir de données incomplètes. Pour cela, nous proposons une nouvelle méthode de substitution des données manquantes basée sur l'entropie. Contrairement au but poursuivi par les méthodes usuelles, il nous importe peu que la valeur de remplacement soit proche de la valeur réelle que l'on ne connaît pas ou que la distribution des données obtenue soit proche de la distribution initiale. Dans ce contexte, notre objectif est plus pragmatique.

4.2.1 Principe

En classification, on emploie les valeurs connues des attributs décrivant un exemple pour identifier la classe à laquelle il appartient. On doit donc considérer la relation entre un attribut et la classe. Plus cette relation est forte, meilleur est l'attribut. Un bon attribut est celui qui caractérise bien les différentes classes. Une mesure de discrimination, en particulier une de celles issues de la théorie de l'information, est utilisée pour évaluer cette capacité. Si l'entropie de Shannon est utilisée, elle évalue la quantité d'information sur la classe des exemples, possédée par l'attribut en question. Notre hypothèse est que le manque de données, relativement à un attribut, détériore sa capacité de discrimination. En conséquence nous proposons une méthode pour remplir les données manquantes, attribut par attribut, afin de leur restaurer, dans la mesure du possible, cette capacité.

Rappelons que $\xi = \{e_1, e_2..., e_N\}$ est un ensemble d'exemples, constituant la base d'apprentissage. Supposons que la base ait un certain nombre d'exemples ayant des

valeurs d'attributs manquantes. Considérons l'attribut symbolique A qui prend ses valeurs dans l'ensemble $\{v_1, v_2, .., v_m\}$.

Notons ξ_A^{manq} le sous-ensemble des exemples de ξ dont la valeur pour A est manquante et ξ_A^{obs} le sous-ensemble des exemples de ξ dont la valeur pour A est observée.

$$\xi_A^{manq} = \{e_i | e_i \in \xi \text{ et la valeur de } e_i(A) \text{ est inconnue}\}$$

$$\xi_A^{obs} = \{e_i | e_i \in \xi \text{ et la valeur de } e_i(A) \text{ est connue}\}$$

Pour simplifier, dans ce qui suit où il n'y a pas d'ambiguïté, on ignore l'indice A dans ξ_A^{manq} et dans ξ_A^{obs}.

On a :

$$\xi = \xi^{obs} \cup \xi^{manq} \text{ et } \xi^{obs} \cap \xi^{manq} = \emptyset$$

Une substitution s pour A est une application définie comme suit :

$$\begin{cases} s : \xi^{manq} & \longrightarrow \{v_1, v_2, .., v_m\} \\ e_i & \longmapsto s(e_i) \end{cases}$$

Dans ce cas, A et s permettent de définir un nouvel attribut A_s pour les exemples de ξ, tel que :

$$e_i(A_s) = \begin{cases} e_i(A) & \text{si } e_i \in \xi^{obs} \\ s(e_i) & \text{si } e_i \in \xi^{manq} \end{cases} \tag{4.1}$$

A_s est donc un attribut pour lequel il n'y a aucune valeur manquante pour les exemples de ξ.

Notons S l'ensemble des substitutions possibles. Comme il y a m possibilités de remplacement pour une valeur manquante, la cardinalité de S est donc :

$$|S| = m^{|\xi^{manq}|} \tag{4.2}$$

où m est le nombre de valeurs pour A et $|\xi^{manq}|$ est le nombre d'exemples dans ξ^{manq}.

Nous proposons alors de remplir les valeurs manquantes des exemples de ξ^{manq} pour l'attribut A afin d'obtenir l'attribut A_s qui apporte le plus d'information sur les classes des exemples dans ξ.

Soit A et s, on note :

$$\triangle I(A_s, \xi) = I(\xi) - I(\xi | A_s)$$

La meilleure substitution $s_{optimale}$ de S est celle qui maximise $\triangle I(A_s, \xi)$. De plus, comme l'entropie $I(\xi)$ est indépendante de s, $s_{optimale}$ est celle qui minimise $I(\xi | A_s)$.

Substitution des données numériques

Le principe décrit ci-dessus s'applique aux données symboliques. Il peut être adapté pour des données numériques. Dans ce cas, il suffit de les discrétiser. Un intervalle est donc affecté à une valeur manquante. Si une valeur numérique est requise, une valeur centrale de l'intervalle peut être utilisée. C'est souvent une valeur centrale de l'intervalle ou une valeur tirée aléatoirement selon une distribution spécifique. Dans la suite, nous étudions le cas des attributs symboliques.

On peut démontrer maintenant les propositions suivantes :

Proposition 4.2.1. *Dans la solution optimale, toutes les valeurs manquantes d'un attribut correspondant à des exemples d'une même classe sont substituées par une même valeur.*

Cette proposition semble intuitive car pour remplir des valeurs manquantes d'un attribut, on utilise uniquement les valeurs existantes et les classes. Donc toutes les valeurs manquantes correspondant à une même classe doivent être traitées d'une même manière. Cela permet de construire des modèles qui auront tendance à mieux généraliser les données.

Cette proposition permet ainsi de réduire éventuellement le nombre de possibilités de substitution à :

$$|S_{reduit}| = m^n \tag{4.3}$$

où n est le nombre de classes et m est le nombre de valeurs possibles pour A. C'est significatif dans le cas ou le nombre de valeurs manquantes est plus grand que le nombre de classes.

Démonstration. Considérons une base d'apprentissage pour laquelle d_k exemples de la classe C_k ont des valeurs manquantes pour un attribut A. Le tableau 4.1 décrit la partie observée de la base pour les valeurs de l'attribut A. N_{ij} est le nombre d'exemples de la classe C_i ayant $A = v_j$. $N_{i.}$ est le nombre d'exemples de la classe C_i et $N_{.j}$ est le nombre d'exemples ayant $A = v_j$. Le tableau 4.2 décrit la base dont toutes les valeurs manquantes de la classe C_k sont substituées. d_{kj} est le nombre de valeurs manquantes de C_k substituées par la valeur v_j. Pour simplifier la notation, on note : $E_j = I(\xi^{obs}|A = v_j)$ et $E'_j = I(\xi|A = v_j)$.

On a :

$$\sum_{i=1}^{n}\sum_{j=1}^{m} N_{ij} = N \quad ; \quad N_{i.} = \sum_{j=1}^{m} N_{ij} \quad ; \quad N_{.j} = \sum_{i=1}^{n} N_{ij}$$

$$\sum_{i=1}^{n} N_{i.} = \sum_{j=1}^{m} N_{.j} = N \quad ; \quad \sum_{j=1}^{m} d_{kj} = d_k$$

On démontre que lorsque l'entropie conditionnelle $I(\xi|A_s)$ est minimale, il existe $j \in \{1, 2, .., m\}$ tel que $d_{kj} = d_k$ et $\forall l \in \{1, 2, .., m\}$, $l \neq j : d_{kl} = 0$.

On définit :

$$E_j = -\sum_{i=1}^{n} \frac{N_{ij}}{N_{.j}} \log \frac{N_{ij}}{N_{.j}} = \frac{1}{N_{.j}} \left(N_{.j} \log N_{.j} - \sum_{i=1}^{n} N_{ij} \log N_{ij} \right) \tag{4.4}$$

Et :

$$I(\xi^{obs}|A) = \sum_{j=1}^{m} \frac{N_{.j}}{N} E_j = \frac{1}{N} \sum_{j=1}^{m} \left(N_{.j} \log N_{.j} - \sum_{i=1}^{n} N_{ij} \log N_{ij} \right)$$

Valeur / Classe	v_1	...	v_j	...	v_m	$\sum$
C_1	N_{11}	...	N_{1j}	...	N_{1m}	$N_{1.}$
...	...	...	...	...	...	...
C_k	N_{k1}	...	N_{kj}	...	N_{km}	$N_{k.}$
...	...	...	...	...	...	...
C_n	N_{n1}	...	N_{nj}	...	N_{nm}	$N_{n.}$
$\sum$	$N_{.1}$	...	$N_{.j}$	...	$N_{.m}$	N
Entropie	E_1	...	E_j	...	E_m	

TAB. 4.1 – Distribution et entropie de la partie observée ξ^{obs} correspondant à l'attribut A

Valeur / Classe	v_1	...	v_j	...	v_m	$\sum$
C_1	N_{11}	...	N_{1j}	...	N_{1m}	$N_{1.}$
...	...	...	...	...	...	...
C_k	$N_{k1} + d_{k1}$	...	$N_{kj} + d_{kj}$	...	$N_{km} + d_{km}$	$N_{k.} + d_k$
...	...	...	...	...	...	...
C_n	N_{n1}	...	N_{nj}	...	N_{nm}	$N_{n.}$
$\sum$	$N_{.1} + d_{k1}$	...	$N_{.j} + d_{kj}$	...	$N_{.m} + d_{km}$	$N + d_k$
Entropie	$E_1^{'}$	...	$E_j^{'}$	...	$E_m^{'}$	

TAB. 4.2 – Distribution et entropie de la base dans laquelle les valeurs manquantes de l'attribut A correspondant à la classe C_k sont substituées

Selon ce principe, il nous faut chercher les valeurs $d_{k1}, d_{k2}, .., d_{km}$ tel que $I(\xi|A_s)$

est minimale. En appliquant la formule ci-dessus au tableau 4.2 on a :

$$\begin{aligned} I(\xi|A_s) &= \sum_{j=1}^{m} \frac{N_{.j}+d_{kj}}{N+d_k} E_j^{'} \\ &= \frac{1}{N+d_k}\left(\sum_{j=1}^{m}(N_{.j}+d_{kj})\log(N_{.j}+d_{kj}) - \sum_{j=1}^{m}\sum_{i=1,i\neq k}^{n} N_{ij}\log N_{ij} \right. \\ &\qquad \left. - \sum_{j=1}^{m}(N_{kj}+d_{kj})\log(N_{kj}+d_{kj})\right) \end{aligned}$$

Il faut donc minimiser la fonction suivante dont les variables sont les nombres de valeurs manquantes d_{kj} avec la contrainte $\sum_{j=1}^{m} d_{kj} = d_k$:

$$\begin{aligned} Q(d_{k1}, d_{k2}, .., d_{km}) &= \sum_{j=1}^{m}((N_{.j}+d_{kj})\log(N_{.j}+d_{kj}) - (N_{kj}+d_{kj})\log(N_{kj}+d_{kj})) \\ &= \sum_{j=1}^{m} q_j(d_{kj}) \end{aligned}$$

où :

$$q_j(x) = (N_{.j}+x)\log(N_{.j}+x) - (N_{kj}+x)\log(N_{kj}+x)$$

Calculons les dérivées première $q_j^{'}$ et seconde $q_j^{''}$ de q_j par rapport à x :

$$q_j^{'}(x) = \log(N_{.j}+x) - \log(N_{ij}+x) \geq 0$$

$$q_j^{''}(x) = \log e\left(\frac{1}{N_{.j}+x} - \frac{1}{N_{ij}+x}\right) \leq 0$$

Donc la fonction multivariable $Q(d_{k1}, d_{k2}, .., d_{km})$ est la somme de fonctions monovariables continues, deux fois dérivables, croissantes et concaves $q_j(x)$ quand $x \geq 0$. L'espace de définition pour Q est décrit par $\sum_{j=1}^{m} d_{kj} = d_k$ où $d_{kj} \geq 0$, qui est donc borné et convexe. Selon les propriétés de cette catégorie de fonctions, sa valeur minimale est atteinte à l'une des bornes. C'est un point décrit par $d_{kj} = d_k$ et $d_{kl} = 0$, $\forall l \neq j$ avec un $j \in \{1, 2, .., m\}$. □

Proposition 4.2.2. *Pour les données manquantes de la classe C_i, le principe proposé favorise la valeur v_j qui maximise $P(C_i|v_j)$ dans la base complétée.*

La démonstration de cette proposition se fait en calculant l'entropie conditionnelle de la base sans tenir compte des valeurs manquantes, ainsi que l'entropie conditionnelle de la base lorsque l'on considère d exemples ayant des valeurs manquantes sur l'attribut A. Sous l'hypothèse que dans la base remplie le nombre d'exemples de la classe C_i ayant v_j comme valeur pour A soit grand, on peut simplifier les calculs pour arriver à la conclusion. Cette proposition permet éventuellement de simplifier la réalisation de la méthode sous certaines conditions. En particulier ce résultat peut être appliqué comme initialisation de l'algorithme itératif décrit dans la section suivante.

Démonstration. Considérons une substitution pour un attribut A. Supposons que le nombre de valeurs manquantes de la classe C_i substituées par v_j soit d_{ij}. Notons $d_{.j}$ le nombre de valeurs manquantes substituées par v_j, et d_i le nombre de valeurs manquantes pour la classe C_i, d est le nombre de valeurs manquantes. Selon la proposition 4.2.1, les d_i valeurs manquantes correspondant à la classe C_i sont substituées par une même valeur. Cela signifie que parmi les $d_{i1}, d_{i2}, .., d_{im}$, il y a au maximum une valeur égale à d_i et les autres sont nulles.

On a :

$$\begin{aligned} I(\xi|A_s) &= \sum_{j=1}^{m} \frac{N_{.j}+d_{.j}}{N+d} E_j^{'} \\ &= \sum_{j=1}^{m} \left(\frac{N_{.j}+d_{.j}}{N+d} E_j^{'} - \frac{N_{.j}}{N+d} E_j \right) + \frac{N}{N+d} \sum_{j=1}^{m} \frac{N_{.j}}{N} E_j \\ &= \sum_{j=1}^{m} \left(\frac{N_{.j}+d_{.j}}{N+d} E_j^{'} - \frac{N_{.j}}{N+d} E_j \right) + \frac{N}{N+d} I(\xi^{obs}|A) \end{aligned}$$

Comme $I(\xi|A)$ et N ne varient pas, selon la méthode proposée, on doit donc choisir parmi les valeurs possibles $v_1, v_2, .., v_m$ une valeur qui minimise :

$$\sum_{j=1}^{m} ((N_{.j}+d_{.j})E_j^{'} - N_{.j}E_j)$$

À partir de l'équation (4.4), on a :

$$\begin{aligned} (N_{.j}+d_{.j})E_j^{'} - N_{.j}E_j &= (N_{.j}+d_{.j}) \log(N_{.j}+d_{.j}) - \sum_{i=1}^{n} (N_{ij}+d_{ij}) \log(N_{ij}+d_{ij}) \\ &- \left(N_{.j} \log N_{.j} - \sum_{i=1}^{n} N_{ij} \log N_{ij} \right) \\ &= N_{.j} \log \frac{N_{.j}+d_{.j}}{N_{.j}} + d_{.j} \log(N_{.j}+d_{.j}) \\ &- \sum_{i=1}^{n} \left(N_{ij} \log \frac{N_{ij}+d_{ij}}{N_{ij}} + d_{ij} \log(N_{ij}+d_{ij}) \right) \\ &= \left(N_{.j} \log \frac{N_{.j}+d_{.j}}{N_{.j}} - \sum_{i=1}^{n} N_{ij} \log \frac{N_{ij}+d_{ij}}{N_{ij}} \right) \\ &- \sum_{i=1}^{n} \left(d_{ij} \log \frac{N_{ij}+d_{ij}}{N_{.j}+d_{.j}} \right) \end{aligned} \tag{4.5}$$

Or on a :

$$\lim_{x \to +\infty} x \log \left(1 + \frac{d}{x} \right) = d \log e$$

Donc :

$$\lim_{N_{.j}\to+\infty} N_{.j} \log \frac{N_{.j} + d_{.j}}{N_{.j}} = d_{.j} \log e$$

et

$$\lim_{N_{ij}\to+\infty} N_{ij} \log \frac{N_{ij} + d_{ij}}{N_{ij}} = d_{ij} \log e$$

Or :

$$d_{.j} = \sum_{i=1}^{n} d_{ij}$$

En passage à la limite (pour $N_{.j}$ et N_{ij} suffisamment grands par rapport à d_{ij}) on peut éliminer la première différence dans (4.5) :

$$\lim_{N_{ij}\to+\infty} \left(N_{.j} \log \frac{N_{.j} + d_{.j}}{N_{.j}} - \sum_{i=1}^{n} N_{ij} \log \frac{N_{ij} + d_{ij}}{N_{ij}} \right) = 0$$

Dans ce cas, il nous faut maximiser :

$$\sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} \log \frac{N_{ij} + d_{ij}}{N_{.j} + d_{.j}}$$

Selon la proposition 4.2.1, soit $d_{ij} = 0$, soit $d_{ij} = d_i$ (c'est-à-dire v_j est la valeur de substitution pour toutes les valeurs manquantes de la classe C_i). Dans le cas où $d_{ij} = d_i > 0$, il faut donc maximiser :

$$P(C_i|v_j) = \frac{N_{ij} + d_{ij}}{N_{.j} + d_{.j}}$$

C'est la probabilité de la classe C_i conditionnellement à la valeur v_j dans la base avec des valeurs substituées. □

À noter que l'analyse faite dans cette section n'est valide que sous la contrainte sur la grandeur des N_{ij}. Dans le contexte de la classification par arbre de décision, la maximisation de $P(C_i|v_j)$ est justifiée par le fait qu'un exemple e dont la valeur sur l'attribut en question est inconnue, serait dirigé vers le nœud $\mathcal{N}$ dont la probabilité de la classe de e dans $\mathcal{N}$ est la plus grande possible.

4.2.2 Algorithme

Nous étudions dans la suite les algorithmes permettant de mettre en pratique le principe décrit ci-dessus. Ces algorithmes sont issus de travaux communs avec Thomas Delavallade [46, 55].

Le premier algorithme est exhaustif. Il consiste à évaluer toutes les substitutions possibles pour choisir la meilleure substitution selon le principe décrit dans la section 4.2.1. La solution obtenue respecte parfaitement le principe proposé. Dans ce cas,

le nombre de possibilités à évaluer est exponentiel selon le nombre de valeurs manquantes (selon (4.2)) ou le nombre de classes (selon (4.3)). Ceci rend cet algorithme coûteux surtout dans les cas où le nombre de données manquantes, le nombre de classes et le nombre de valeurs de remplacement possibles sont grands.

Pour surmonter cette difficulté, nous proposons de sacrifier la perfection du résultat obtenu par la solution exhaustive pour construire des algorithmes moins coûteux qui réalisent approximativement le principe proposé. Nous décrivons ci-dessous deux algorithmes. Le premier est simple, non-itératif et le deuxième est itératif. Évidemment, la deuxième se rapproche mieux du principe initial au prix d'un accroissement de la complexité.

L'algorithme non-itératif traite les valeurs manquantes une à une. Chacune est substituée comme si elle était la seule à traiter. Pour chaque exemple e de ξ^{manq}, on procède en deux étapes :

1. Pour chaque valeur v_j dans l'ensemble des valeurs possibles $\{v_1, v_2, .., v_m\}$ de A, calculer l'entropie conditionnelle de $\xi^{obs} \cup \{e\}$ conditionnelle à l'attribut A, en supposant que la valeur de e pour A est observée et $e(A) = v_j$:

$$I(\xi^{obs} \cup \{e\} | A \text{ avec } e(A) = v_j)$$

2. Choisir l'entropie conditionnelle la plus petite et affecter à $e(A)$ la valeur v associée à cette entropie conditionnelle minimale :

$$v = \arg \min_{v_j \in \{v_1, v_2, .., v_m\}} I(\xi^{obs} \cup \{e\} | A \text{ avec } e(A) = v_j)$$

Cette version remplit chaque valeur manquante en cherchant à minimiser une entropie conditionnelle liée à une seule valeur manquante en question. Cependant, cela n'assure pas la minimisation de l'entropie conditionnelle selon le principe décrit dans la section 4.2.1. Notons que la sortie de l'algorithme est indépendante de l'ordre dans lequel les exemples sont traités. Pour un attribut, comme toutes les valeurs manquantes des exemples d'une même classe sont remplacées par une même valeur, on peut donc calculer une fois et reprendre le résultat pour plusieurs valeurs manquantes. Toutes les combinaisons possibles entre valeurs d'attributs et classes d'exemples sont évaluées. La complexité de l'algorithme est donc linéaire par rapport au produit du nombre de valeurs manquantes et du nombre de classes : $\mathcal{O}(\min\{|\xi^{manq}|, n\} * m)$.

L'algorithme itératif permet de se rapprocher du principe proposé. À la première itération, les valeurs manquantes sont estimées d'une manière simple, qui peut être celle de l'algorithme non-itératif sur des données initiales. Ensuite, pour toutes les itérations suivantes, on recalcule toutes les valeurs manquantes, mais cette fois-ci, chaque valeur est calculée en supposant que toutes les autres valeurs de substitution estimées dans l'itération précédente sont des valeurs observées. L'algorithme s'arrête lorsque l'entropie conditionnelle ne décroît plus significativement. Cet algorithme n'assure qu'une solution optimale locale. Le nombre de calculs d'entropie est cette fois de l'ordre de : $\mathcal{O}(\min\{|\xi^{manq}|, n)\} * mL)$ où L est le nombre d'itérations effectuées.

Exemple : L'exemple suivant illustre la réalisation de l'algorithme sur une base de 10 exemples $\xi = \{e_1, e_2, .., e_{10}\}$, et un attribut A possèdant trois valeurs notées v_1, v_2 et v_3. Pour l'attribut A il y a deux données manquantes. Pour simplifier la notation dans les tableaux 4.3 et 4.4, on note $x_i = e_i(A)$, la valeur pour l'attribut A de l'exemple e_i. On a : $\xi^{obs} = \{e_1, e_2, e_3, e_4, e_5, e_7, e_8, e_{10}\}$ et $\xi^{manq} = \{e_6, e_9\}$. La tâche est de trouver les valeurs pour x_6 et x_9.

TAB. 4.3 – Base initiale avec des données manquantes pour e_6 et e_9

ξ	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
A	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Valeur	v_1	v_2	v_1	v_3	v_1	?	v_2	v_2	?	v_1
Classe	C_1	C_1	C_2	C_2	C_1	C_1	C_2	C_2	C_2	C_1

$$
\begin{array}{lcll}
x_6 = v_1 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6\} | A \text{ avec } x_6 = v_1\right) & = & -\frac{5}{9}\left(\frac{4}{5}\log\frac{4}{5} + \frac{1}{5}\log\frac{1}{5}\right) \\
 & & & & -\frac{3}{9}\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right) \\
 & & & & -\frac{1}{9}\left(\frac{1}{1}\log\frac{1}{1} + \frac{0}{1}\log\frac{0}{1}\right) \\
 & & & = & \underline{0.707} \\
x_6 = v_2 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6\} | A \text{ avec } x_6 = v_2\right) & = & 0.805 \\
x_6 = v_3 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6\} | A \text{ avec } x_6 = v_3\right) & = & 0.888 \\
\hline
 & \Rightarrow & \hat{x}_6 = v_1 & & \\
\hline
x_9 = v_1 & \Rightarrow & I\left(\xi^{obs} \cup \{e_9\} | A \text{ avec } x_9 = v_1\right) & = & 0.846 \\
x_9 = v_2 & \Rightarrow & I\left(\xi^{obs} \cup \{e_9\} | A \text{ avec } x_9 = v_2\right) & = & 0.721 \\
x_9 = v_3 & \Rightarrow & I\left(\xi^{obs} \cup \{e_9\} | A \text{ avec } x_9 = v_3\right) & = & \underline{0.666} \\
\hline
 & \Rightarrow & \hat{x}_9 = v_3 & &
\end{array}
$$

TAB. 4.4 – Deuxième et dernière itération (les valeurs de $\hat{x}_6$ et $\hat{x}_9$ ne changent pas)

ξ	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
A	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Valeur	v_1	v_2	v_1	v_3	v_1	?	v_2	v_2	?	v_1
Classe	C_1	C_1	C_2	C_2	C_1	C_1	C_2	C_2	C_2	C_1

$$
\begin{array}{lcll}
x_6 = v_1 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6, e_9\} | A \text{ avec } x_9 = v_3 \text{ et } x_6 = v_1\right) & = & -\frac{5}{10}\left(\frac{4}{5}\log\frac{4}{5} + \frac{1}{5}\log\frac{1}{5}\right) \\
 & & & & -\frac{3}{10}\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right) \\
 & & & & -\frac{2}{10}\left(\frac{2}{2}\log\frac{2}{2} + \frac{0}{2}\log\frac{0}{2}\right) \\
 & & & = & \underline{0.636} \\
x_6 = v_2 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6, e_9\} | A \text{ avec } x_9 = v_3 \text{ et } x_6 = v_2\right) & = & 0.724 \\
x_6 = v_3 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6, e_9\} | A \text{ avec } x_9 = v_3 \text{ et } x_6 = v_3\right) & = & 0.875 \\
\hline
 & \Rightarrow & \hat{x}_6 = v_1 & & \\
\hline
x_9 = v_1 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6, e_9\} | A \text{ avec } x_6 = v_1 \text{ et } x_9 = v_1\right) & = & 0.826 \\
x_9 = v_2 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6, e_9\} | A \text{ avec } x_6 = v_1 \text{ et } x_9 = v_2\right) & = & 0.685 \\
x_9 = v_3 & \Rightarrow & I\left(\xi^{obs} \cup \{e_6, e_9\} | A \text{ avec } x_6 = v_1 \text{ et } x_9 = v_3\right) & = & \underline{0.636} \\
\hline
 & \Rightarrow & \hat{x}_9 = v_3 & &
\end{array}
$$

Le tableau 4.3 illustre le comportement de l'algorithme non-itératif. Le nombre de calculs d'entropie est ici linéaire par rapport au nombre de données manquantes ou bien du nombre de classes.

Le tableau 4.4 illustre, sur le même exemple ce qui est fait lors de la deuxième itération (la première itération est réalisée à l'aide de l'algorithme non-itératif). La deuxième itération est la dernière car les valeurs de substitution pour x_6 et x_9 ne changent pas et en conséquence l'entropie conditionnelle ne diminue pas.

4.3 Expérimentations

Pour compléter l'étude présentée ci-dessus, un certain nombre d'expérimentations ont été menées dans le contexte de la classification supervisée. Ces expérimentations ont été faites en commun avec Thomas Delavallade [46, 55].

Nous préférons désormais voir comment notre méthode se comporte de manière empirique. Au travers d'expériences sur des données artificielles et réelles, nous souhaitons d'une part identifier les conditions qui lui sont les plus favorables et d'autre part juger de sa qualité comparativement aux techniques existantes pour justifier l'intérêt qu'il peut y avoir à l'utiliser. Cela permet d'avoir une idée pour choisir la technique la plus adéquate relativement à un problème concret.

Dans le contexte de la classification supervisée, la performance des techniques de substitution n'est évaluée ni sur la proximité entre les valeurs de substitution et les valeurs réelles, ni sur le respect de la distribution de certaines statistiques. Évaluer une technique de substitution reviendra à évaluer le classifieur construit sur la base que cette technique aura complétée. Parmi les diverses mesures d'évaluation des classifieurs, nous avons choisi le taux de bonnes classifications qui est le seul utilisé dans les différentes études comparatives sur le sujet qui nous ont servi de référence [4, 14, 69].

4.3.1 Protocole des expérimentations

Pour mieux gérer les paramètres, nous avons mené des expériences sur les bases complètes. Sur ces bases, les valeurs manquantes sont artificiellement générées en enlevant aléatoirement certaines valeurs. Soit une base de données complète. Le protocole a été conçu comme suit [55, 46] :

1. **Génération de bases de données** : Créer 10 paires de bases apprentissage-test selon la procédure de la validation croisée : découper aléatoirement la base en 10 parties de telle façon que les taux d'une même classe dans toutes les parties soient identiques. À chaque fois, garder une partie pour la base de test, regrouper les 9 autres parties pour former une base d'apprentissage.
2. **Génération artificielle de données manquantes** : Pour chaque paire de bases apprentissage-test, « trouer » la base d'apprentissage selon l'hypothèse que les données manquantes surviennent de manière complètement aléatoire,

ne dépendant d'aucune variable. Nous testons avec plusieurs taux de données manquantes pour les bases d'apprentissage (10%, 20%, 30%, 40% et 50%).

3. **Substitution des données manquantes** : Substituer les données manquantes dans les bases d'apprentissage par une technique de substitution (supervisée ou non-supervisée).
4. **Apprentissage** : Pour chaque paire de bases apprentissage-test ainsi remplies :
   (a) Appliquer un algorithme d'apprentissage sur la base d'apprentissage afin de construire un modèle de classification. Nous avons appliqué les algorithmes suivants : $k-$plus proches voisins (IB1), arbre de décision C4.5 (J48), réseau bayésien naïf (NB). Ils sont implémentés dans Weka-3.4.7 [161]. Cela permet d'examiner l'influence de la technique de substitution sur les performances des différents algorithmes d'apprentissage
   (b) Utiliser le modèle obtenu pour classifier les exemples de la base de test. Comme notre objectif est d'optimiser le modèle obtenu, en particulier sur le taux de bonnes classifications, nous évaluons les résultats de classification obtenus.
5. **Agrégation des résultats** : Moyenner les indices de performance sur toutes les paires de bases apprentissage-test.

Nous comparons la méthode proposée avec les méthodes suivantes :

1. Les substitutions basées sur une mesure de tendance centrale :
   (a) *Moyenne*, *Médiane* et *Mode* : Les valeurs manquantes de chaque variable sont remplacées par la moyenne arithmétique ou la médiane (pour être moins sensible aux valeurs aberrantes) pour les données numériques et le mode pour les données symboliques.
   (b) *CMoyenne*, *CMédiane*, *CMode* : Les valeurs de substitution sont des mesures de tendance calculées pour les exemples d'une même classe.
   (c) *MoyenneS*, *CMoyenneS* : Les valeurs de substitution sont tirées aléatoirement (sous certaines contraintes) aux alentours des valeurs moyennes pour tenir compte de l'incertitude liée au processus de substitution.

   Comme il y a plusieurs méthodes dans la famille, nous ne présentons que la comparaison avec les meilleurs méthodes de la famille : *Mode* et *CMode* pour les données symboliques, *Moyenne* et *CMoyenneS* pour les données numériques.
2. *Aléatoire* : Les valeurs de substitution sont tirées aléatoirement dans le domaine de définition de la variable considérée.
3. k-plus proches voisins :
   (a) *kppv* : remplacer la valeur manquante par la moyenne des valeurs du même attribut pour les k-plus proches voisins de l'exemple contenant la valeur manquante en question. Dans le cas de variables symboliques, on utilise la valeur majoritaire (le mode) des voisins.

(b) *kppvI* : une extension de *kppv* qui effectue le remplacement de manière itérative jusqu'à ce que les valeurs de substitution ne changent plus significativement.

4. La variable contenant des données manquantes est considérée comme une variable à prédire à partir des valeurs prises par les autres variables d'un même exemple.
   (a) *LLS* (*Local Least Square*) : C'est une technique de régression itérative et locale : seuls les k-plus proches voisins de l'attribut que l'on cherche à prédire sont retenus [84]. D'abord toutes les valeurs manquantes sont substituées par une méthode simple comme la *moyenne*, puis les k-plus proches voisins de l'attribut en question sont choisis et utilisés comme variables indépendantes dans un modèle de régression linéaire. Cette étape est répétée jusqu'à ce que les valeurs substituées ne changent plus significativement.
   (b) *classifieur J48*, *classifieur IB1* et *classifieur NB* : L'attribut en question est considéré temporairement comme la classe et elle est prédite en utilisant d'autres attributs. Si ces derniers sont numériques, ils serons discrétisés par les méthodes *EW* (*equal width*), *EF* (*equal frequency*) et la discrétisation basée sur l'entropie *ID3* (discrétisation binaire, récursive, visant à maximiser le gain d'information) (cf. chapitre 2, page 69). Nous utilisons les méthodes de classification implémentées dans Weka [161] : J48, IB1 et NB. La valeur numérique est alors obtenue par un même processus que pour la méthode proposée (voir page 120). Le nom des méthodes de discrétisation précède le nom du classifieur dans le tableau de résultats.

4.3.2 Méthode d'analyse des résultats

La méthode d'analyse des résultats a été étudiée et développée dans [46, 55].

Nous souhaitons comparer les performances des différentes méthodes de substitution. Notre objectif étant d'obtenir les meilleurs classifieurs possibles, nous comparons en fait les performances des classifieurs construits sur les bases de données complétées par les différentes méthodes de substitution.

Comme tous nos résultats sont obtenus sur des ensembles finis d'exemples et non sur la population entière des exemples, nous devons nous appuyons sur une certaine méthodologie statistique pour nous assurer que nos conclusions sont suffisamment fiables. Nous voulons garantir pour une probabilité d'erreur fixée que les différences observées reflètent des différences effectives entre les méthodes et qu'elles ne sont pas dues au processus d'échantillonnage.

Comme dans [55], nous avons adopté le test d'analyse de variance de Friedman, un équivalent non paramétrique d'ANOVA (*ANalysis Of VAriance*) car il ne fait aucune hypothèse sur la forme des distributions sous-jacentes. En fait le test de Friedman consiste à appliquer l'ANOVA sur les rangs des performances des algorithmes, plutôt que directement sur leurs indices de performances.

Quand l'hypothèse nulle, qui confirme qu'il n'y a aucune différence significative entre les méthodes, est rejetée, nous devons réaliser les tests post-hoc pour savoir quels algorithmes se différencient. Suivant [56, 55], nous utilisons le test de Nemenyi quand nous souhaitons comparer toutes les paires d'algorithmes. C'est l'équivalent non paramétrique du test de Tukey.

Quand un algorithme sert de référence pour comparer avec tous les autres, nous nous appuyons sur la procédure de test de Bonferroni-Dunn modifiée, appelée procédure de Holland-Copenhaver, dénotée ci-après HC-SU.

4.3.3 Description des données

Les expérimentations ont été menées sur 13 bases de données de l'UCI qui se divisent en deux catégories : 5 bases de données symboliques et 8 bases de données numériques. Les principales caractéristiques (nombre d'attributs, nombre d'exemples, nombre de classes, numérique/symbolique) de chacune des bases sont décrites dans le tableau 4.5.

4.3.4 Résultats expérimentaux et discussions

La substitution des données manquantes d'un attribut numérique doit être réalisée après une phase de discrétisation. Ainsi, elle possède des propriétés spécifiques. Pour cela nous présentons donc séparément les expérimentations sur les bases de données symboliques et les bases de données numériques.

4.3.4.1 Bases de données symboliques

Les résultats des tests statistiques pour les classifieurs et les méthodes de substitution sont présentés dans le tableau 4.6.

Les résultats obtenus avec un même taux de données manquantes et avec une même technique de classification, mais avec différentes techniques de substitution sont triés (de 1 à 7). Les rangs, correspondant à une même technique de classification et à une même technique de substitution, sont moyennés sur plusieurs tests et décrits dans le tableau 4.6.

Chaque colonne correspond à un algorithme de classification (J48, IB1, NB). Parmi les rangs dans une même colonne, plus le rang d'une méthode de substitution est petit, meilleure est la méthode. Nous avons comparé notre nouvelle technique aux techniques de substitution existantes. Nous avons utilisé les tests post-hoc HC-SU évoqués ci-dessus pour évaluer statistiquement les différences observées quand le test de Friedman conclut qu'il existe des différences significatives.

Les chiffres en gras signifient que la performance de la technique correspondante est statistiquement moins bonne que celle de notre technique (*entropie*) avec un niveau de confiance de 95%. Les chiffres suivis par une étoile signifient la même chose mais avec un niveau de confiance de 90%. Avec le réseau bayésien naïf (NB), le test de Friedman ne rejette pas l'hypothèse nulle donc les tests post-hoc ne sont pas effectués.

Base de données	#attributs	#exemples	#classes	propriété
Car Evaluation	6	1728	4	symbolique
House Votes	16	435	2	symbolique
Tic Tac Toe	9	958	2	symbolique
Zoo	16	100[1]	7	symbolique
Promoter Gene Sequence	57	106	2	symbolique
Iris	4	150	3	numérique
Wine recognition	13	178	3	numérique
Ionosphere	32[2]	351	2	numérique
Liver-disorders	6	345	2	numérique
Pima Indians diabetes	8	768	2	numérique
Breast Cancer	9	683[3]	2	numérique
Glass identification	9	214	2	numérique
Yeast	8	1484	10	numérique

TAB. 4.5 – Description des bases de données utilisées

[1] Cette base contient en fait 101 exemples mais nous avons supprimé un doublon (frog).

[2] Cette base contient en fait 34 attributs mais nous en avons supprimé 2 selon la suggestion de [4].

[3] Cette base contient en fait 699 exemples mais nous avons supprimé 16 exemples qui contiennent des valeurs manquantes.

Nous constatons que notre méthode se comporte assez bien. Elle n'est jamais statistiquement inférieure à aucune technique et elle est supérieure à plusieurs techniques avec un niveau de confiance de 90% pour les classifieurs IB1 ou J48.

Pourtant elle est inférieure à *Cmode* sur tous les classifieurs, bien que la différence observée ne soit pas significative. Parmi les trois classifieurs, nous observons que notre méthode est plus performante si les méthodes J48 ou IB1 sont utilisées par la suite, alors qu'elle est plutôt mauvaise si le réseau bayésien naïf (NB) est utilisé. Avec J48, cela est peu surprenant. Cela peut être expliqué si on remarque que notre méthode et J48 optimisent, toutes les deux, le même critère durant leurs exécutions. Concernant les mauvais résultats avec le réseau bayésien naïf, il faut noter que toutes les méthodes sont jugées équivalentes par le test de Friedman. En outre, le meilleur rang moyen est relativement haut. Il est étonnant de constater que la méthode de substitution *aléatoire* conduit au meilleur rang moyen. Il serait intéressant d'étudier plus en détails ce phénomène. À partir de ces remarques, il semble clair que la qualité d'une méthode de substitution, dans notre contexte, dépend de la méthode

Substitution \ Classifieurs	J48	IB1	NB
entropie	2.94	2.84	4.3
Mode	**4.82***	3.4	4.52
CMode	2.26	2.36	3.64
classifieur J48	4.54*	**4.8***	3.7
classifieur IB1	4.36	4.34*	3.98
classifieur NB	4.28	**4.72***	4.56
Aléatoire	**4.8***	**5.54***	3.3

TAB. 4.6 – Données symboliques : les rangs moyens avec différentes techniques de substitution et techniques de classification

de classification qui suit. Tout cela est en accord avec notre hypothèse selon laquelle le classifieur utilisé après joue un rôle non négligeable dans le choix de technique de substitution dans un contexte de classification supervisée.

En regardant de manière plus détaillée, sans utilisation d'un test statistique, lors de l'analyse des résultats avec les différents taux de données manquantes, il s'avère que quand le taux de données manquantes augmente, le taux de bonnes classifications obtenu décroît. Il semble évident que le résultat dépend ainsi de la qualité des données d'entrée.

Avec un taux faible de données manquantes, les méthodes ne se différencient pas significativement. Par exemple, avec 3% de valeurs manquantes, les taux de bonnes classifications obtenus sur une même base de données sont très similaires. À 50% de données manquantes, il y a un écart significatif entre les résultats obtenus par les différentes méthodes.

4.3.4.2 Bases de données numériques

Comme nous l'avons mentionné, avec les bases de données numériques il faut effectuer une étape de discrétisation avant la substitution. Nous avons implémenté 3 techniques de discrétisation : EW, EF et discrétisation basée sur l'entropie (cf. section 2.4.2) que nous notons ici ID3. Avec la technique de substitution proposée, cela donne lieu à 3 combinaisons différentes : EW-*entropie*, EF-*entropie* et ID3-*entropie*.

Le tableau 4.7 décrit les rangs moyens avec différentes techniques de substitution et différentes techniques de classification sur les bases numériques. Les résultats sont moyennés sur les 8 bases de données numériques décrites dans le tableau 4.5. Les moyennes sont calculées de la même façon que dans le cas des valeurs symboliques. La présentation des résultats est identique à celle du tableau 4.6. Lorsque les tests post-hoc HC-SU sont effectués, nous comparons toutes les autres méthodes à $ID3-$

Substitution \ Classifieur	J48	IB1	NB
EW - entropie	5.8	5.625	**5.6375***
EF - entropie	4.6875	4.55	5.0875*
ID3 - entropie	4.5125	4.225	3.625
Moyenne	5.3875	4.75	**7.025***
CMoyenneS	5.25	3.025	3.725
ID3 - classifieur J48	5.6125	**6.7***	**5.5625***
5ppv	5.2375	6*	5.2875*
1ppvI	5.5375	**6.8125***	4.975*
LLSI	5.3	4.7125	**6.6875***
Aléatoire	**7.675***	**8.6***	**7.3875***

Tab. 4.7 – Données numériques : les rangs moyens avec différentes techniques de substitution et techniques de classification

entropie car elle semble la meilleure parmi les méthodes d'*entropie*. Beaucoup plus de techniques ont été examinées, mais seules les meilleures de chaque famille sont présentées. C'est pourquoi, par exemple nous présentons seulement le classifieur J48 avec la technique de discrétisation ID3.

Les résultats présentés dans le tableau 4.7 confirment les observations faites sur les données symboliques. Notre méthode n'est plus faible qu'aucune méthode pour les deux niveaux de confiance 90% et 95%. En outre, notre méthode est supérieure à toutes les autres méthodes au moins une fois avec le niveau de confiance de 90%, excepté *CmoyenneS*. La méthode *CmoyenneS* peut être considérée comme l'adaptation pour données numériques de la méthode *Cmode*. Comme pour le cas des données symboliques cette fois-ci, les deux techniques, la nôtre et *Cmode*, se trouvent en tête. Bien que de manière non significative, les rangs moyens de *ID3-entropie* sont légèrement meilleurs que ceux de *CmoyenneS* avec deux classifieurs (J48 et NB), alors que *Cmode* semble toujours meilleure sur des données symboliques.

Concernant la phase de discrétisation qui précède la substitution, il s'avère que la méthode ID3 a toujours un meilleur rang moyen que les deux autres (EW, EF). C'est probablement parce que son but est semblable à celui de notre technique : elle utilise également une mesure d'entropie et elle construit une partition en optimisant la capacité de discrimination d'un attribut. Cependant, nous ne pouvons confirmer aucune différence significative par un test statistique, excepté pour NB. Pourtant, la méthode de discrétisation *EF* est beaucoup moins complexe que la méthode de discrétisation ID3.

Parmi les méthodes de substitution, la méthode ID3-classifieur J48 semble mauvaise. Elle est clairement dominée par les deux techniques aux premiers rangs :

la nôtre et *CmoyenneS*. Cette remarque est également valable pour la méthode *aléatoire*. Nous l'avons utilisé seulement pour obtenir une référence qui doit être surpassée.

Parmi les deux versions de k-plus proches voisins, avec ou sans itération, on n'observe pas de différence significative. On peut se demander si le surcoût de la version itérative est très utile.

4.4 Conclusion

Dans ce chapitre, une nouvelle méthode de substitution des données manquantes dans le contexte de la classification supervisée a été proposée, issue de travaux menés en collaboration avec Thomas Delavallade [46, 55]. Son principe s'appuie sur l'hypothèse que la capacité de discrimination d'un attribut se dégrade en présence de données manquantes. La méthode proposée cherche donc à restaurer cette capacité pour un attribut donné en minimisant l'entropie de la base conditionnellement à celui-ci une fois complétée. L'objectif est d'optimiser la performance des classifieurs construits à partir des données complétées. Quelques propriétés de cette méthode ont été mathématiquement démontrées. Des expérimentations ont été menées pour valider cette méthode. Les résultats obtenus sont prometteurs.

Dans les sections précédentes, la mesure de discrimination utilisée est l'entropie de Shannon qui est la mesure la plus utilisée dans le domaine, elle sert à évaluer la capacité de discrimination de l'attribut. Cependant, théoriquement, ce choix n'est pas obligatoire, n'importe quelle mesure de discrimination présentée dans le chapitre 1 peut être utilisée. L'idée principale de la méthode est toujours la même : restaurer la capacité de discrimination d'un attribut ayant des valeurs manquantes.

Dans le cas où les attributs pour lesquels les valeurs manquantes sont floues, on peut garder toujours le même principe, le choix est donc parmi les modalités floues existantes. Les valeurs substituées sont celles qui rendent l'attribut le plus discriminant possible. Les mesures de discriminations floues peuvent servir à l'évaluation de cette capacité.

Nous pensons également à la construction d'un sous-ensemble flou qui remplace la valeur manquante en question. Le degré d'appartenance d'une valeur serait calculé en fonction de la mesure de discrimination obtenue quand la donnée manquante est substituée par cette valeur. Plus la mesure de discrimination correspondant à une valeur est grande, plus son degré d'appartenance est grand. L'étude plus formelle et les validations expérimentales restent à faire.

Troisième partie

APPLICATIONS

Chapitre 5

Applications

Dans ce chapitre, nous appliquons quelques-unes des techniques développées dans les chapitres précédents pour résoudre des problèmes réels. Deux problèmes ont été étudiés. Le premier est issu de travaux de thèse de Marc Damez [42] sur la modélisation d'utilisateurs par des traces d'interactions homme-machine. Le second consiste à la classification des messages électroniques envoyés à une société de service en informatique.

5.1 Classification de traces d'interactions homme-machine

Nous présentons ici un travail réalisé en collaboration avec Marc Damez. Ses recherches portent sur la modélisation d'utilisateurs afin de construire des systèmes d'éducation adaptatifs. L'un des buts de ces derniers est de caractériser les utilisateurs d'une interface. Une fois les caractéristiques de l'utilisateur identifiées, le système présente un comportement adéquat à l'utilisateur selon, entre autres, son niveau d'expertise ou son objectif d'usage du système. En particulier, le système est capable de donner des conseils et des suggestions aux utilisateurs. Nous proposons dans cette section d'intégrer des techniques d'arbres de décision dans un tel problème de modélisation d'utilisateurs.

Avec Marc Damez, nous avons développé le prototype d'un système adaptatif. Ce système est brièvement décrit dans les sections suivantes. La description détaillée du système, de l'expérimentation menée ainsi que de son résultat sortent du cadre de ce rapport de thèse. Ils sont décrits dans le rapport de thèse de Marc Damez [42] et dans [43].

5.1.1 Principe du système

5.1.1.1 Description générale

Lorsqu'un utilisateur utilise une interface afin d'atteindre ses objectifs, il doit effectuer plusieurs actions cognitives. Notre hypothèse est que les actions cognitives

impliquées dans la résolution d'une tâche reflètent le niveau d'expertise que l'utilisateur a de l'interface. Par exemple, dans la recherche d'un mot dans une page web, un utilisateur débutant parcourt la page en appuyant sur les touches flèches, tandis qu'une personne plus habituée peut faire l'appel à des fonctionnalités de recherche du navigateur. En fonction de ses habitudes, elle peut utiliser la souris pour accéder au menu ou utiliser une combinaison de touches (Ctrl + F).

En s'appuyant sur cette hypothèse, une application d'arbres de décision dans ce domaine a été développée [43]. Le système, intitulé TAFPA (Tree Analysis For Providing Advices), a été conçu pour récolter, exploiter des traces d'interaction homme-machine et fournir des aides à des utilisateurs débutants. Ce système fait partie des systèmes qui utilisent l'information acquise afin de concevoir des instructions (ou des conseils) aux utilisateurs. Ces conseils sont liés à la conception de l'environnement et sont fournis aux utilisateurs pour leur suggérer l'emploi de certaines fonctionnalités utiles de l'interface. Cela permet aux utilisateurs d'accélérer leur activité et facilite leur utilisation de l'interface.

TAFPA se compose de 2 phases : la phase d'apprentissage et la phase d'exploitation. La phase d'apprentissage est assurée essentiellement par un module d'apprentissage basé sur DTGen. À partir des fichiers de logs qui contiennent des descriptions des traces d'interaction homme-machine, elle construit des arbres de décision. Ces arbres caractérisent les différences entre des utilisateurs débutants et des utilisateurs expérimentés. Pendant la phase d'exploitation, TAFPA utilise les caractéristiques qui ont été précédemment apprises pendant le processus d'apprentissage pour reconnaître les débutants et leur donner des instructions. Cette conception de TAFPA est basée sur le principe proposé dans [71].

5.1.1.2 Phase d'apprentissage

Pendant la première phase, les traces sont d'abord récoltées par des techniques développées par Marc Damez. Elles sont formées des événements produits par l'interface graphique de l'environnement. Autant d'événements que possible devraient être collectés dans les fichiers de logs afin d'obtenir une meilleure représentation des interactions. Ces événements sont fortement reliés aux composants de l'interface. Ils reflètent également des tâches que les utilisateurs cherchent à résoudre. Pour chaque événement, un ensemble de descripteurs est conservé : sa cible (bouton, zone de saisie de texte, etc), son contexte et la date de son occurrence ainsi que par exemple l'étiquette du bouton, le texte de description. Ensuite les descripteurs extraits des fichiers de logs sont exploités par DTGen, qui est l'agent d'apprentissage de TAFPA pour caractériser les utilisateurs débutants et les utilisateurs expérimentés. Entre autres, la technique d'arbre de décision permet de faire ressortir un sous-ensemble d'attributs qui permet de discriminer les utilisateurs débutants des utilisateurs expérimentés. Les attributs qui ne dépendent que des caractéristiques des tâches à traiter ne sont ainsi que peu ou même pas considérés car ils sont communs pour les débutants et les expérimentés. Par exemple, la fréquence des passages entre les deux périphériques (le souris et le clavier) indiquera si l'utilisateur est à l'aise et/ou est ex-

périmenté avec l'interface. En s'appuyant sur les attributs discriminants retenus, on est capable de montrer sur quelles caractéristiques les débutants et les expérimentés sont différents et ainsi de produire des conseils aux débutants pour qu'ils puissent imiter le comportement des expérimentés. Cela renforce les raisons pour lesquelles l'arbre de décision est employé.

5.1.1.3 Phase d'exploitation

Pendant la deuxième phase, les traces de nouveaux utilisateurs sont classées par les arbres obtenus à l'étape précédente. Dès que les classifieurs obtiennent l'identification de la catégorie d'utilisateurs, TAFPA peut soit donner des conseils aux utilisateurs débutants, soit utiliser ces traces pour reconstruire des arbres afin d'exploiter encore les descripteurs qui différencient les débutants et les expérimentés. Les conseils donnés aux utilisateurs débutants sont dynamiquement conçus à partir des descripteurs qui caractérisent les utilisateurs expérimentés.

5.1.1.4 Construction de descripteurs

Pour reconnaître la catégorie d'utilisateurs en cours de manipulation de l'interface, la tâche globale doit être découpée en plusieurs étapes. Deux méthodes de décomposition sont considérées :

1. Manuelle : un expert doit regarder la tâche générale et trouver les différentes étapes que toutes les personnes doivent accomplir pour atteindre le but.
2. Automatique : cette méthode se compose de la recherche de la plus longue séquence commune (LCS pour *longest common sequence*) d'événements.

Pendant la phase d'exploitation de TAFPA, l'arbre appris doit fournir la catégorie de l'utilisateur (débutant ou expérimenté) à chaque étape. Pour cela, un arbre de décision doit être construit à la fin de chaque étape de la tâche globale à partir de tous les descripteurs des données rassemblées pendant les étapes précédentes. Dès que la réponse est jugée suffisamment précise, TAFPA prend une décision pour donner ou non un conseil à l'utilisateur. Par exemple, dans notre expérimentation, la tâche globale est de répondre à un questionnaire qui se compose de plusieurs questions. Chaque étape correspond donc à une question. Le système doit caractériser les utilisateurs après chaque question traitée afin de présenter un comportement adéquat pour la question suivante.

5.1.2 Expérimentations

L'expérimentation décrite ici a été conçue par Marc Damez ([42, 43]).

5.1.2.1 Description d'expérimentation

La navigation dans une page web est un processus constitué de plusieurs processus cognitifs. C'est pourquoi l'expérience réalisée a été basée sur la navigation

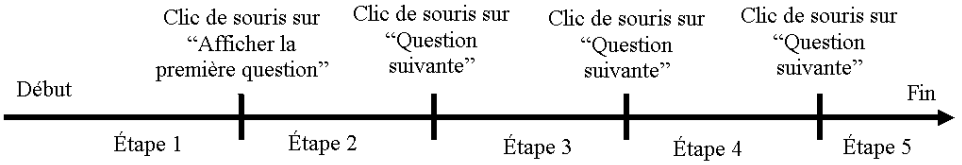

FIG. 5.1 – Décomposition manuelle : 5 étapes

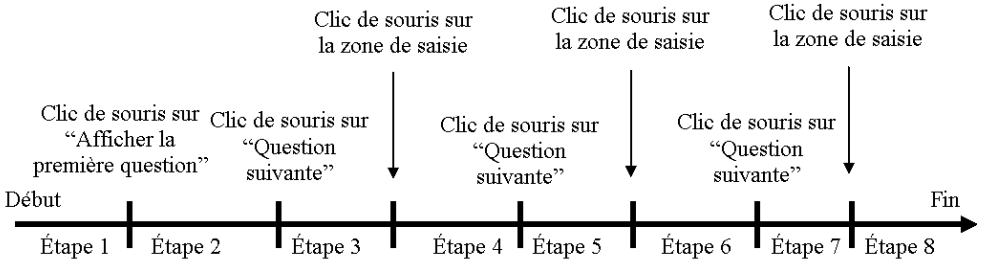

FIG. 5.2 – Décomposition automatique par l'algorithme basé sur LCS : 8 étapes

dans une page web pour accomplir quelques tâches en répondant à un questionnaire conçu par Marc Damez [43].

La première partie de la page du questionnaire présente la tâche à l'utilisateur. Elle lui demande de lire les textes donnés et de répondre à quelques questions. La tâche était, d'abord, de lire deux textes et de cliquer sur le bouton « Première Question » . Pour une question, une zone de saisie de texte pour la réponse et un bouton « Question suivante » apparaissent. Quatre questions sont ainsi posées. Cette expérience est menée sur deux ensembles d'utilisateurs : un ensemble d'utilisateurs expérimentés (familiarisés avec l'utilisation de navigateurs), et un ensemble d'utilisateurs débutants. Ces deux catégories d'utilisateurs sont considérées par l'agent d'apprentissage comme les classes à reconnaître.

Deux méthodes de décomposition de tâches décrites précédemment ont été étudiées. Les résultats sont présentés dans la partie suivante. Pour la décomposition par un expert, chaque étape est délimitée par des clics sur le bouton « Question suivante » . De cette façon, il a été établi 5 étapes, dont 4 étapes similaires (répondre à une question) qui peuvent être analysées avec les mêmes descripteurs et la première étape de lecture d'annonces (figure 5.1). L'utilisation de LCS sur l'ensemble des données expérimentées donne 8 étapes. Tous les délimiteurs par l'expert sont conservés et, de plus, 3 étapes qui correspondent aux clics sur les zones de saisie de texte où la réponse est dactylographiée y sont ajoutées (figure 5.2).

5.1.2.2 Résultats

L'expérience de Marc Damez a été menée avec la participation de 29 personnes dont 8 débutants et 21 expérimentés. Le système a collecté plus de 33000 événements

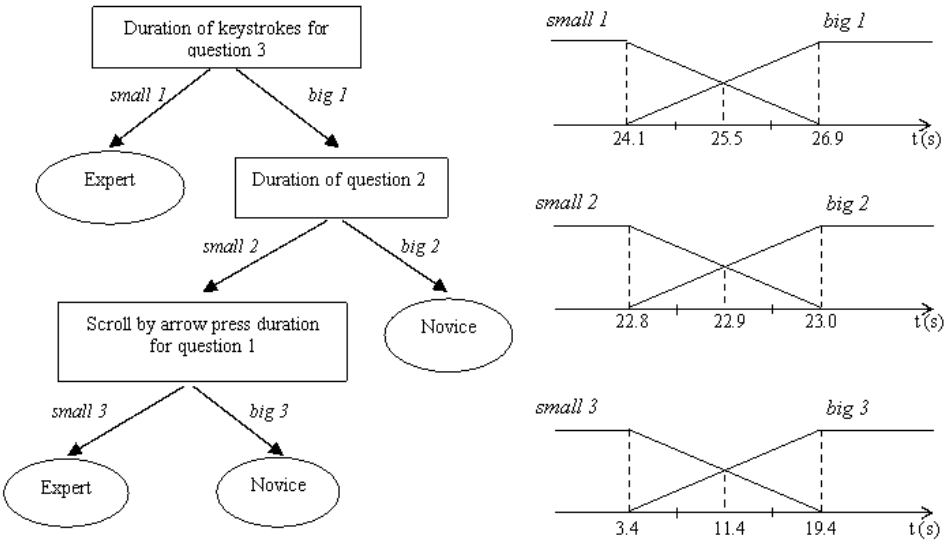

FIG. 5.3 – Un arbre construit par TAFPA

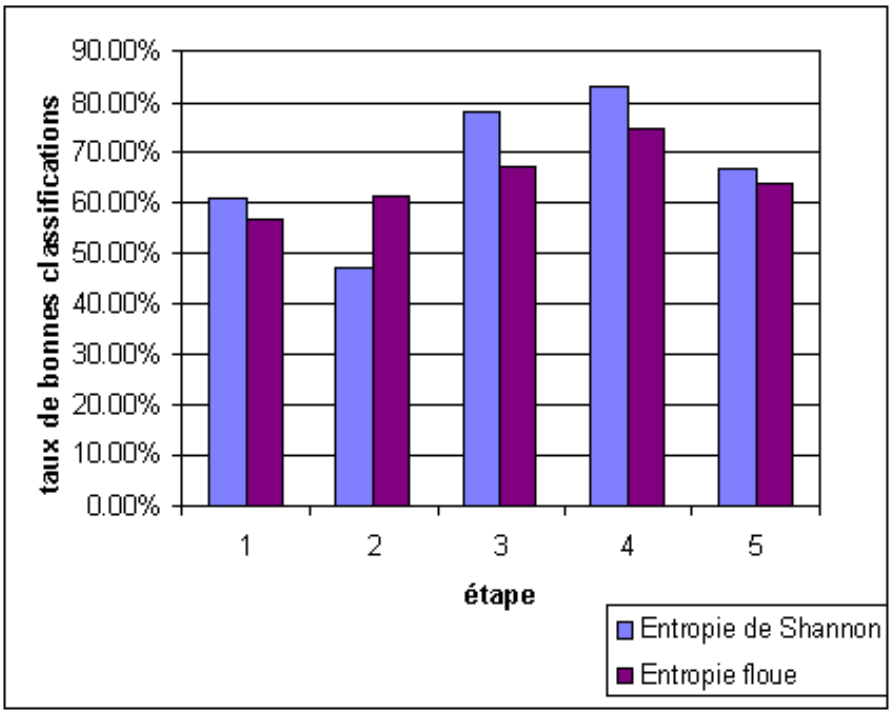

FIG. 5.4 – Taux de bonnes classifications avec la décomposition en 5 étapes

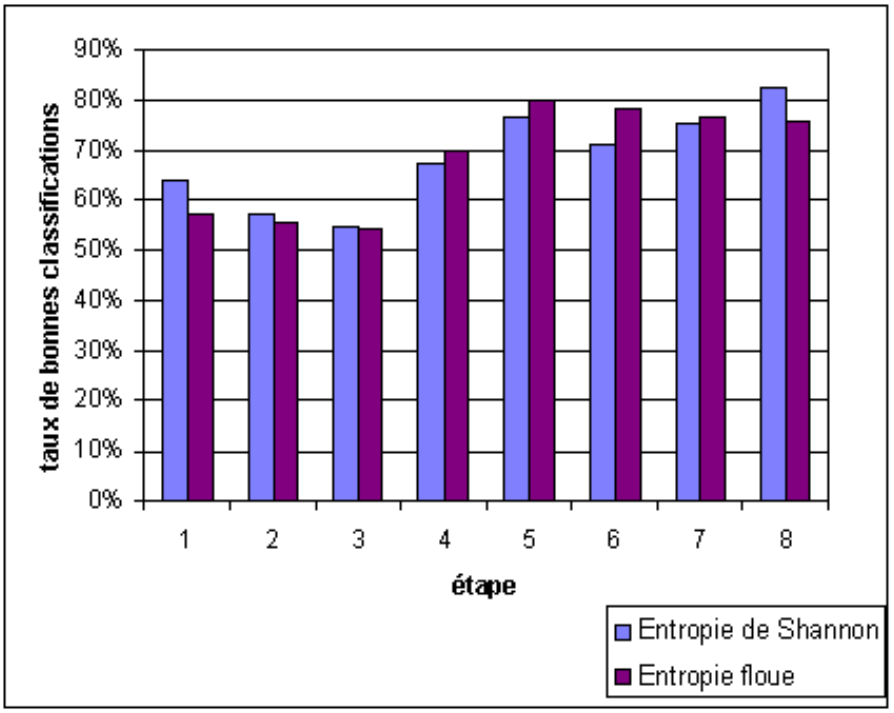

FIG. 5.5 – Taux de bonnes classifications avec la décomposition en 8 étapes

sur l'interface graphique. Le protocole *leave-one-out cross-validation* a été réalisé. À chaque fois, un exemple est retiré pour servir de base de test, le reste servant pour la base d'apprentissage. L'expérimentation est répétée 29 fois de telle sorte que chaque exemple est exactement retiré une fois.

Dans cette expérimentation, DTGen a été configuré avec pour but de générer des arbres de décision flous par l'entropie floue et des arbres de décision classiques par l'entropie de Shannon. Un exemple d'arbre de décision flou est présenté dans la figure 5.3. Les taux de bonnes classifications correspondant aux différentes méthodes de décomposition sont présentés dans les figures 5.4 et 5.5.

Dans le cas de 5 étapes, il s'avère que le système donne les meilleurs taux de bonnes classifications pour les étapes correspondant aux questions plus complexes (2 et 3). Dans le graphique de 8 étapes, les étapes sont délimités par des clics sur le bouton ou la zone de saisie de texte. Ici encore, nous pouvons observer de meilleures performances des arbres lorsque l'utilisateur est confronté à des questions complexes.

L'utilisation des arbres de décision flous n'apporte aucun avantage dans ces expérimentations. Probablement l'expérimentation a été menée sur un petit jeu de données dont l'espace d'exemples n'est pas assez dense, les exemples étant bien éloignés. Les classes sont ainsi bien séparées. Les expérimentés et les débutants utilisent vraiment différemment l'interface et les dispositifs, et leurs actions cognitives sont très différentes. Généralement, l'utilisation d'un arbre de décision flou est plutôt justifiée pour des données liées à des actions cognitives. Elle permet une classification

plus flexible et semble naturellement plus appropriée. Si plus de données sont incorporées pendant la phase d'apprentissage, on peut s'attendre à un meilleur résultat avec l'entropie floue.

5.1.3 Conclusion

Fournir une analyse des processus cognitifs est l'un des défis majeurs dans la modélisation d'utilisateurs et la conception adaptative d'hypermédias. Avec Marc Damez, nous avons développé un prototype de système d'analyse des traces d'interaction à l'aide de techniques d'arbres de décision. Il peut générer des conseils qui sont conçus à partir des comportements cognitifs des utilisateurs expérimentés. Cela permet de construire un système qui traite un problème applicatif. Il a été montré que l'arbre de décision possède des caractéristiques adaptées à ce genre de problèmes. Cette méthodologie peut donc être appliquée aux systèmes hypermédia d'éducation adaptatifs [42] et elle est développée par Marc Damez dans ses travaux de thèse.

5.2 Classification de courriels

5.2.1 Problématique

De nos jours, l'envoi de courriels est l'un des principaux modes de communication. Sa popularité peut être justifiée, entre autres, par sa rapidité, son faible coût et sa simplicité. Cependant, l'augmentation incessante du nombre de courriels échangés donne naissance à des défis considérables liés à leur traitement, en particulier, pour la classification des messages afin de filtrer les courriels non sollicités, ou pour acheminer les courriels vers le destinataire concerné. Ces deux problèmes sont connus dans le domaine de la classification de textes et font l'objet de nombreuses études [82, 89]. Cependant, si le filtrage de courriels non sollicités fait l'objet de plusieurs recherches et de nombreux développements, le deuxième problème est, lui, moins considéré.

Dans cette partie, nous nous intéressons plus particulièrement à ce problème qui apparaît maintenant dans de nombreuses applications. Ainsi, par exemple, on notera des applications pour les sociétés de service, mais aussi pour l'organisation d'une conférence scientifique, où les courriels sont généralement adressés à une adresse électronique unique et doivent, ensuite, être acheminés au service approprié qui devra les traiter.

Plusieurs méthodes existantes dans la litérature [153] permettent de traiter ce problème, mais aucune méthode n'utilise des arbres de décision flous ou, même des arbres de décision construits à l'aide d'autres mesures que l'entropie de Shannon. Ainsi l'objectif de nos travaux est double : d'une part, créer un outil générique, basé sur l'utilisation des arbres de décision, tant classiques que flous, pour la classification de courriels afin de les acheminer vers le destinataire concerné ; d'autre part, caractériser un certain nombre de techniques de construction d'arbres de décision afin de renforcer les conclusions du chapitre 2.

5.2.2 Solution proposée

La solution que nous proposons consiste à appliquer les techniques d'apprentissage par arbres de décision sur un corpus de courriels afin d'obtenir un ensemble de règles qui les caractérisent. Ces règles servent ensuite à classer les nouveaux courriels et les acheminer vers le destinataire approprié.

Tout d'abord, pour constituer la base d'apprentissage utilisable dans la construction d'arbres de décision, un ensemble d'attributs doit être choisi pour représenter les courriels. Ainsi, chaque courriel est décrit par un ensemble d'attributs, les valeurs qui leur sont associées et une classe. La classe à reconnaître est le thème du courriel qui sert à déterminer le destinataire. Ensuite, afin d'améliorer la performance du système, une présélection des attributs est effectuée pour réduire leur nombre. Enfin, l'algorithme de construction d'arbres de décision est appliqué sur la base d'apprentissage. L'arbre de décision ainsi obtenu sert ensuite à classer les nouveaux courriels. Le résultat de cette classification permet d'acheminer les courriels vers le destinataire approprié.

5.2.2.1 Extraction des attributs

La méthode que nous avons choisie pour représenter les courriels s'appuie sur le concept dit du « *sac de mots* ». Ainsi, un courriel est représenté par l'ensemble des mots qui le composent, sans prise en compte de leur ordonnancement.

De plus, dans notre approche, nous introduisons l'utilisation de n-grammes extraits du corps et du sujet des courriels [167, 112]. Il s'agit d'extraire, de chaque mot, toutes les sous-chaînes de n lettres consécutives qui le constituent. Un digramme est une suite de deux lettres consécutives, un trigramme est une suite de trois lettres dans un mot, etc. Dans le cas où on ne considère que des courriels composés en caractères latins (sans compter des lettres comme ç,à,é,è,ë,ù,û,û,...), on a 676 (26^2) digrammes et 17576 (26^3) trigrammes possibles. Certes certains digrammes ne figurent jamais dans la langue dans laquelle le courriel est écrit. Cependant il est possible que des fautes de frappe fassent apparaître certains d'entre eux. Certains figurent rarement et les autres apparaissent plus fréquemment. Pour information, une statistique effectuée sur la base de courriels montre qu'il y a un peu plus de 300 digrammes et 2000 trigrammes. Nous ne disposons pas de statistiques sur un échantillon plus grand.

De nombreux travaux ont montré l'intérêt de la représentation de textes par n-grammes. En particulier, Jalam et Chauchat [76] ont utilisé les n-grammes pour la recherche de mots-clefs pertinents. Ils ont énuméré les avantages suivants de codage en n-grammes :

1. capture automatique des racines de mots,
2. opération indépendante des langues,
3. tolérance aux fautes d'orthographes et aux déformations de texte causées par des raisons diverses,

4. non-nécessité d'éliminer les conjonctions, ou les articles, ni de procéder à la lemmatisation.

Dans la classification de courriels, nous avons rajouté des statistiques supplémentaires pour les caractériser : la longueur en nombre de mots du sujet, celle du corps, le degré de similarité du texte par rapport à des mots-clefs (déterminés automatiquement ou manuellement), l'apparition de dates, de chiffres, d'unités de monnaies, d'adresses, et de caractères spéciaux (par exemple la ponctuation).

Les mots clés sont déterminés selon le principe suivant : si un mot apparaît fréquemment (on doit fixer un seuil minimal pour le nombre d'occurrences) dans les courriels d'une classe et rarement (on fixe le taux maximal d'occurrences) dans les autres, alors il est considéré comme mot-clef de la classe. Dans les tests, les mots-clés sont identifiés sur la base d'apprentissage.

Pour renforcer la robustesse aux fautes de frappe, la similarité entre deux mots est prise en compte pour le calcul du degré de similarité entre textes. La similarité entre deux mots se mesure par le rapport entre le nombre de digrammes communs et le nombre total des digrammes des deux mots. Par exemple :

$$\mathrm{sim}(<logique>, <logiqe>) = 4/7$$

Pour un traitement plus fin, nous envisageons également de tenir compte des positions des digrammes dans un mot : la première position semble plus importante que les autres.

5.2.2.2 Sélection des attributs

Les attributs construits dans la phase précédente sont nombreux. Pour garantir la performance, en particulier pour que le temps de calcul et l'espace mémoire utilisé soient raisonnables, on ne peut pas les utiliser tous. Par la suite, seul un sous-ensemble d'attributs jugés les plus significatifs est sélectionné à l'aide d'heuristiques. Un élément (un mot, un digramme ou un trigramme) est considéré pertinent s'il apparaît dans un nombre minimal de courriels, ceci afin de réduire le biais introduit par les éléments trop rares provoquant une perte de généralisation. Entre autres, de tels éléments sont souvent engendrés par des fautes de frappe.

Les autres éléments sont ensuite filtrés selon leur capacité de discrimination dans la base d'apprentissage. Un élément t est choisi si sa présence dans un courriel entraîne significativement l'identification de la classe du courriel. Cela est mesuré en fixant un seuil maximal pour l'entropie de la base conditionnée par t : $I(\xi|t)$.

Pendant la phase d'extraction des attributs, pour les expérimentations, nous ne retenons que les mots apparaissant dans au moins 3 courriels et dont la présence entraîne une valeur d'entropie de Shannon inférieure à 0.6 pour les courriels où ils figurent. Les valeurs correspondantes pour les digrammes sont respectivement de 2 courriels et de 0.8 pour l'entropie, et pour les trigrammes de 2 courriels et de 0.6 pour l'entropie. Cela permet de retenir en moyenne environ une centaine de mots, une centaine de trigrammes et une vingtaine de digrammes.

5.2.3 Expérimentations

5.2.3.1 Description des données

L'expérimentation a été menée sur un corpus comportant 468 courriels, tous rédigés en français. Ce corpus a été fourni par une société de vente de jeux vidéo. Les courriels sont envoyés par les clients à une adresse électronique unique de la société. Chaque courriel est caractérisé par son sujet, son corps (le contenu) et la classe à laquelle il appartient. Les courriels étudiés sont courts et contiennent souvent des fautes d'orthographe. Les courriels se repartissent de manière assez équilibrée en 6 classes correspondant à des services différents de la société à qui sont destinés les courriels :

1. Classe 1 : Disponibilité de produits : 80 courriels
2. Classe 2 : Catalogue : 80 courriels
3. Classe 3 : Service après-vente (SAV) : 69 courriels
4. Classe 4 : Recherche de colis : 80 courriels
5. Classe 5 : Remboursement : 79 courriels
6. Classe 6 : Login/mot de passe : 80 courriels.

5.2.3.2 Protocole

Le protocole d'expérimentation décrit dans la section 2.3.3 est appliqué : à partir du corpus, 50% des courriels de chaque classe sont pris aléatoirement pour former une base d'apprentissage utilisée pour construire un arbre de décision. Tous les autres courriels constituent alors une base de test pour évaluer les arbres construits. Les courriels de cette base de test sont classés par l'arbre de décision appris sur la base d'apprentissage. Nous avons répété 8 fois cette expérimentation avec le corpus initial en faisant varier le choix aléatoire. Les résultats obtenus lors de chacune de ces expérimentations sont moyennés pour obtenir les résultats globaux.

Les entropies conditionnelles généralisées de Daróczy, celles de Rényi, l'entropie conditionnelle de Shannon et l'entropie floue sont intégrées dans la plateforme DTGen pour construire les arbres de décision et peuvent ainsi être comparées.

5.2.3.3 Résultats

La figure 5.6 montre les résultats de classification moyens, obtenus en faisant varier de 0.1 à 10 l'ordre β des entropies de Daróczy et de Rényi. Le taux de bonnes classifications obtenu en utilisant l'entropie floue est de 84.04%±2.31%. Il est légèrement meilleur que celui obtenu par l'entropie de Shannon qui est de 83.24% ±2.33%. Ces deux taux ne dépendent pas de β. Sur la figure 5.6, ils sont représentés par des lignes horizontales. Le meilleur taux de bonnes classifications, de 85.57%±3.88%, est obtenu avec l'entropie conditionnelle de Rényi de Type 1 pour $\beta = 0.5$. Le meilleur taux de bonnes classifications lorsque $\beta > 1$ est de 85.24%±3.58% et est obtenu avec l'entropie conditionnelle de Daróczy de Type 1 avec $\beta = 3.9$. Le taux moyen

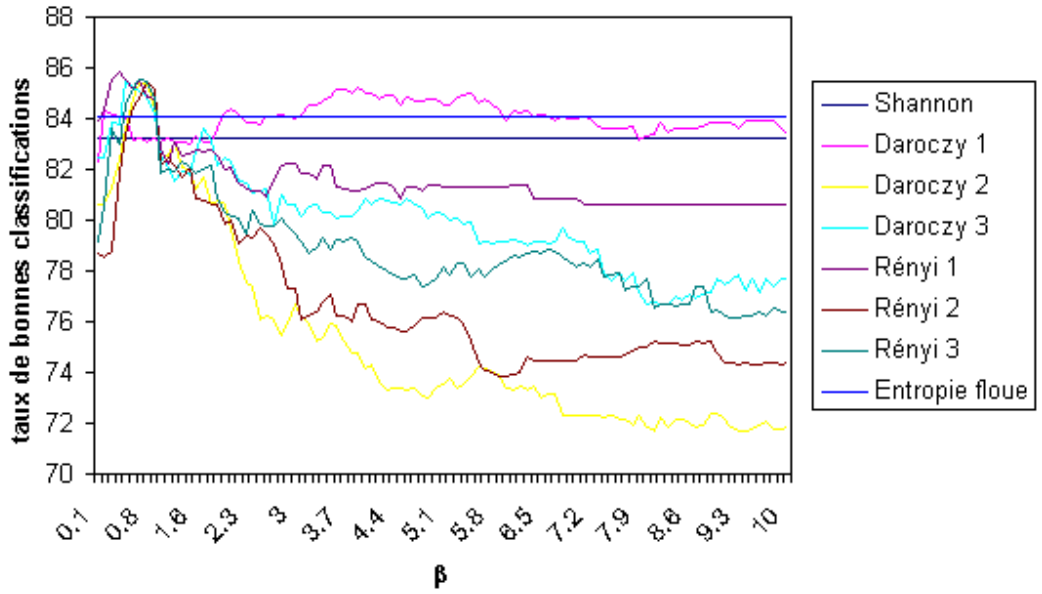

FIG. 5.6 – Taux moyen de bonnes classifications sur la base des courriels avec différentes formules conditionnelles, $\beta \in (0, 10]$

de gain d'information obtenu en utilisant l'entropie floue et l'entropie de Shannon sont respectivement de 64.93% et de 64.51%. Le résultat sur la base de courriels est relativement en accord avec les résultats présentés dans la section 2.3.3 du chapitre 2 (cf. figure 2.2 page 62) : les tendances de variation des performances sont similaires.

D'autres expérimentations ont été conduites. Nous avons obtenu une meilleure moyenne des taux de bonnes classifications (89.5%) en utilisant l'entropie conditionnelle de Shannon et en appliquant une technique d'élagage lors de la construction de l'arbre de décision : le processus de caractérisation sur une branche de l'arbre s'arrête lorsque l'entropie devient inférieure à 0.6. Les techniques d'élagage avec d'autres formules d'entropies conditionnelles n'ont pas encore été étudiées.

5.2.4 Discussion

À travers cette étude, il s'avère que le codage influence beaucoup les résultats obtenus. Les trigrammes semblent plus utiles que les digrammes et les mots. Lors de l'utilisation exclusive de mots, ou de digrammes, ou de trigrammes, les meilleurs résultats sont fournis par les trigrammes. Il est probable que, pour ce corpus, le vocabulaire est trop étendu, et qu'il existe énormément de fautes d'orthographe, ce qui les rend difficile à les exploiter. Au contraire, les digrammes semblent trop élémentaires car ils n'apportent que peu d'information sur la sémantique des courriels.

Dans notre système, la sélection d'attributs pour réduire le nombre d'attributs s'appuie sur une mesure d'entropie. Le principe de sélection est ainsi proche de celui utilisé dans la construction d'arbres de décision. D'autres techniques de sélection

d'attributs sont également envisageables et devraient être examinées.

L'utilisation de caractéristiques supplémentaires devrait aider à améliorer le résultat. Dans une application réelle, d'autres éléments devront être pris en compte, en particulier l'interaction avec l'utilisateur et l'évolution des courriels au cours du temps. Cependant, l'identification des caractéristiques supplémentaires dans notre approche nécessite l'intervention humaine, notamment dans l'identification des mots clés. Afin de renforcer l'applicabilité de notre système, il faut automatiser cette tâche.

Sur les techniques d'apprentissage, l'utilisation d'une technique d'élagage par critère d'arrêt donne des effets très positifs sur le taux de bonnes classifications et sur la taille de l'arbre obtenu (qui est réduite). Nous nous proposons d'approfondir l'étude sur cet effet d'élagage dans des travaux ultérieurs. Enfin, afin de valider notre étude, il reste à comparer notre méthode aux méthodes d'apprentissage existantes, en particulier, avec des classifieurs généralistes performants tels que les SVM ou Adaboost.

CONCLUSION ET PERSPECTIVES

Conclusion et perspectives

Nos travaux concernent l'étude des mesures de discrimination, tant classiques que floues, et leurs applications en apprentissage inductif. La technique d'apprentissage que nous avons considérée est l'apprentissage par arbre de décision. Les mesures ont été tout d'abord examinées dans le cadre d'une approche axiomatique. Cette étude a conduit à la proposition d'un modèle hiérarchique pour les mesures de discrimination floues. Ces mesures ont ensuite été appliquées aux différentes étapes de la construction des arbres de décision, en particulier la sélection du meilleur attribut et la discrétisation des attributs numériques. Dans chacune de ces étapes, les caractéristiques des mesures ont été considérées. L'étude de ces mesures a donné lieu à un choix plus large de mesures dans la construction d'arbres de décision dans le but d'obtenir des arbres possédant des propriétés spécifiques. Un système de construction d'arbres de décision a été réalisé et testé sur différentes bases de données, y compris des données issues d'applications réelles.

Nous avons ensuite étudié l'utilisation des mesures de discrimination pour d'autres problématiques existant dans le cadre du processus d'apprentissage inductif. Le traitement des données manquantes a d'abord été considéré. Nous avons proposé dans cette étude une technique de substitution des valeurs manquantes basée sur la capacité de discrimination. Nous avons considéré ensuite la capacité de discrimination des classifieurs et proposé d'utiliser cette capacité comme critère d'évaluation.

Le point commun entre les trois tâches que nous avons étudiées est le suivant : la sélection du meilleur attribut, la discrétisation des attributs numériques et l'évaluation des classifieurs concernent toutes les trois la capacité de discrimination que ce soit d'un attribut, d'un point de coupure ou d'un classifieur. Dans chacune de ces étapes, une partition de la base d'exemples est engendrée :

1. Partition selon les valeurs d'un attribut : les exemples possédant une même valeur pour un attribut sont regroupés dans un sous-ensemble.
2. Partition selon les coupures du domaine d'un attribut numérique : les exemples dont la valeur pour un attribut se trouve dans un même intervalle engendré par les points de coupure sont regroupés dans un sous-ensemble.
3. Partition selon les classes issues d'un classifieur : les exemples classés dans une même classe sont regroupés dans un sous-ensemble.

En apprentissage inductif, il est préférable que chacune de ces partitions soit la plus adéquate possible avec la partition selon les classes réelles des exemples. Le modèle

hiérarchique que nous avons présenté dans le chapitre 1 constitue un outil d'étude de cette adéquation dans le contexte de l'apprentissage inductif aussi bien classique que flou. La capacité de discrimination est donc considérée comme un critère pour évaluer un attribut, une discrétisation ou un classifieur. De plus, notre méthode de traitement de données manquantes se fonde sur l'hypothèse que la donnée manquante détériore la capacité de discrimination d'un attribut. Elle cherche alors à restituer ce type de capacité.

Principaux résultats obtenus

Caractérisation des mesures de discrimination

Nous avons mené une étude axiomatique sur les mesures de discrimination classiques et les mesures de discrimination floues. En particulier, nous avons caractérisé les mesures d'entropie et les formes conditionnelles qui leur sont associées. Nous avons montré que les mesures classiques satisfont les propriétés imposées par le modèle hiérarchique introduit par Marsala en 1998 [102] pour des mesures de discrimination.

Nous avons construit un nouveau modèle, appelé modèle $\mathcal{FGH}^*$, qui généralise le modèle précédent et qui intègre des concepts provenant de la théorie des sous-ensembles flous afin de prendre en compte les mesures de discrimination floues. Ce modèle impose un certain nombre de caractéristiques aux mesures de discriminations floues. Dans le cadre de ce modèle, les mesures de discrimination floues existantes et les mesures généralisées à partir des mesures classiques sont caractérisées. Les résultats obtenus ont donné lieu à plusieurs publications [45, 47, 44].

Utilisation des mesures de discrimination dans la construction des arbres de décision

Nous avons introduit des mesures de discrimination, classiques et floues, utilisées dans les différentes étapes de la construction des arbres de décision. En particulier, nous avons caractérisé ces mesures dans la sélection du meilleur attribut et la discrétisation des attributs numériques. Cela nous a conduit à proposer une taxonomie des méthodes de construction d'arbres de décision flous qui s'appuie sur la méthode de sélection du meilleur attribut et la stratégie d'identification des fonctions d'appartenance. Les résultats obtenus ont été publiés dans [45, 23].

Les mesures de discrimination proposées ont été intégrées dans le système DT-Gen. Ce dernier nous a permis d'étudier le comportement des mesures de discrimination. Des expérimentations sur des bases de données conventionnelles ont été menées avec cet outil. En particulier, en collaboration avec Marc Damez, nous avons développé un système d'analyse de traces d'interactions homme-machine. Cette application a été publiée dans [43].

Évaluation de la capacité de discrimination de classifieurs

Enfin, nous avons examiné la capacité de discrimination des modèles de classification et proposé de mesurer la qualité d'un classifieur, en particulier, sur sa capacité à discriminer des exemples dans les différentes classes. Ce critère prend en compte l'adéquation entre la partition de l'ensemble des exemples selon leurs vraies classes et selon celles obtenues par le classifieur. Il donne une estimation de l'apport effectif du classifieur sur l'identification des classes des exemples en tenant compte de la complexité du problème. Cette proposition a fait l'objet d'une publication [48].

Cette idée a également été développée dans le domaine de la recherche d'images : une technique de caractérisation des bases d'images et d'évaluation des résultats de classification basée sur une mesure d'information a été proposée [18].

Substitution de données manquantes par une méthode basée sur l'entropie

Nous avons proposé, en collaboration avec Thomas Delavallade, une nouvelle méthode de substitution des données manquantes dans le contexte de la classification supervisée. Cette nouvelle méthode s'appuie sur l'hypothèse que la capacité de discrimination d'un attribut se dégrade à cause des données manquantes. Notre méthode substitue donc des valeurs manquantes afin de restituer cette capacité pour un attribut donné. L'objectif est d'optimiser la performance des classifieurs construits à partir des données complétées et non de trouver des valeurs aussi proches que possible des valeurs réelles qui, dans la pratique, ne sont pas connues.

Nos travaux sur les données manquantes ont donné lieu à deux publications [46, 55].

Perspectives

Nos travaux ouvrent de nombreuses perspectives pour chacun des axes que nous avons développés. Nous les détaillons dans ce qui suit.

Sur les mesures de discrimination et la construction d'arbres de décision

En ce qui concerne les mesures de discrimination, nous proposons d'approfondir l'étude sur le modèle hiérarchique flou afin de mieux caractériser les mesures de discrimination floues et ainsi caractériser les arbres obtenus avec chacune des mesures. L'étude des mesures de discrimination doit être élargie notamment aux mesures récemment proposées dans l'état de l'art, en particulier dans [154]. Le rapport entre ces mesures et les propriétés imposées par les modèles $\mathcal{FGH}$ et $\mathcal{FGH}^*$ doit être mis en évidence. Comme le modèle hiérarchique flou est un modèle constructif, des nouvelles mesures de discrimination peuvent être introduites. Il est possible que des

variantes des mesures de discrimination soient utilisées pour prendre en compte certaines propriétés spécifiques du problème considéré, en particulier le problème de classification avec la prise en compte de coûts d'erreur non uniformes, ou la prise en compte du coût d'obtention des données. Dans les modèles hiérarchiques, peu de propriétés sont imposées aux niveaux $\mathcal{H}$ et $\mathcal{H}^*$ et il sera intéressant de considérer la possibilité d'y ajouter des propriétés particulières pour s'adapter à des problèmes particuliers.

Cette étude sera, dans un premier temps, utile pour aider les utilisateurs dans leur choix d'une mesure de discrimination appropriée en fonction des propriétés de l'arbre qu'ils souhaitent obtenir. Elle devra ultérieurement doter le système de la capacité de choisir lui-même la mesure de discrimination la plus adaptée au problème qu'il doit traiter.

Il serait également intéressant de mettre en évidence les avantages (entre autres la robustesse, l'interprétabilité, la performance) ainsi que les faiblesses des arbres de décision flous (temps de calcul par exemple) construits par différentes mesures de discrimination floues. Cette étude doit être menée aussi bien sur le plan théorique que sur le plan expérimental. Quelques expériences ont été menées dans le chapitre 2 mais elles sont à compléter.

L'utilisation des mesures de discrimination dans la sélection d'attributs et dans la discrétisation des attributs numériques doit être mieux caractérisée. L'une des questions que nous souhaitons approfondir vise à caractériser les partitions obtenues par chaque mesure ou chaque famille de mesures. L'intervention de mesures de discrimination dans d'autres étapes de la construction des arbres de décision, en particulier l'élagage, doit être examinée plus profondément. Comme nous l'avons noté dans l'application concernant les courriels, l'élagage permet de renforcer la performance du système, en particulier la taille du modèle, sa précision ainsi que son interprétabilité.

Le modèle hiérarchique flou permet actuellement d'étudier l'adéquation entre des sous-ensembles flous et des partitions floues. Pourtant, cette étude doit être poursuivie afin d'adapter ces modèles à des entropies d'événements flous.

Nous envisageons également de doter DTGen de certaines fonctionnalités plus avancées, en particulier des techniques d'élagage, de discrétisation et de combinaison de plusieurs arbres différents pour résoudre une tâche. Ces techniques existent dans la version actuelle mais elles sont encore simples. Nous préférons également développer des techniques de traitement de bases ayant des attributs flous et de bases avec des classes floues. La mise en pratique de ce système doit être étendue.

Sur l'évaluation de modèle de classification

Des perspectives de recherche apparaissent également dans le cadre du processus d'évaluation de la performance des classifieurs. L'étude menée dans cette thèse doit être approfondie pour mieux caractériser la capacité de discrimination des classifieurs. La relation entre cette mesure et les autres est à examiner.

Notre critère se fonde sur l'étude de l'adéquation entre la partition par de vraies

classes et la partition issue du classifieur. La mesure de discrimination est une manière d'évaluer l'adéquation. Cependant, une mesure de similarité entre deux partitions peut servir dans ce but. Cette idée est également à étudier.

En particulier, la qualité d'un arbre de décision est un concept difficile à définir. En rejoignant des perspectives sur les méthodes de construction d'arbres de décision, il sera intéressant de considérer les différents critères de performance d'arbre de décision, en particulier d'arbres de décision flous. Il s'avère qu'aucun critère n'est suffisant pour déterminer si un arbre est meilleur qu'un autre ou, plus généralement, si une méthode de construction d'arbres est meilleure qu'une autre. Une combinaison des critères doit être étudiée. La manière de combiner doit prendre en compte les préférences des utilisateurs sur une application spécifique et mettre l'accent sur les critères de qualité importants à l'application.

Sur le traitement des valeurs manquantes

En ce qui concerne notre méthode de substitution des données manquantes, nous nous proposons de caractériser plus finement la méthode. Cela permettra éventuellement une implémentation plus efficace, en limitant l'espace de recherche par exemple ou en conduisant à une initialisation de l'algorithme par une substitution plus proche de la solution optimale.

Nous voulons approfondir l'étude concernant l'impact de la phase de discrétisation sur la performance de notre méthode de substitution lorsqu'elle est appliquée sur des données numériques.

Les résultats expérimentaux ont montré qu'aucune technique n'est dominante. Il faut donc identifier les problèmes pour lesquels la méthode proposée est la plus adaptée. Ces problèmes peuvent être caractérisés par la quantité de données manquantes, le mécanisme de génération des données manquantes, etc. En particulier, nous aimerions étudier comment fonctionne notre méthode avec différents mécanismes de génération des données manquantes.

Par ailleurs, notre méthode a été conçue dans une optique que l'on peut qualifier d'optimiste, en supposant que les attributs sont discriminants par défaut. Il sera intéressant d'analyser le comportement de cette méthode lorsqu'elle est appliquée sur un attribut ayant une faible capacité de discrimination.

La relation entre la méthode de traitement des valeurs manquantes et la méthode d'apprentissage qui l'utilise ensuite doit être étudiée, en particulier pour les techniques d'arbres de décision car les principes sont relativement liés. Nous préférerions savoir si notre méthode de substitution favorise une méthode d'apprentissage spécifique.

ANNEXES

DTGen

DTGen (*Decision Tree Generation*) est une plateforme d'expérimentation que nous avons enrichie et utilisée dans le cadre de notre étude. Elle fournit un outil pour construire des arbres de décision à partir d'une base de données numériques et/ou symboliques selon la stratégie TDIDT et pour les utiliser pour classer de nouveaux exemples. Le système DTGen initial a été conçu et développé par Longuet et Stermann [145] en Java. La version initiale du système permettait de construire un arbre de décision à partir d'une base d'apprentissage ayant des attributs symboliques ou numériques par ID3, la seule mesure de discrimination utilisée était l'entropie de Shannon. Ensuite, plusieurs techniques ont été ajoutées au cours de cette thèse. Différentes mesures de discrimination ont été implémentées : l'entropie de Rényi et de Daróczy et leurs différentes formes conditionnelles, des entropies floues. Certaines techniques d'élagage et de discrétisation ont été intégrées. Le système est actuellement doté de la capacité de construire et d'utiliser plusieurs arbres. Il est capable de générer et d'utiliser des arbres flous. Cependant, ce système ne permet pas encore de traiter des bases ayant des attributs flous ou des bases avec des classes floues. Les autres mesures de discrimination seront implémentées par la suite.

Le processus de construction et d'utilisation d'arbres de décision dans DTGen se compose des étapes suivantes :

Discrétisation d'attributs numériques

À chaque itération, avant la sélection du meilleur attribut, les attributs numériques sont d'abord dynamiquement discrétisés en deux intervalles par une mesure d'entropie (classique ou floue) au choix des utilisateurs. Plusieurs mesures sont proposées : l'entropie conditionnelle de Shannon, les entropies conditionnelles de Rényi, celles de Daróczy et les versions floues de ces entropies. Le système est capable de donner une coupure floue ou une coupure précise. La coupure choisie est centrée au milieu de deux valeurs voisines d'un attribut, et rend les plus homogènes possible les parties issues de la discrétisation. Avec les mesures classiques (non-floues), deux types de coupure sont disponibles : coupure précise et coupure floue. Une coupure précise est une valeur réelle, déterminée par l'algorithme basé sur une mesure d'entropie présentée dans le chapitre 1. Une coupure floue est une extension d'une coupure précise des deux côtés par étalement. L'étalement est un paramètre de l'algorithme et il est choisi comme une fonction de la différence entre deux valeurs les plus proches (à gauche et à droite) de la coupure précise. Avec les mesures floues,

seule une coupure floue est retournée. Pour mettre en évidence l'apport de la mesure de discrimination dans le processus de discrétisation, une détermination aléatoire de la coupure est également implémentée.

Sélection du meilleur attribut

Les attributs numériques discrétisés sont considérés ensuite comme des attributs symboliques. Le sélection du meilleur attribut est réalisée à l'aide de l'une des mesures d'entropie (classiques ou floues) présentées dans le chapitre 1 : l'entropie conditionnelle de Shannon, les entropies conditionnelles de Rényi, celles de Daróczy et les versions floues correspondantes. La discrétisation et le sélection du meilleur attribut peuvent être effectuées par une même mesure de discrimination ou par des mesures différentes.

Partitionnement en sous-bases

La base associée au nœud est partitionnée selon les valeurs d'attribut. Chaque valeur de l'attribut engendre un nœud fils du nœud considéré. Dans la construction des arbres de décision flous, l'identification des fonctions d'appartenance se fait selon la procédure décrite par la figure 2.13 (page 84). Si un exemple appartient avec un degré trop faible à une sous-base, il est éliminé. Le seuil d'élimination est fixé à 0.05.

Condition d'arrêt

La construction d'un arbre de décision s'arrête lorsque l'un des critères d'arrêt est vérifié. Les critères d'arrêt sont déterminés comme paramètre de l'algorithme. Par défaut, l'algorithme s'arrête quand tous les exemples associés au nœud appartiennent à une seule classe. Le paramétrage système permet d'arrêter le partitionnement quand les ensembles des exemples sont suffisamment homogènes ou quand leur effectif est petit. Pour les arbres de décision flous, ces conditions d'arrêt sont adaptées. On utilise l'entropie floue au lieu des entropies classiques et la cardinalité floue au lieu du nombre d'exemples. À chaque étape, les exemples appartenant à la sous-base avec un faible degré d'appartenance sont éliminés.

Utilisation d'arbres de décision

DTGen dispose de différentes stratégies de classification qui peuvent être choisies. L'utilisateur peut choisir de construire un ou plusieurs arbres selon différentes mesures. Dans le cas d'utilisation de plusieurs arbres, les résultats donnés par chacun sont agrégés. Le choix entre arbre flou et arbre classique doit être spécifié.

DTGen est l'outil de construction d'arbres de décision utilisé dans les expériences menées dans cette thèse. En particulier, l'application de ce système à quelques problèmes réels tels que la modélisation des utilisateurs par les traces d'interaction homme-machine et la classification des courriels est présentée dans le chapitre 5.

Bibliographie

[1] S. Abe. Axioms and uniqueness theorem for Tsallis entropy. *Physics Letters A*, 271A :74–79, 2000.

[2] S. Abe. Tsallis entropy : how unique ? *Continuum Mechanics and Thermodynamics*, 16 :237–244, 2004.

[3] S. Abe and A. Rajagopal. Nonadditive conditional entropy and its significance for local realism. *Physica A*, 289 :157–164, 2001.

[4] E. Acuna and C. Rodriguez. The treatment of missing values and its effect in the classifier accuracy. In *Classification, Clustering and Data Mining Applications*, pages 639–648. Springer-Verlag, 2004.

[5] J. Aczél. On different characterizations of entropies. *Probability and Information Theory : Lecture Notes in Mathematics*, pages 1–11, 1971.

[6] J. Aczél and Z. Daróczy. On measures of information and their characterizations. *Mathematics in Science and Engineering*, 115 :577–579, 1975.

[7] J. Aguilar-Ruiz, J. Cacardit, and F. Divina. Experimental evaluation of discretization schemes for rule induction. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'04*, volume 3102 of *LNAI*, pages 828–839, Seattle, WA, USA, 2004. Springer-Verlag.

[8] I. Alvarez. Explaining the result of a decision tree to the end-user. In *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'04*, pages 411–415, Valencia, Spain, 2004.

[9] N. B. Amor, S. Benferhat, and Z. Elouedi. Towards a definition of evaluation criteria for probabilistic classifiers. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainly, ECSQARU'05*, volume 3571 of *LNCS*, pages 921–931, 2005.

[10] N. B. Amor, S. Benferhat, and Z. Elouedi. Information-based evaluation functions for probabilistic classifiers. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'06*, pages 395–400, Paris, France, 2006.

[11] A. An and N. Cercone. Discretization of continuous attributes for learning classification rules. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD'99*, volume 1574 of *LNAI*, pages 509–514. Springer-Verlag, 1999.

[12] L. Bartczuk and D. Rutkowska. A new version of the Fuzzy-ID3 algorithm. In *Proceedings of the International Conference on Artificial Intelligence and Soft Computing, ICAISC'06*, pages 1060–1070, 2006.

[13] M. Basu and T. K. Ho. *Data Complexity in Pattern Recognition (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[14] G. Batista and M. C. Monard. An analysis of four missing data treatment methods fo supervised learning. *Applied Artificial Intelligence*, 6(3) :309–327, 2003.

[15] P. Benvenuti. Sur l'indépendance dans l'information. In *Colloques Internationaux du CNRS, No 276, Théorie de l'information*, pages 49–55, 1975.

[16] J. Bezdek and J. Harris. Fuzzy partitions and relations : An axiomatic basis for clustering. *Fuzzy Sets and Systems*, 2(1) :111–127, 1978.

[17] S. Bothorel. *Analyse d'image par arbre de décision flou, Application à la classification sémiologique des amas de microcalcifications*. PhD thesis, Université Paris VI, Paris, France, 1996.

[18] A. Boucher, T. H. Dang, and T. L. Le. Classification vs recherche d'information : vers une caractérisation des bases d'images. In *Actes de la 12ème Rencontre de la Société Francophone de Classification, SFC'05*, pages 75–78, Montréal, Canada, 2005.

[19] B. Bouchon. Fuzzy partitions. In M. G. Singh, editor, *Systems and Control Encyclopedia*, pages 1835–1838. Pergamon Press, 1988.

[20] B. Bouchon-Meunier. *La logique floue*. No 2702 in Collection : Que sais-je ? Presses Universitaires de France, 3ème édition, France, 1994.

[21] B. Bouchon-Meunier. *La logique floue et ses applications*. Addison Wesley, France, 1995.

[22] B. Bouchon-Meunier, G. Coletti, and C. Marsala. Independence and possibilistic conditioning. *Annals of Mathematics and Artificial Intelligence*, 35 :107–124, 2002.

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

[24] X. Boyen and L. Wehenkel. Automatic induction of fuzzy decision trees and its application to power system security assessment. *Fuzzy Sets and Systems*, 102 :3–19, 1999.

[25] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7) :1145–1159, 1997.

[26] L. Breiman. Technical note : Some properties of splitting criteria. *Machine Learning*, 24(1) :41–47, 1996.

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

[28] L. A. Breslow and D. W. Aha. Simplifying decision trees : a survey. *Knowledge Engineering Review*, 12(1) :1–40, 1997.

[29] C. E. Brodley and P. E. Utgoff. Multivariate decision trees. *Machine Learning*, 19(1) :45–77, 1995.

[30] W. Buntine and T. Niblett. A further comparison of splitting rules for decision tree induction. *Machine Learning*, 8 :75–85, 1992.

[31] R. Caruana, T. Joachims, and L. Backstrom. KDD-Cup 2004 : results and analysis. *SIGKDD Explorations*, 6(2) :95–108, 2004.

[32] R. Caruana and A. Niculescu-Mizil. Data mining in metric space : An empirical analysis of supervised learning performance criteria. In *Proceedings of the 1st Workshop on ROC Analysis in AI, ROCAI'04*, pages 9–18, 2004.

[33] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23th International Conference on Machine Learning, ICML'06*, pages 161–168, 2006.

[34] J. Catlett. On changing continuous attributes into ordered discrete attributes. In Y. Kodratoff, editor, *Proceedings of the European Working Session on Learning*, volume 482 of *LNCS*, pages 164–178. Springer-Verlag, 1991.

[35] J.-Y. Chang, C.-W. Cho, S.-H. Hsieh, and S.-T. Chen. Genetic algorithm based Fuzzy ID3 algorithm. In *Proceedings of the International Conference on Neural Information Processing, ICONIP'04*, pages 989–995, 2004.

[36] I.-J. Chiang and J. Y.-J. Hsu. Fuzzy classification trees for data analysis. *Fuzzy Sets and Systems*, 130(1) :87–99, 2002.

[37] K. Cios and L. Sztandera. Continuous ID3 algorithm with fuzzy entropy measures. In *Proceedings of the 1st IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'92*, pages 469–476, San Diego, USA, 1992.

[38] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20 :37–46, 1960.

[39] M. Contat, V. Nimier, and R. Reynaud. Request management using contextual information for classification. In *Proceedings of the The 5th International Conference on Information Fusion, FUSION'02*, volume 2, pages 1147– 1153, Annapolis, Maryland, 2002.

[40] A. Cornuéjols, L. Miclet, and Y. Kodratoff. *Apprentissage artificiel : Concepts et algorithmes*. Eyrolles, France, 2002.

[41] K. Crockett, Z. Bandar, D. Mclean, and J. O'Shea. On constructing a fuzzy inference framework using crisp decision trees. *Fuzzy Sets and Systems*, 157 :2809–2832, 2006.

[42] M. Damez. Modélisation à partir de traces d'utilisation : application à la modélisation cognitive de l'utilisateur et personnalisation d'interface adaptative. Manuscrit de thèse, Université Paris VI, 2007.

[43] M. Damez, T. H. Dang, C. Marsala, and B. Bouchon-Meunier. Fuzzy decision tree for user modeling : From human-computer interactions. In *Proceedings of the 5th International Conference on Human System Learning, ICHSL'05*, pages 287–302, Marrakech, Morocco, 2005.

[44] T. H. Dang. Modèle hiérarchique des mesures de discrimination floues. In *Actes des Rencontres francophones sur la Logique Floue et ses Applications, LFA'06*, pages 21–28, Toulouse, France, 2006.

[45] T. H. Dang, B. Bouchon-Meunier, and C. Marsala. Measures of information for inductive learning. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'04*, pages 1495–1502, Perugia, Italy, 2004.

[46] T. H. Dang and T. Delavallade. Utilisation de l'entropie pour substituer des données manquantes symboliques dans un problème de classification supervisée. In *Actes des 4ème Journées Nationales sur les Systèmes Intelligents : Théories et Applications, SITA'06*, pages 45–54, Mohammedia, Maroc, 2006.

[47] T. H. Dang and C. Marsala. Extension of hierarchical model for fuzzy measures of discrimination. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'06*, pages 1284–1291, Paris, France, 2006.

[48] T. H. Dang, C. Marsala, B. Bouchon-Meunier, and A. Boucher. Discrimination-based criteria for the evaluation of classifiers. In *Proceedings of the 7th International Conference on Flexible Query Answering Systems, FQAS'06*, volume 4027 of *LNAI*, pages 552–563, Milano, Italy, 2006.

[49] Z. Daróczy. Generalized information functions. *Information and Control*, 16 :36–51, 1970.

[50] J. K. de Fériet. Mesure de l'information fournie par un événement. In *Séminaire sur les questionnaires*, pages 1–27, 1971.

[51] J. K. de Fériet. L'indépendance des événements dans la théorie généralisée de l'information. In *Journées lyonnaises des questionnaires*, pages 1–30, 1975.

[52] R. L. de Mántaras. A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6 :81–92, 1991.

[53] T. V. de Merckt. Decision trees in numerical attribute spaces. In *Proceedings of the 13th International Joint Conferences on Artificial Intelligence, IJCAI'93*, pages 1016–1021, Chambery, France, 1993.

[54] T. Delavallade. Prévision de risques appliquée à la détection des conflits intra-étatiques. Rapport de pré-soutenance, Université Paris VI, France, 2006.

[55] T. Delavallade and T. H. Dang. Using entropy to impute missing data in a classification task. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'07*, pages 577–582, London, UK, 2007.

[56] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, (7) :1–30, 2006.

[57] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the International Conference on Machine Learning, ICML'95*, pages 194–202, San Francisco, USA, 1995.

[58] C. Drummond and R. C. Holte. Explicitly representing expected cost : an alternative to ROC representation. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining, KDD'00*, pages 198–207, 2000.

[59] C. Drummond and R. C. Holte. What ROC curves can't do (and cost curves can) ? In *Proceedings of the 1st Workshop on ROC Analysis in AI, ROCAI'04*, pages 19–26, 2004.

[60] C. Drummond and R. C. Holte. Cost curves : An improved method for visualizing classifier performance. *Machine Learning*, 65(1) :95–130, 2006.

[61] F. Esposito, D. Malerba, and G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5) :476–491, 1997.

[62] T. Fawcett. ROC graphs : Notes and practical considerations for researchers. Technical report, HP Labs Tech Report HPL-2003-4, 2003.

[63] U. M. Fayyad and K. B. Irani. What should be minimized in a decision tree ? In *Proceedings of the 8th National Conference on Artificial Intelligence, AAAI'90*, pages 749–754, Boston, Massachusetts, USA, 1990.

[64] U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8 :87–102, 1992.

[65] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI'93*, pages 1022–1027, Chambery, France, 1993.

[66] J. H. Friedman. A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computer*, 26 :404–408, 1977.

[67] J. Gama and P. Brazdil. Characterization of classification algorithms. In *Proceedings of the 7th Portuguese Conference on Artificial Intelligence, EPIA'95*, pages 189–200. Springer-Verlag, 1995.

[68] J. Goguen and L. Carlson. Axioms for discrimination information. *IEEE Transactions on Information Theory*, pages 572–574, 1975.

[69] J. W. Grzymala-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. In *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing, RSCTC'00*, pages 378–385, 2000.

[70] M. Ha and C. Zhang. A new algorithm for generating fuzzy decision trees. In *Proceedings of the International Fuzzy Systems Association World Congress, IFSA'05*, pages 1666–1671, Beijing, China, 2005.

[71] D. Hilbert and D. Redmiles. Extracting usability information from user interface events. *ACM Computing Surveys*, 32(4) :384–421, 2000.

[72] K. M. Ho and P. D. Scott. An efficient global discretization method. In *Proceedings of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD'98*, volume 1394 of *LNCS*, pages 383–384. Springer, 1998.

[73] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3) :289–300, 2002.

[74] S. Huet, A. Bouvier, M. Poursat, and E. Jolivet. *Statistical tools for nonlinear regression : a practical guide with S-PLUS examples*. Springer series in statistics. 1996.

[75] H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi. Neuro-fuzzy ID3 : a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning. *Fuzzy Sets and Systems*, 81(1) :157–167, 1996.

[76] R. Jalam and J. Chauchat. Pourquoi les n-grammes permettent de classer des textes ? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques. In *Actes des 6èmes Journées internationales d'Analyse statistique des Données Textuelles, JADT'02*, volume 1 of *Lexicometrica*, pages 381–390, St Malo, France, 2002.

[77] J. Jang. Structure determination in fuzzy modeling : A fuzzy CART approach. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'94*, pages 480–485, Orlando, Florida, USA, 1994.

[78] C. Janikow. Exemplar learning in fuzzy decision trees. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'96*, pages 1500–1505, 1996.

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

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

[81] C. Janikow and K. Kawa. Fuzzy decision tree FID. In *Proceedings of the 24th International Conference of the North American Fuzzy Information Processing Society, NAFIPS'05*, pages 379–384, 2005.

[82] R. Kessler, J. M. Torres-Moreno, and M. El-Beze. Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage. *Ingénierie des systèmes d'information*, 11(2) :93–112, 2006.

[83] B. Kim and D. Landgrebe. *Hierarchical decision tree classifiers in high-dimensional and large class data*. PhD thesis, School of Elec. Eng. Purdue Univ., West Lafayette, 1990.

[84] H. Kim, G. H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data : local least square. *Bioinformatics*, 21(2) :187–198, 2005.

[85] M.-W. Kim, J. W. Ryu, S. Kim, and J. G. Lee. Optimization of fuzzy rules for classification using genetic algorithm. In *Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD'03*, pages 363–375, 2003.

[86] G. Klir and T. Folger. *Fuzzy Sets, Uncertainty and Information*. Prentice Hall, 1988.

[87] I. Kononenko. On biases in estimating multi-valued attributes. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI'95*, pages 1034–1040, 1995.

[88] I. Kononenko and I. Bratko. Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6 :67–80, 1991.

[89] I. Koprinska, J. Poon, J. Clark, and J. Chan. Learning to classify e-mail. *Information Sciences*, 177(10) :2167–2187, 2007.

[90] K. B. Korb, L. R. Hope, and M. J. Hughes. The evaluation of predictive learners : Some theoretical and empirical results. In *Proceedings of the 12th European Conference on Machine Learning, ECML'01*, pages 276–287, London, UK, 2001. Springer-Verlag.

[91] L. Kurgan and K. Cios. Fast class-attribute interdependence maximization (CAIM) discretization algorithm. In *Proceedings of the International Conference on Machine Learning and Applications, ICMLA'03*, pages 30–36, Los Angeles, California, USA, 2003. CSREA Press.

[92] S. Lallich, P. Lenca, and B. Vaillant. Construction d'une entropie décentrée pour l'apprentissage supervisé. In *Actes du 3ème Atelier Qualité des Connaissances à partir des Données, QDC-EGC'07*, pages 45–54, Namur, Belgique, 2007.

[93] G. H. Landeweerd, T. Timmers, E. S. Gelsema, M. Bins, and M. R. Halie. Binary tree versus single level tree classification of white blood cells. *Pattern Recognition*, 16(6) :571–577, 1983.

[94] K.-M. Lee, K.-M. Lee, J.-H. Lee, and H. Lee-Kwang. A fuzzy decision tree induction method for fuzzy data. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'99*, pages 16–21, 1999.

[95] G. Legrand and N. Nicoloyannis. Data preprocessing and Kappa coefficient. In *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC'05*, volume 3641 of *LNCS*, pages 176–184, Regina, Canada, 2005. Springer.

[96] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40(3) :203–228, 2000.

[97] R. J. A. Little and D. B. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[98] H. Liu, F. Hussain, C.L.Tan, and M. Dash. Discretization : An enabling technique. *Journal of Data Mining and Knowledge Discovery*, pages 393–423, 2002.

[99] H. Liu and R. Sentiono. Chi2 : Feature selection and discretization of numeric attributes. In *Proceedings of the IEEE International Conference Tools with Artificial Intelligence 7*, pages 338–391, 1995.

[100] S. Marcellin, D. A. Zighed, and G. Ritschard. An asymmetric entropy measure for decision trees. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'06*, pages 1292–1299, Paris, France, 2006.

[101] S. Marcellin, D. A. Zighed, and G. Ritschard. Detection of breast cancer using an asymmetric entropy measure. In *Proceedings of the 17th Symposium on Computational Statistics, COMPSTAT 2006*, pages 975–982, Rome, Italy, 2006.

[102] C. Marsala. *Apprentissage inductif en présence de données imprécises : construction et utilisation d'arbres de décision flous*. PhD thesis, Université Paris VI, France, 1998.

[103] C. Marsala and B. Bouchon-Meunier. Fuzzy partioning using mathematical morphology in a learning scheme. In *Proceedings of the 5th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'96*, pages 1512–1517, New Orleans, USA, 1996.

[104] C. Marsala and B. Bouchon-Meunier. Construction methods of fuzzy decision trees. In *Proceedings of the 4th Joint Conference on Information Sciences, JCIS'98*, pages 17–20, 1998.

[105] C. Marsala and B. Bouchon-Meunier. Choice of a method for the construction of fuzzy decision trees. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'03*, pages 584–589, St Louis, USA, 2003.

[106] C. Marsala and B. Bouchon-Meunier. Selection of attribute for fuzzy decision trees. In *Proceedings of the Workshop on Soft Computing for Information Mining, 27th conference on Artificial Intelligence*, Ulm, Germany, 2004.

[107] C. Marsala, B. Bouchon-Meunier, and A. Ramer. Hierarchical model for discrimination measures. In *Proceedings of the 8th the International Fuzzy Systems Association World Congres, IFSA'99*, pages 339–343, Taipei, Taiwan, 1999.

[108] J. Mingers. An empirical comparison of selection measures for decision tree induction. *Machine Learning*, 3 :319–342, 1989.

[109] T. M. Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11) :30–36, 1999.

[110] S. K. Murthy. Automatic construction of decision trees from data : A multidisciplinary survey. *Data Mining and Knowledge Discovery*, 2(4) :345–389, 1998.

[111] S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2 :1–32, 1994.

[112] G. Neumann and S. Schmeier. Shallow natural language technology and text mining. *Künstliche Intelligenz*, 16(2) :23–26, 2002.

[113] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998.

[114] S. Oba, M.-A. Sato, I. Takemasa, M. Monden, K.-I. Matsubara, and S. Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16) :2088–2096, 2003.

[115] C. Olaru and L. Wehenkel. A complete fuzzy decision tree technique. *Fuzzy Sets and Systems*, (138) :221–254, 2003.

[116] W. Pedrycz and Z. A. Sosnowski. C-fuzzy decision trees. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 35(4) :498–511, 2005.

[117] Y. Peng and P. Flach. Soft discretization to enhance the continuous decision tree induction. In *Workshop notes : Integrating Aspects of Data Mining, Decision Support and Meta-Learning, IDDM'01 (in ECML/PKDD'01)*, pages 109–118, Freiburg, Germany, 2001.

[118] C.-F. Picard. *Théorie des questionnaires*. PhD thesis, Université des Sciences Mathématiques, Paris, France, 1963.

[119] C.-F. Picard. *Graphes et questionnaires*. North Holland, 1980.

[120] F. J. Provost and T. Fawcett. Analysis and visualization of classifier performance : Comparison under imprecise class and cost distributions. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining, KDD'97*, pages 43–48, 1997.

[121] F. J. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15th International Conference on Machine Learning, ICML'98*, pages 445–453, San Francisco, CA, USA, 1998.

[122] J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1) :81–106, 1986.

[123] J. Quinlan. Probabilistic decision trees. *Machine Learning*, 3 :140–152, 1990.

[124] J. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc, 1993.

[125] S. Rabaséda. *Contributions à l'extraction automatique de connaissances : application à l'analyse clinique de la marche*. PhD thesis, Université Claude Bernard, Lyon I, France, 1996.

[126] S. Rabaséda-Loudcher, M. Sebban, and R. Rakotomalala. Discretization of continuous attributes : a survey of methods. Technical report, Université Lumière Lyon 2, 1996.

[127] R. Rakotomalala. Arbres de décision. *Revue MODULAD*, 33 :163–187, 2005.

[128] R. Rakotomalala, S. Lallich, and S. D. Palma. Studying the behavior of generalized entropy in induction trees using a M-of-N concept. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD'99*, pages 510–517, 1999.

[129] M. Ramdani. Une approche floue pour traiter les valeurs numériques en apprentissage. In *Actes des Journées Francophones d'Apprentissage et d'explication des connaissances*, Dourdan, France, 1992.

[130] M. Ramdani. *Système d'induction formelle à base de connaissances imprécises*. PhD thesis, Université Paris VI, France, 1994.

[131] A. Ramer, B. Bouchon-Meunier, and C. Marsala. Analytical structure of hierarchical discrimination. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'99*, pages 1050–1053, Seoul, Korea, 1999.

[132] A. Rényi. *Calcul des probabilités*. Dunod, Paris, France, 1966.

[133] M. Rifqi. *Mesures de comparaison, typicalité et classification d'objets flous : théorie et pratique*. PhD thesis, Université Paris VI, France, 1996.

[134] M. Rifqi, V. Berger, and B. Bouchon-Meunier. Discrimination power of measures of comparison. *Fuzzy Sets and Systems*, 110 :189–196, 2000.

[135] G. Ritschard. Arbre BIC optimal et taux d'erreur. In *Actes de l'Atelier qualité des données et connaissances, DKQ'05*, pages 57–64, Paris, France, 2005.

[136] G. Ritschard and D. A. Zighed. Goodness-of-fit measures for induction trees. In *Proceedings of the 14th International Symposium on Intelligent Systems, ISMIS'03*, volume 2871 of *Lecture Notes in Computer Science*, pages 57–64, Maebashi City, Japan, 2003. Springer.

[137] G. Ritschard and D. A. Zighed. Qualités d'ajustement d'arbres d'induction. *Revue des Nouvelles Technologies de l'Information*, pages 45–67, 2004.

[138] L. Rokach and O. Maimon. Top-down induction of decision trees classifiers - a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 35(4) :476–487, 2005.

[139] R. S. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(21) :660–674, 1991.

[140] C. Schaffer. A conservation law for generalization performance. In *Proceedings of the 11th International Conference on Machine Learning, ICML'94*, pages 259–265, 1994.

[141] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423 and 623–656, 1948.

[142] K.-S. Shin, H.-J. Kim, and S.-B. Kwon. A GA-based fuzzy decision tree approach for corporate bond rating. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence, PRICAI'04*, pages 505–514, 2004.

[143] P. Smets. Probability of a fuzzy event : An axiomatic approach. *Fuzzy Sets and Systems*, 7 :153–164, 1982.

[144] R. J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7(1) :1–22, 1964.

[145] F. Stermann and N. Longuet. Document technique de DTGen. Technical report, LIP6, France, 2003.

[146] R. Sundaresan. A measure of discrimination and its geometric properties. In *Proceedings of the IEEE International Symposium on Information Theory, ISIT'02*, pages 264–264, Lausanne, Switzerland, 2002.

[147] H. Tanaka, T. Okuda, and K. Asai. Fuzzy information and decision in statistical model. In *Advance in Fuzzy Set Theory and Applications*, pages 303–320, Holland, 1979.

[148] H. Theil. On the estimation of relationships involving qualitative variables. *American Journal of Sociology*, 76 :103–154, 1970.

[149] E. Tsang, D. Yeung, and X. Wang. OFFSS : optimal fuzzy-valued feature subset selection. *IEEE Transactions on Fuzzy Systems*, 11(2) :202– 213, 2003.

[150] M. Umano, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita. Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'94*, pages 2113–2118, Orlando, FL, USA, 1994.

[151] P. E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4 :161–186, 1989.

[152] P. E. Utgoff and J. A. Clouse. A Kolmogorov-Smirnoff metric for decision tree induction. Technical report, UM-CS-1996-003, Amherst, MA, USA, 1996.

[153] R. Vinot. *Classification automatique de textes dans des catégories non thématiques*. PhD thesis, ENST, France, 2004.

[154] X. Wang and C. Borgelt. Information measures in fuzzy decision trees. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'04*, pages 1269–1275, Budapest, Hungary, 2004.

[155] X. Wang, B. Chen, G. Qian, and F. Ye. On the optimization of fuzzy decision trees. *Fuzzy Sets and Systems*, 112(1) :117–125, 2000.

[156] X. Wang, D. D. Nauck, M. Spott, and R. Kruse. Intelligent data analysis with fuzzy decision trees. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 11(5) :439–457, 2007.

[157] X. Z. Wang, D. S. Yeung, and E. C. C. Tsang. A comparative study on heuristic algorithms for generating fuzzy decision trees. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 31(2) :215–226, 2001.

[158] R. Weber. Fuzzy-ID3 : a class of methods for automatic knowledge acquisition. In *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks*, pages 265–268, Iizuka, Japan, 1992.

[159] L. Wehenkel. On uncertainty measures used for decision tree induction. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'96*, pages 413–418, Granada, Spain, 1996.

[160] A. P. White and W. Z. Liu. Bias in information-based measures in decision tree induction. *Machine Learning*, 15(3) :321–329, 1994.

[161] I. H. Witten and E. Frank. *Data Mining : Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann Publishers, Inc, San Francisco, USA, 2005.

[162] Y. Yuan and M. J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, (69) :125–139, 1995.

[163] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338–353, 1965.

[164] L. A. Zadeh. Probability measures of fuzzy events. *Journal of Mathematical Analysis and Applications*, 23 :421–427, 1968.

[165] J. Zeidler and M. Schlosser. Continuous valued attributes in fuzzy decision trees. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'96*, pages 395–400, Grenada, Spain, 1996.

[166] A. Zighed, J. P. Auray, and G. Duru. *SIPINA Méthode et logiciel*. Edition Alexandre Lacassagne - Lyon, 1992.

[167] D. A. Zighed, M. Côté, and N. Troudi. The data-mining and the technology of agents to fight the illicit electronic messages. In *Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining, PAKDD'99*, pages 464–468, London, UK, 1999. Springer-Verlag.

[168] D. A. Zighed, S. Marcellin, and G. Ritschard. Mesure d'entropie asymétrique et consistante. In *Actes des Journées francophones d'Extraction et de Gestion des Connaissances, EGC'07*, pages 81–86, Namur, Belgique, 2007.

[169] D. A. Zighed, S. Rabaséda, and R. Rakotomalala. FUSINTER : A method for discretization of continuous attributes. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, pages 307–326, 1998.