Induction de règles floues interprétables

S. Guillaume

Rapport Laas N^o : 01592 **Année 2001**

Thèse

préparée au
Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

En vue de l'obtention du titre de
Docteur de l'Institut National des Sciences Appliquées de Toulouse

Spécialité : **Systèmes Informatiques**

Par
Serge GUILLAUME
Ingénieur au *Cemagref*,
guillaume@montpellier.cemagref.fr

Induction de règles floues interprétables

Soutenue le 14 novembre 2001 devant le jury :

Président :	Didier DUBOIS, DR CNRS, IRIT, Toulouse
Rapporteurs :	Bernadette BOUCHON-MEUNIER, DR CNRS, LIP6, Paris
	Pierre-Yves GLORENNEC, Professeur, INSA, Rennes
Examinateurs :	Luis MAGDALENA, Professeur, UPM, Madrid
	Bruno MAILLIARD, Ingénieur Agronome, Cave d'Arzens
	Jean-Michel ROGER, Ingénieur, Cemagref, Giqual, Montpellier
	Brigitte CHARNOMORDIC, Ingénieur, INRA, LASB, Montpellier
Directeur de thèse :	André TITLI, Professeur, INSA, Toulouse

Cette thèse a été préparée conjointement dans les laboratoires :

Giqual

LASB

LAAS

Thèse INSA, École doctorale systèmes, option systèmes informatiques

Nom : Serge GUILLAUME

Date : 14 novembre 2001

Titre : Induction de règles floues interprétables

Résumé :

L'objectif du travail présenté dans ce mémoire est l'induction de règles floues interprétables à partir de données dans le but de la coopération homme machine. Dans l'état de l'art que nous avons réalisé, les méthodes d'induction de règles floues sont organisées en trois familles. Leur comparaison montre que l'interprétabilité n'est pas garantie par le formalisme flou.

La partie principale de ce mémoire décrit la méthode d'induction de règles floues que nous proposons. Elle vise à satisfaire trois conditions d'interprétabilité : lisibilité du partitionnement, nombre de règles minimal, règles incomplètes. La procédure est décomposée en trois étapes : une phase intradimensionnelle pour générer une famille de partitions par variable d'entrée, une composition multidimensionnelle pour construire un premier système performant, et une simplification de la base de règles.

Elle s'appuie sur des concepts originaux tels qu'une distance entre observations qui prenne en compte la structure de la partition, ou encore le contexte défini par un groupe de règles. Elle est guidée par des indices, indice de couverture et d'hétérogénéité, que nous avons introduits en complément de l'index de performance numérique.

Après une validation sur des exemples connus, la méthodologie est appliquée à la conception d'un système d'aide à la décision. Il s'agit d'induire les actions de conduite, sous forme de règles, qui accentuent la couleur du vin rouge au cours de la vinification.

Mots clés :

Système d'inférence floue, partitionnement, distance, interprétabilité, induction de règles, apprentissage, procédé agro-alimentaire, vinification.

Title : Induction of interpretable fuzzy rules

Abstract :

This report deals with interpretable fuzzy rule induction from data for human-computer cooperation purposes. A review of fuzzy rule induction methods shows that they can be grouped into three families. Their comparison highlights the fact that the interpretability is not guaranteed.

The central part of our work is a new fuzzy rule induction method. It aims to fulfill three interpretability conditions : readable fuzzy partitions, a number of rules as small as possible, incomplete rules. This is achieved through a three step procedure : generating a family of fuzzy partitions for each input variable, building an accurate fuzzy inference system, simplifying the rule base.

The procedure is based on original concepts such as a metric distance suitable for fuzzy partitioning, and the input context defined by a set of rules. We introduced coverage and heterogeneity related indices to guide the prodedure, complementary with a numerical performance index.

The method is first validated using well known data and then applied to decison making in a complex system. This application means to extract winemaking rules which enhance the color of red wine.

Key words :

Fuzzy Inference System, Partitioning, Distance, Interpretability, Rule Induction, Machine Learning, Food Processing, Winemaking.

Table des matières

Table des figures

Liste des tableaux

Avant propos

Pour moi, je suis bien persuadé que ce n'est pas dans les cendres du temps mais dans les dangereuses flammes de l'événement que naissent les images valables de l'homme, dût celui qui a l'audace de les y arracher s'en brûler affreusement les mains, en être défiguré, en périr.

Louis Aragon

(Aragon, 1973)

J'ai eu la chance de faire une thèse conforme à mes souhaits : bien ancrée dans une discipline et appuyée sur des applications.

La tentation de produire un mémoire qui serait en fait un bilan d'activités a été rapidement écartée : c'était l'objet du rapport écrit en 1994 pour obtenir, par validation des acquis, le titre d'ingénieur diplômé par l'état.

Le DEA d'informatique, suivi en 1998, et l'année passée au laboratoire d'analyse des systèmes et de biométrie de l'INRA, m'ont ouvert des horizons nouveaux. L'objet de la thèse fut décidé au moment de la définition du stage intégré dans le DEA : au sein de l'intelligence artificielle, travailler à la production de connaissances à partir d'une base de données. La logique floue, interface naturelle entre le symbolique et le numérique, s'est imposée.

J'ai dû appréhender ce champ, complètement neuf pour moi, avec méthode. C'est ainsi que j'ai pu apprécier la puissance de la méthodologie scientifique. J'ai consacré un temps qui peut paraître important, près d'une année, à l'étude bibliographique. Je ne le regrette pas, ce travail constitue un investissement pérenne. Au delà de l'inventaire de l'existant, cette étude a rempli un double objectif. D'une part, elle a été un facteur important de mon intégration dans la communauté scientifique de la logique floue. L'autre facteur d'intégration, non moins important, a été ma participation régulière aux rencontres francophones de la logique floue et ses applications, les LFA. D'autre part, l'effort de synthèse de cette revue, qui s'est concrétisé par une publication, m'a permis de définir précisément le sillon que je voulais approfondir.

Le développement de ma contribution proprement dite a emprunté des chemins que je n'aurais pas imaginés il y a seulement quelque temps. Les principaux concepts, choix et directions de travail ont été écrits de façon détaillée avant d'être mis en œuvre. Ils sont d'abord le fruit d'un travail intellectuel, avant d'avoir été vérifiés expérimentalement. Je dois avouer que je retire de ce fait une certaine satisfaction. Cette rupture dans ma pratique professionnelle symbolise la transition entre une carrière technique d'environ 15 ans et le début d'une approche scientifique : j'essaie de ne pas tester sur le clavier le premier bout d'idée.

C'est dans ce contexte que je me suis interrogé sur quelques questions régulièrement posées au sein d'organismes tels que le Cemagref. Celui-ci utilise volontiers le terme d'ingénieur-chercheur. Après avoir exercé ces deux activités, je peux faire le constat qu'il est difficile, voire impossible, d'être chercheur et ingénieur au même moment. Tout sépare ces activités : la problématique, les méthodes de travail, les productions. La problématique du chercheur est souvent très précise : son travail s'inscrit dans un champ bien délimité, au sein d'une communauté, parmi une équipe qui approfondit un point particulier. L'ingénieur a un problème à résoudre ou des prestations à assurer, il doit pour ce faire mobiliser l'ensemble des connaissances et des moyens à sa disposition. Quand le chercheur a obligation de méthode, l'ingénieur a obligation de résultats. Le chercheur sera principalement jugé sur la qualité de ses publications, l'ingénieur sur celle de ses réalisations.

Par l'usage du terme ingénieur-chercheur, l'organisme affiche la volonté de conduire une activité de recherche appliquée. Il me semble qu'une telle activité doit éviter deux écueils. Le premier est caractérisé par la dérive "fondamentaliste" pour qui les applications ne sont qu'une illustration des méthodes, qui sont par nature universelles. De ce point de vue, extrémiste, si la méthode ne convient pas, la responsabilité en incombe à l'application qui n'a pas su s'adapter ! Le deuxième écueil consiste à vouloir seulement résoudre des applications. Le risque est alors de se contenter d'agréger des résultats de recherche acquis dans un domaine différent. Cette application de la recherche se rapproche davantage du travail d'ingénieur. Je pense qu'il existe une voie médiane qui consiste à identifier des problèmes de recherche à partir de certaines classes d'applications, et à conduire une réelle activité de recherche, au sens académique du terme, qui puisse aussi être validée par ces applications.

Les applications sont par nature interdisciplinaires. Au bout de ce court voyage de trois ans, le champ disciplinaire m'apparaît comme indispensable pour la formation scientifique. C'est sur ce terrain balisé que s'acquiert la méthodologie scientifique, que peut s'opérer l'évaluation par les pairs. Cependant, chacun a conscience qu'aucun problème n'a de solution intradisciplinaire pure. Le concept n'est jamais toute la chose. Les pratiques de recherche interdisplinaires que je connais demeurent décevantes, au mieux elles se réduisent à une confrontation de points de vue disciplinaires. Serait-il possible de régir l'espace interdisciplinaire comme une discipline, avec l'instauration, notamment, de critères scientifiques d'évaluation ?

Au terme de ce parcours, je souhaite sincèrement remercier les personnes qui ont accompagné ce travail. Les réunions semestrielles du comité de thèse ont été l'occasion de débats riches et animés. J'ai eu du plaisir à les vivre.

Merci à André Titli, qui, en sa qualité de directeur de thèse, a été le garant des grandes orientations de ce travail. Nous avons su trouver un rythme de travail satisfaisant et efficace. J'ai toujours reçu au Laas un accueil chaleureux tant de la part des personnels permanents que des thésards et stagiaires. J'adresse des remerciements particuliers à José Aguilar Martin pour m'avoir mis en contact avec ses amis espagnols.

Merci à Véronique Bellon-Maurel, responsable de mon laboratoire au Cemagref, sans qui rien n'aurait été possible. Merci particulièrement pour m'avoir accordé, sans aucune contre partie pour l'équipe, une année de disponibilité à l'INRA pour passer le DEA d'informatique. Merci également pour avoir compris mon point de vue quant à la façon de conduire ce travail de thèse. Merci aussi à l'ensemble des collègues du laboratoire Giqual pour le travail réalisé en commun et tous les bons moments partagés.

Merci à Jean-Michel Roger, avec qui, au delà de ce travail, nous avons surtout vécu l'expérience, très riche, de l'enseignement de l'analyse des données. Je dois avouer que je n'aurais pas osé m'y lancer tout seul !

Merci à Jean-Pierre Vila, responsable du laboratoire INRA dans lequel je suis accueilli depuis quatre années maintenant, qui a apporté sur ce travail son regard de mathématicien et son expérience. Ces remerciements s'adressent également à l'ensemble des collègues de ce laboratoire, et plus précisément aux informaticiens, Pascal Neveu, Philippe Vismara et Luc Menut, qui ont toujours été de bon conseil.

Merci à Jean-Philippe Steyer, qui a rejoint le groupe en cours de parcours, et a toujours su se rendre disponible.

Merci à Bruno Mailliard, œnologue de la cave coopérative d'Arzens, pour l'enthousiasme dont il a fait preuve pour le montage du projet de collaboration. Le savoir faire en vinification exposé dans ce mémoire est le sien. Sa mise en forme est le résultat d'explications patientes et d'une analyse, réalisée en commun, des règles induites.

Merci à Bernadette Bouchon-Meunier et à Pierre-Yves Glorennec pour avoir accepté de rapporter ce travail. Merci à Didier Dubois pour avoir accepté de présider le jury.

Merci à Luis Magdalena qui m'accueillera prochainement au sein de son laboratoire à l'université polytechnique de Madrid. Notre collaboration s'annonce des plus fructueuses.

Merci enfin à Brigitte Charnomordic, mon équipière du quotidien. Permets moi d'essayer de dire combien j'ai apprécié de travailler avec toi. Tes compétences vont bien au delà de celles, unanimement

reconnues, en informatique. J'ai été particulièrement sensible à ta disponibilité : dans une dynamique aussi intense, il est précieux, voire essentiel, de pouvoir compter sur des retours rapides. Cette célérité n'a rien sacrifié à la qualité de l'accompagnement. J'ai pu mesurer combien tu as su être respectueuse des chemins que j'avais choisis tout en me permettant d'aller au bout de la démarche. Accompagner c'est parfois simplement poser une question : "es-tu sûr que ton objectif soit bien celui-là ?", c'est aussi être capable de passer beaucoup de temps sur l'analyse des résultats pour affiner des réglages ou étudier le comportement de la méthode en développement. C'est enfin, du fait d'une position suffisamment proche mais tout de même distante, donner la petite impulsion qui permet de sortir d'un minimum local, ce qui se produit quand on a le nez trop collé à la vitre.

Nous avons rapidement acquis une vision commune de nos objectifs et de notre pratique professionnelle. Le bilan de ces quatre ans est tout simplement une collaboration sans tache malgré la diversité, et le volume !, des activités gérées ensemble : le travail méthodologique de la thèse bien sûr et les publications associées, mais aussi la composante programmation, la conception d'interfaces utilisateur, le recrutement et la gestion des stagiaires et personnels temporaires, la conduite des projets applicatifs, les relations parfois tendues avec les partenaires ... C'est rare, c'est efficace et c'est appréciable.

J'avais déjà eu la chance de rencontrer une telle complicité professionnelle : dans les années 1991-1994 en faisant équipe avec Frédéric Ros. Chargé de la gestion technique du projet qui était le support de sa thèse, j'ai été partiellement associé à son travail de recherche. Ces liens perdurent. Merci Frédéric pour avoir aimablement investi dans la critique de mon travail.

Seul comptable des imperfections du contenu de ce mémoire, je tiens à remercier Vincent Fromion et Brigitte Charnomordic pour leur relecture attentive. Leurs remarques ont grandement amélioré la lisibilité du document.

Ce mémoire est la trace de trois années de travail mais aussi de trois années de vie. La vie professionnelle ne peut pas être réduite au travail, elle a pour moi une dimension sociale et relationnelle forte. Je gère la dimension sociale principalement au sein du syndicat, et j'ai choisi mon ami Jean-Louis Vigneau pour représenter tous les collègues avec qui j'ai fait un bout de chemin. C'est avec Jean-Louis que je partage, depuis plus de 10 ans, le souci du syndicat au sein du Cemagref, mais aussi, plus largement, dans la communauté de la recherche et de l'enseignement supérieur de Montpellier et dans le ministère de l'agriculture au niveau national.

Les repas figurent parmi les moments agréables de la journée, ils sont l'occasion d'échanges variés. Je les ai principalement partagés avec Michèle Egéa et Nathalie Gorretta au Cemagref, Brigitte Charnomordic, Nadine Hilgert et Vincent Fromion à l'INRA.

Ma vie professionnelle a eu ces derniers temps, et aura encore plus dans un futur proche, une composante internationale. Les possibilités de contact et d'échange sont une des richesses de notre métier. Mes pensées vont du Québec à l'Espagne, avec un petit détour par l'Australie.

La vie c'est aussi tout ce qui se passe hors du boulot, et j'essaie d'être un citoyen du monde. Ces trois années correspondent à la naissance du mouvement ATTAC et à son émergence sur le devant de la scène. Il milite pour une mondialisation solidaire en clamant, à l'opposé de la doctrine néo-libérale, que le monde n'est pas une marchandise. A l'échelle du village, notre implication dans la vie sociale et associative de Prades nous permet de rencontrer des gens formidables. Le combat pour la justice et la fraternité, tout comme, dans un autre registre, les sorties en VTT, contribuent à tisser des liens d'amitié solides.

La famille que nous formons avec Bernadette, depuis plus de 20 ans, Elsa, Violaine et Loïc est source d'un bonheur toujours renouvelé.

Chapitre 1

Introduction

Le mathématicien américain John von Neumann (1903 - 1957) s'est posé la question de la différence entre les machines artificielles, qui s'usent, et les machines vivantes, qui se régénèrent. Il est paradoxal que les éléments de machines artificielles, très bien usinées et très perfectionnées, se dégradent dès que la machine commence à fonctionner et que les machines vivantes, composées d'éléments très peu fiables comme les protéines, qui ne cessent de se dégrader, possèdent l'étrange propriété de se développer, de se reproduire, de s'auto-régénérer en remplaçant, justement, les molécules dégradées par de nouvelles et les cellules mortes par des cellules neuves. La machine artificielle ne peut se réparer elle-même alors que la machine vivante se régénère en permanence, à partir de la mort de ses cellules, selon le célèbre aphorisme d'Héraclite : "Vivre de mort, mourir de vie".

Arnaud Spire

(Spire, 1999)

Le travail présenté dans ce mémoire vise à l'induction de règles floues interprétables à partir de données dans le but de la coopération homme machine. Tout en présentant une composante méthodologique forte, bien située dans une dynamique de la communauté scientifique de la logique floue, notre travail s'efforce de répondre à une problématique formulée par toute une classe d'applications dont les procédés agro-alimentaires sont un représentant typique. La maîtrise des procédés agro-alimentaires repose largement sur l'expertise humaine. En effet, ces procédés sont complexes et les opérations unitaires qui les composent sont elles-mêmes complexes et mal modélisées. Les difficultés de modélisation et de quantification des interactions sont un obstacle à leur compréhension, et à leur automatisation. Pour la conduite, les experts utilisent des règles du type *Si Situation Alors Conclusion* qui manipulent des connaissances souvent imprécises, par exemple *Si le produit est très doré dans cette partie du four Alors baisser un peu la température dans la zone précédente.* Les besoins, croissants, de simulation s'expliquent par l'objectif de qualité des produits. L'amélioration est toujours recherchée, mais il est également important de garantir une certaine constance dans la qualité. Ces besoins s'expliquent aussi par la volonté des experts d'accroître leur intelligence du procédé.

Les systèmes experts, très populaires jusqu'aux début des années 1990, encore appelés systèmes à base de connaissances, visaient à reproduire le comportement de l'expert du procédé à des fins de simulation et de conduite. Le cœur du système expert, le moteur d'inférence, est constitué d'une base de règles utilisant la logique booléenne de la forme *Si variable i est inférieure à seuil j Alors conclusion k.* Il était, dans ces conditions, difficile de reproduire la progressivité des phénomènes et l'imprécision des connaissances. De plus, la présence de seuils rendait le comportement du système sensible à de faibles perturbations. Toutes ces raisons font que ces systèmes experts ne sont quasiment plus utilisés aujourd'hui.

La logique floue (Zadeh, 1965) excelle dans la représentation de connaissances imprécises ou incomplètes. Elle constitue une interface commode pour la modélisation du langage naturel, en particulier des concepts linguistiques utilisés par les experts d'un procédé tels que *chaud*, *un peu* ou *très bon.* Son utilisation dans les systèmes experts a permis de dépasser les limites de la logique du tout ou rien. Les systèmes experts flous représentent un premier type de systèmes d'inférence floue. Ils ont été introduits par (Mamdani & Assilian, 1975). Comme les règles sont construites à partir du savoir faire des experts, qui résulte d'une accumulation d'expériences, ce type de système d'inférence floue présente à la fois un haut niveau d'interprétabilité et de bonnes performances en généralisation. Ces systèmes experts flous, qui ont permis de travailler de façon graduelle, ont aussi mis en évidence les limites de l'expertise humaine, et en particulier la difficulté d'évaluer les interactions. De fait, leur utilisation est limitée à des systèmes simples impliquant un petit nombre de variables. Pour la gestion des systèmes complexes, l'expert se heurte à la difficulté de formalisation des règles. Par conséquent, les simulateurs conçus à partir de la seule expertise humaine présentent souvent des performances très moyennes, pour ces systèmes complexes, en terme d'écart entre les sorties observées et les sorties simulées.

Les données expérimentales constituent une source de connaissance complémentaire. L'apprentissage à partir de données permet de reproduire les relations qui existent entre les entrées et la sortie d'un système. Les réseaux de neurones sont les représentants les plus connus des outils disponibles. Les systèmes d'inférence floue, qui sont aussi des approximateurs universels, peuvent être utilisés de façon analogue. Dans ces systèmes, les relations entre l'entrée et la sortie sont exprimées sous la forme de règles floues. Sugeno (Takagi & Sugeno, 1985) a été l'un des premiers à proposer de tels systèmes conçus à partir des données. Ils représentent un second type de systèmes d'inférence floue. L'induction de règles floues est une forme particulière d'apprentissage, apte à formaliser une image des interactions à partir des données disponibles.

Du fait de leur double identité, capacité à modéliser le langage naturel et approximateurs universels, les systèmes d'inférence floue pourraient permettre de combiner les avantages de ces deux types de connaissance. Mais, guidés par la seule amélioration de la performance numérique les systèmes d'inférence floue risquent de perdre leur spécificité. Dubois et Prade ont constaté : "Les contrôleurs flous et les méthodes d'induction de règles floues, qui constituent le côté le plus visible de la théorie des ensembles flous, ne représentent en réalité que la partie émergée de l'iceberg de cette théorie. Plus le temps passe, plus cette technologie s'éloigne du flou pour devenir simplement un outil d'approximation de fonctions." (Dubois & Prade, 1997).

L'objectif que nous fixons à l'induction est non seulement de reproduire avec précision les relations entrée-sortie, mais également de dégager des éléments de connaissance sur le fonctionnement du processus. L'interprétabilité peut être vue comme une contrainte qui permet de passer continûment d'un système de type boîte noire vers un système expert flou. Si elle est complètement relâchée, le système d'inférence floue se comporte comme un approximateur et vise simplement à améliorer la performance numérique, sinon, au prix d'une dégradation contrôlée de cet indice, nous pouvons espérer mettre en lumière des éléments de réflexion. Ceux-ci peuvent permettre d'économiser le recueil d'expertise qui est toujours coûteux et difficile. En effet, même si les règles sont bien connues, leur formalisation n'est pas triviale. Les experts ont, par exemple, de la difficulté à définir les limites des ensembles flous qui modélisent les concepts linguistiques qu'ils utilisent. Quand les règles sont moins connues, comme pour la gestion des interactions, l'induction peut compléter le savoir-faire des experts. La lisibilité de la base des règles induites est indispensable pour que les systèmes d'inférence floue offrent un cadre de coopération entre l'expertise et la connaissance contenue dans les données.

Trois conditions nous semblent nécessaires pour rendre les règles induites interprétables. En premier lieu le partitionnement doit être lisible pour les experts, les ensembles flous doivent correspondre à des termes linguistiques, qui ont un sens pour le procédé. Ainsi les règles peuvent être comparées. Ensuite, le nombre de règles doit être minimal. Cette réduction s'accompagne inévitablement d'une baisse des performances numériques en apprentissage mais conduit à des systèmes plus robustes, présentant de meilleures capacités de généralisation. Enfin, pour les systèmes impliquant un grand nombre de variables, une troisième condition est requise : les règles générées doivent être incomplètes, définies par les seules variables influentes. La présence systématique de l'ensemble des variables dans la définition des prémisses de chacune des règles doit être considérée comme un inconvénient des méthodes d'induction de règles et non pas comme une caractéristique intrinsèque du problème.

La sélection des variables fera l'objet d'une attention particulière. Cet aspect de l'induction de règles est essentiel pour notre classe d'applications. Cependant, il est peu abordé dans la littérature car les applications traitées sont souvent académiques ou impliquent un petit nombre de variables. Les bases de données dont nous sommes susceptibles de disposer n'ont pas été conçues pour nos besoins spécifiques. Toutes les variables enregistrées ne sont pas pertinentes pour le problème étudié. Dans les industries agro-alimentaires, par exemple, les données de fabrication sont souvent enregistrées de façon systématique à des fins de traçabilité ou de contrôle qualité. Elles décrivent le produit au cours de sa transformation et contiennent, mais implicitement, les choix de conduite effectués par l'expert et la conséquence de ces choix sur le produit. L'extraction de connaissances relatives à la conduite du procédé ne peut faire l'impasse sur la sélection des variables pertinentes.

Dans cette thèse, nous voulons développer une méthode d'induction de règles floues qui s'applique à des systèmes complexes. La génération de règles est en soi un problème qui n'est pas simple. Toutes les relations entrée-sortie ne peuvent pas être représentées par des règles interprétables, même dans le cadre de petits systèmes. Les applications que nous voulons traiter, les procédés

agro-alimentaires notamment, présentent des particularités qui compliquent cette tâche. L'expert est au cœur du système, il en fait même partie. Ses décisions sont prises à partir d'une multitude d'informations qui ne sont pas à notre disposition. Les données, dans l'immense majorité des cas, ne représentent que des mesures indirectes, partielles et très imparfaites, des phénomènes observés par l'expert et intégrés dans sa prise de décision. La contrainte d'interprétabilité que nous imposons rend l'objectif encore plus ambitieux.

Le mémoire est composé de quatre chapitres principaux. Le chapitre 2 propose une revue des principales méthodes d'induction de règles guidée par le critère de l'interprétabilité. L'utilisation de ce critère est motivée par le fait qu'historiquement les méthodes de génération de règles sont issues de l'apprentissage automatique. L'efficacité de l'apprentissage est souvent mesurée par un indice purement numérique tel que l'erreur quadratique moyenne. De plus, si les descriptions des méthodes sont nombreuses, les comparaisons, incluant les domaines d'application des différentes familles, sont rares. Ce travail est maintenant publié (Guillaume, 2001). Il propose une classification des méthodes d'induction en trois familles principales, et souligne l'importance de la construction du partitionnement, c'est-à-dire du découpage des variables en sous-ensembles flous.

Prenant appui sur l'état de l'art, notre méthode d'induction est présentée dans les deux chapitres suivants. Le chapitre 3 traite du partitionnement des entrées. Pour chacune des variables d'entrée, une famille hiérarchique de partitions floues est construite. La lisibilité du partionnement est une contrainte de la méthode, elle est garantie pour chaque partition de la hiérarchie. La construction s'appuie sur un concept de distance entre points qui prend en compte la structure de la partition.

Le chapitre 4 montre comment utiliser les familles de partitions monodimensionnelles pour concevoir un système d'inférence flou qui soit à la fois performant et interprétable. Des éléments d'appréciation de la lisibilité sont proposés en complément des classiques indicateurs de performance numérique. La méthodologie est validée à partir de deux exemples bien connus dans la littérature : les iris de Fisher et la relation entre la qualité globale d'un riz et certaines appréciations sensorielles telles que l'arôme, l'apparence, le collant, le goût ou encore la dureté. Cette dernière application a été introduite par (Ishibuchi et al., 1994).

Dans le chapitre 5, la méthode est appliquée à la conception d'un système d'aide à la décision pour améliorer la couleur au cours de la vinification des vins rouge. L'objectif est de produire, à partir des données, des règles opérationnelles et de les mettre à la disposition de l'œnologue. Cette application, en collaboration avec la cave coopérative d'Arzens, dans l'Aude, fait l'objet d'un contrat. Seules quelques étapes du projet sont réalisées : première collecte et traitement des données, induction de règles floues à l'aide de la méthode décrite dans ce mémoire. Les résultats sont encourageants et montrent le potentiel de la démarche.

Enfin, le chapitre de conclusion rappelle les principaux acquis, et souligne les points qu'il nous paraît important de développer.

Le lecteur peu familier avec la logique floue aura avantage à consulter préalablement le glossaire, annexe A, qui définit les principaux termes utilisés dans ce mémoire[a]. Un glossaire analogue, annexe B, est dédié à la vinification. L'annexe C décrit le logiciel réalisé dans le cadre de ce travail. Les principales publications liées à ce mémoire sont indiquées dans l'annexe D.

[a]Pour une introduction plus complète à la logique floue le lecteur pourra se reporter aux ouvrages suivants : (Bouchon-Meunier, 1993; OFTA, 1994; Bouchon-Meunier, 1995; Gacogne, 1997).

Chapitre 2

Induction de règles floues : une revue des méthodes disponibles suivant le critère de l'interprétabilité

La connaissance est le seul bien qu'on puisse partager sans rien en perdre.

Jacques Bonnet

(Bonnet, 2000)

Sommaire

L'induction de règles est le processus qui conduit à la formalisation, sous forme de règles, des relations, apprises à partir d'un ensemble d'exemples, qui existent entre les entrées et les sorties d'un système. Les principales méthodes d'induction de règles floues décrites dans la littérature sont présentées dans ce chapitre.

La plupart des méthodes d'induction ont pour but l'approximation de fonction, ou plus généralement de relations entre l'entrée et la sortie d'un système. Dans le cadre de l'apprentissage supervisé, cette approximation consiste à définir la base de règles, c'est-à-dire leur nombre, leurs prémisses et leurs conclusions, qui minimisent l'écart entre les sorties désirées et celles inférées par le système formé de l'ensemble des règles[a].

Comme les systèmes d'inférence floue sont des approximateurs universels (Wang, 1992), les méthodes d'induction ont été principalement pilotées et évaluées par rapport à ce seul critère de performance numérique. L'histoire a montré que les systèmes d'inférence floue sont aussi performants que d'autres techniques d'approximation.

L'objectif d'extraction de connaissance nous impose une propriété supplémentaire : le respect de la sémantique, les sous-ensembles flous doivent pouvoir être interprétés en termes linguistiques. Celle-ci n'est pas garantie par l'utilisation du formalisme flou.

La figure 2.1 illustre notre préoccupation. Dans la partie inférieure de la figure le recouvrement entre les ensembles flous est tel qu'ils peuvent être ordonnés, et donc interprétés en termes linguistiques, par exemple de *trs petit* vers *trs grand*. La partie supérieure montre un bel exemple de partition induite ininterprétable : il est impossible d'étiqueter les trois sous-ensembles flous centraux avec des labels linguistiques.

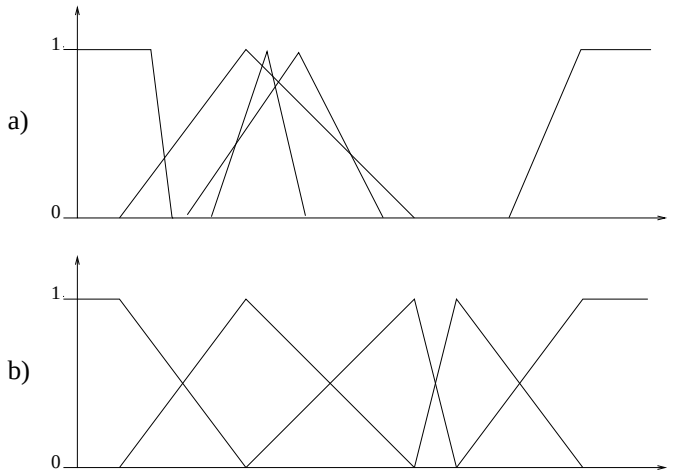


FIG. 2.1 – La partition supérieure ne peut s'interpréter en termes linguistiques

Aujourd'hui chacun s'accorde sur le fait que pour obtenir des règles interprétables, l'induction doit être contrainte. Certaines des méthodes disponibles intègrent quelques unes de ces contraintes.

Qu'elle soit interprétable ou pas, une base de règles est caractérisée par certaines propriétés : la cohérence, la continuité, la complétude.

[a]Lorsque la sortie de référence n'est pas disponible l'apprentissage est dit non supervisé. Des algorithmes d'apprentissage par renforcement pour les systèmes d'inférence floue peuvent être utilisés (Jouffe, 1997; Glorennec, 1999).

La **cohérence** est une propriété essentielle de la base de règles, quelle que soit l'utilisation du système. Elle signifie que les conclusions des règles simultanément activées, par un exemple donné, ne sont pas contradictoires. Cette formulation implique une vérification de la cohérence au niveau de la base de règle elle même. Dans le cadre de l'apprentissage, il est souhaitable de la vérifier au niveau de chacune des règles. Une règle reflète des incohérences si la distribution des valeurs de sortie des exemples activés par la règle est très hétérogène. Ces incohérences traduisent le fait que la règle n'est pas suffisamment spécifique pour rendre compte des données d'apprentissage. Les approximateurs universels divisent l'espace d'entrée, en ajoutant des règles, pour éliminer les incohérences. Lorsque une limite est atteinte, si la partie de l'espace d'entrée définie par la prémisse de la règle ne peut pas être réduite, alors nous considérons que l'incohérence est contenue dans les données.

La **continuité** indique que de petites variations sur l'entrée ne doivent pas impliquer de grandes variations sur la sortie.

Formellement, la **complétude** d'une base de règles assure que chacune des valeurs possibles de l'entrée active au moins une règle, qu'une valeur de sortie est inférée dans tous les cas. Comme indiqué dans le tableau 2.1, cette propriété est indispensable pour des utilisations autonomes, mais secondaire pour des applications en interaction avec un expert. Nous travaillons avec une définition moins contrainte. La complétude ne sera pas mesurée par rapport à l'ensemble des valeurs possibles de l'entrée, mais seulement par rapport à celles contenues dans l'ensemble d'apprentissage.

Les propriétés requises pour la base de règles varient en fonction des applications comme le montre le tableau 2.1. La complétude est indispensable pour les applications autonomes tandis que la sémantique est d'une importance cruciale pour la production de connaissances et pour la coopération homme machine.

Application	**Sémantique**	**Cohérence**	**Continuité**	**Complétude**
Commande[a]	+	+++	+++	+++
Classification	+	++	+	+++
Aide décision	+++	+++	+++	++
Diagnostic	+++	+++	++	+
Simulation	+++	+++	+++	+

TAB. 2.1 – Propriétés souhaitées de la base de règles en fonction des applications

Nous avons décomposé l'induction de règles en deux étapes : une étape de génération des règles, suivie d'une phase d'optimisation du système. Chacune sera abordée dans une partie de ce chapitre. Les méthodes de génération de règles sont regroupées en trois familles. Une fois décrites, elles sont comparées, selon leur domaine principal d'utilisation, en fonction des caractéristiques de l'espace de travail : dimension et niveau de couverture par l'ensemble d'apprentissage.

La **couverture** d'un espace de travail multidimensionnel par un ensemble d'apprentissage est traditionnellement représentée par la répartition des points de cet ensemble dans l'espace. Dans le cas d'une base de règles floues, chacune des règles est apte à couvrir une partie de l'espace, celle définie par la combinaison des ensembles flous de la prémisse et, éventuellement, de la conclusion, ou des conclusions de la règle. A chaque exemple de l'ensemble d'apprentissage, peut être associé le degré de vérité de chaque règle de la base, pour cet exemple[b]. Chaque exemple va donc couvrir, à un certain degré, la partie de l'espace correspondant aux règles qu'il active. Dès lors, il est possible

[b]Ces notions sont définies dans le glossaire de l'annexe A.

de définir un indice, ou un niveau de couverture, directement lié aux exemples actifs pour la base de règles. Un tel indice sera proposé dans la section 4.3.5.

Dans les applications réelles les systèmes sont souvent complexes. Pour extraire, à partir d'une base de données, des règles interprétables qui complètent le savoir faire des experts, une optimisation structurelle du système est nécessaire. Cette optimisation, bien différente d'un simple ajustement de paramètres, est étudiée à part entière dans notre état de l'art. Elle consiste en une sélection des variables, les principales méthodes sont présentées dans la section 2.5, et une réduction de la base de règles, section 2.6. La section 2.7 rappelle quelques méthodes d'optimisation paramétrique disponibles pour optimiser la forme des sous-ensembles flous ou bien la conclusion des règles. Enfin, pour conclure, les méthodes de génération et d'optimisation sont comparées suivant le critère de l'interprétabilité des règles.

A - Générer des règles

Nous proposons de classer les techniques d'induction de règles en trois groupes principaux. Deux de ces groupes sont homogènes, et à chacun d'entre eux correspondent des conditions d'utilisation préférentielles.

Le premier groupe utilise un partitionnement de l'espace multidimensionnel. Ce partitionnement définit un certain nombre d'ensembles flous pour chacune des dimensions, qui peuvent être interprétés comme des labels linguistiques et qui sont communs à l'ensemble des règles. La phase d'apprentissage consiste dans ce cas à optimiser le partitionnement et les conséquences des règles en fonction des données.

La deuxième famille est celle des méthodes de groupage ("clustering"). Les algorithmes de groupage flou sont utilisés pour organiser un ensemble de données. Leur application conduit à une partition des données en groupes homogènes suivant un certain critère. Le partitionnement de l'espace est dérivé de la partition des données et conduit à une règle par groupe dans le cas général. Donc, les ensembles flous résultants sont propres à une règle, ils ne sont pas partagés par l'ensemble des règles. L'interprétation des partitions floues, qui peuvent être du type de celle mentionnée dans la figure 2.1.a, est souvent difficile.

Entre ces deux pôles, de nombreuses variations sont possibles. Nous présenterons quelques méthodes, dites méthodes hybrides, qui mobilisent les techniques du soft computing. Notons toutefois qu'elles ne forment pas un groupe homogène et que leurs performances dépendent souvent de leur implémentation.

2.1 Partitionnement partagé

Pour obtenir de façon simple un partitionnement, il suffit de diviser le domaine de variation de chacune des dimensions en un nombre fixé d'intervalles dont les limites n'ont *a priori* aucune signification physique et ne tiennent pas compte de la fonction de répartition des données. Une grille est ainsi définie. Nous introduisons plusieurs de ces approches.

La méthode la plus intuitive pour générer des règles consiste alors à implémenter toutes les règles possibles correspondant à l'ensemble des combinaisons des sous-ensembles flous des variables d'entrée. Des améliorations permettent de gérer les inconvénients inhérents à cette méthode. Il se peut que certaines règles ne soient pas initialisées du fait d'un niveau de couverture de l'espace insuffisant : elles peuvent l'être par une procédure de diffusion.

Le choix du nombre d'ensembles flous par dimension $(K_1, K_2, \ldots, K_p)$ est délicat et lourd de conséquences : la deuxième approche permet de le choisir de façon dynamique soit par affinement de partition, soit à l'aide d'un algorithme génétique.

Le nombre de combinaisons peut devenir trop important, la troisième approche initialise une règle par exemple.

Enfin, les arbres de décision flous sont présentés à la fin de cette section. Ils permettent de générer des règles incomplètes mais nécessitent un partitionnement.

2.1.1 Toutes les règles possibles sont implémentées

Ishibuchi et al. (Ishibuchi et al., 1994) considèrent un système comprenant plusieurs entrées et une sortie. Les données sont ramenées dans un espace $[0,1]^p$ pour les entrées et $[0,1]$ pour la sortie. Chaque entrée i est divisée automatiquement en K_i ensembles flous, notés $A_i^1, A_i^2, \ldots, A_i^{K_i}$. La figure 2.2 illustre le procédé pour la variable x_1 avec $K_1 = 5$. La forme des fonctions d'appartenance peut être quelconque, la plus employée étant le triangle (Pedrycz, 1994) :

$$\mu_{ij}(x) = max\{1 - \frac{|x - a_i^j|}{b^{K_i}}, 0\} \quad avec \quad a_i^j = \frac{j-1}{K_i - 1} \quad j = 1, \ldots, K_i \quad et \quad b^{K_i} = \frac{1}{K_i - 1}$$

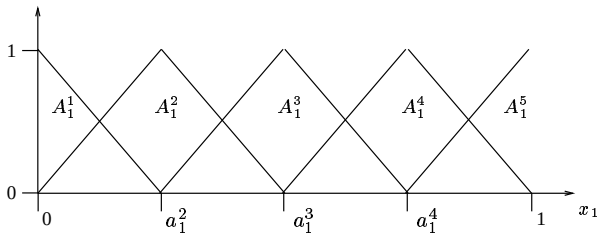


FIG. 2.2 – Partitionnement automatique en cinq sous-ensembles flous triangulaires

Toutes les règles correspondant à l'ensemble des combinaisons des entrées sont implémentées. Pour un système de p entrées, le nombre total de règles est donc : $K_1 \times K_2 \times \ldots \times K_p$.

Nozaki et al. (Nozaki et al., 1997), proposent une heuristique simple pour calculer les conclusions, nombre réels, des règles. Pour la règle i, la conclusion, notée b_i , est calculée comme :

$$b_i = \frac{\sum_{j=1}^{n} w_i(j) * y(j)}{\sum_{j=1}^{n} w_i(j)}$$

où $y(j)$ est la sortie observée pour l'exemple j et $w_i(j)$ le degré de vérité de la règle i pour l'exemple j.

La sortie inférée par le système pour l'exemple j s'écrit :

$$\widehat{y}(j) = \frac{\sum_{i=1}^{r} w_i(j) * b_i}{\sum_{i=1}^{r} w_i(j)} \tag{2.1}$$

Certains préconisent de ne considérer, pour le calcul de la sortie que les exemples qui activent le plus la règle.

Initialiser toutes les règles : En fonction du niveau de couverture de l'espace de travail par la base d'apprentissage et des choix faits pour les valeurs des K_i, il se peut que certaines règles ne soient activées par aucun exemple. Dans ce cas, Glorennec propose d'initialiser ces règles par diffusion (Glorennec, 1999).

Soit S l'ensemble des règles dont les conclusions ont été initialisées ; nous pouvons définir le voisinage de la règle i au sens de S, V_i étant le voisinage de la règle i, comme :

$$V_{S_i} = \begin{cases} \{i\} & si\ i \in S \\ V_i & sinon \end{cases}$$

V_i est généralement défini comme l'ensemble des règles dont la prémisse diffère de celle de i par un seul sous-ensemble flou. D'après cette définition, chacune des règles a au plus deux voisins par dimension, et donc le cardinal de V_i est borné : $|V_i| \leq 2^p$.

La diffusion se fait par la suite $g^{(n)}$ telle que :

$$\begin{aligned} g^{(0)}(i) &= g(i) \quad si\ i \in S \\ g^{(0)}(i) &= 0 \quad sinon \\ g^{(n+1)}(i) &= \frac{1}{2} \sup_{j \in V_{S_i}} g^{(n)}(j) + \frac{1}{2} \inf_{j \in V_{S_i}} g^{(n)}(j) \end{aligned}$$

Cette suite converge quand n tend vers l'infini, la méthode est stable.

2.1.2 Choix dynamique du nombre de sous-ensembles flous

Pour utiliser la méthode précédente, il est nécessaire de fixer le nombre d'ensembles flous par dimension $(K_1, K_2, \ldots, K_p)$. Ce choix est délicat et lourd de conséquences. S'il est trop faible le système aura de la difficulté à rendre compte de comportements non linéaires ; mais il est difficile de l'augmenter déraisonnablement pour les deux raisons suivantes. Tout d'abord, les ensembles flous correspondant risquent de devenir trop spécifiques et, ainsi, de réduire les aptitudes en généralisation du système. Ensuite, n'oublions pas que le nombre de règles du système est égal au produit de ces nombres $(\prod_i K_i)$.

Pour éviter de fixer le nombre d'ensembles flous de façon arbitraire, plusieurs auteurs proposent d'adapter la grille aux données, soit par un affinement de partition, soit par l'utilisation d'un algorithme génétique.

Affinement de partition

L'affinement de partition consiste en un processus itératif. La partition initiale est grossière, formée généralement de deux sous-ensembles flous par entrée, et à chaque étape un sous-ensemble flou est ajouté sur l'une des entrées du système. Plusieurs critères sont disponibles pour choisir la portion de l'espace qui doit être découpée plus finement. Les critères d'arrêt sont très importants, le risque étant l'introduction au sein du modèle du bruit contenu dans les données.

(Bortolet, 1998) propose d'ajouter des sous-ensembles flous sur les entrées qui sont responsables de la mauvaise performance du modèle. Au départ, chaque entrée est découpée en deux sous-ensembles flous, de forme triangulaire, centrés sur les valeurs extrêmes de la base de données. A chaque étape, toutes les règles correspondant aux combinaisons possibles des entrées sont prises en compte.

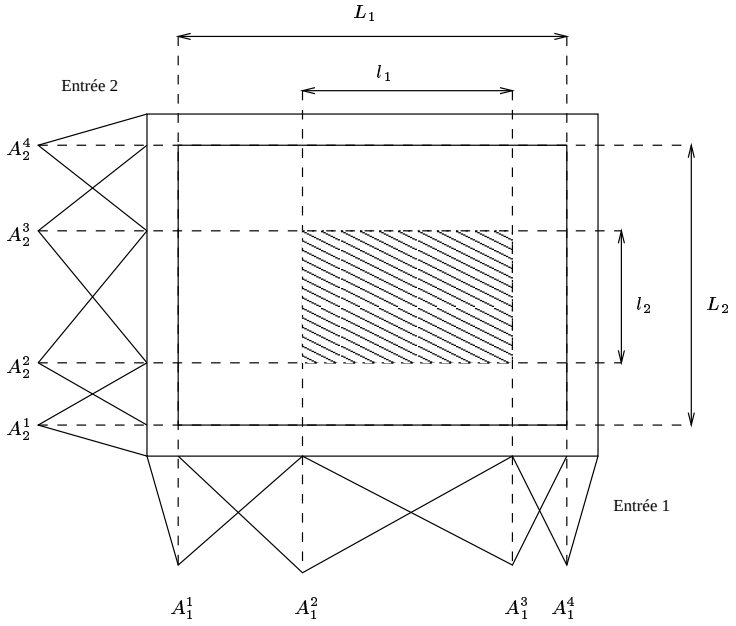
Le système est de type Takagi-Sugeno d'ordre zéro, la conséquence de chacune des règles est une constante. Elle est calculée en deux temps. Tout d'abord en estimant, par les moindres carrés, les coefficients de l'équation :

$$\widehat{y}(j) = a_0 + \sum_{i=1}^{p} a_i * x_i(j) \tag{2.2}$$

tels que l'écart entre $\widehat{y}$ *et* y soit minimal sur l'ensemble des exemples pris en compte. Seuls les exemples linéairement indépendants et dont le degré d'appartenance à chacun des sous-ensembles flous est supérieur à 0.5, sont pris en compte[c]. Ensuite, la conséquence de la règle i sera obtenue en remplaçant dans l'équation 2.2 les x_i par les valeurs des sommets des sous-ensembles flous pour lesquels le coefficient d'appartenance de x est maximum.

Algorithme : Le modèle, complètement défini, est appliqué à l'ensemble des exemples. Il convient maintenant d'ajouter un nouvel ensemble flou. Il faut pour cela identifier une région de l'espace d'entrée, puis une variable dans cette région et enfin déterminer le centre du nouveau sous-ensemble flou. Nous présentons maintenant les indicateurs utilisés pour réaliser ces choix.

Un indice d'erreur est associé à chacune des régions. Il est calculé comme le produit de l'erreur moyenne pour les exemples qui appartiennent à la région (n_i pour la région i) par la part du domaine d'entrée couverte par la région. Dans la zone où cette erreur est maximale, l'entrée sera sélectionnée de façon analogue. La figure 2.3 illustre les régions délimitées par des partitions de quatre ensembles flous dans un système de deux entrées. Elle indique les formules d'erreur correspondant à la région i, en grisé sur la figure, définie par les ensembles A_1^2, A_1^3, A_2^2 et A_2^3.



Erreur de la région i :

$$E_i = \frac{\sum_{k=1}^{n_i} |y_k - \widehat{y}_k|}{n_i} * \frac{l_1}{L_1} * \frac{l_2}{L_2}$$

Erreur de l'entrée j dans la région i :

$$E_i^j = \frac{\sum_{k=1}^{n_i} |y_k - \widehat{y}_k|}{n_i} * \frac{l_j}{L_j}$$

FIG. 2.3 – Affinement de partition

La région et l'entrée sélectionnées sont celles pour lesquelles les indices d'erreur sont les plus importants. La position du nouveau centre sur l'entrée j est donnée par la formule suivante :

[c]Si le nombre d'exemples est insuffisant pour effectuer la régression, i.e. si $nb < p + 1$, la région de collecte des exemples est élargie.

$$A_j^{new} = \frac{\sum_{k=1}^{n_i} x_{kj} |y_k - \widehat{y}_k|}{\sum_{k=1}^{n_i} |y_k - \widehat{y}_k|}$$

Les limites du support du nouveau sous-ensemble flou sont les centres des sous-ensembles modifiés, ceci permet de maintenir une partition floue forte.

Avant de passer à l'étape suivante, il faut recalculer les conséquences des règles dont les prémisses ont été modifiées par l'ajout.

L'algorithme se termine lorsque le critère d'erreur, de type erreur quadratique moyenne, passe par un minimum ou devient inférieur à un seuil donné.

Notons que la méthode ne contient aucune protection contre l'introduction du bruit contenu dans les données dans le modèle. En effet, le seul critère pour décider de l'ajout d'un nouvel ensemble flou est l'erreur, il ne prend pas en compte l'éventuelle proximité de sous-ensembles flous existants.

L'auteur propose une méthodologie semblable en considérant des sous-ensembles flous multidimensionnels. L'initialisation est réalisée en créant une seule règle correspondant au centre de gravité du nuage. Puis, le point pour lequel l'erreur est maximale devient le centre d'une nouvelle partition. On calcule les nouveaux coefficients d'appartenance, les nouvelles conclusions et on répète l'opération jusqu'à ce que l'erreur soit inférieure à un seuil donné.

Approches voisines : D'autres approches analogues existent (Klawonn & Keller, 1997; Lin et al., 1997). Nous décrivons ci-après l'indice de controverse utilisé par (Rojas et al., 2000).

Cet indice, défini pour une règle, donne une idée de la différence entre les sorties observées pour les exemples qui activent la règle et la conclusion de la règle. Il traduit également l'hétérogénéité des exemples qui activent la règle. Son expression pour la règle i est :

$$CI(i) = \sum_{k=1}^{n} \left([(y_k - R_i)\ w_i(k)]^2 \right)^{1/2}$$

où R_i est la conclusion de la règle i, y_k la sortie observée de l'exemple k et $w_i(k)$ le degré de vérité de la règle i pour l'exemple k.

Cette définition s'étend à un sous-ensemble flou, X_v^j. L'indice s'appelle alors la somme des controverses pour l'ensemble flou et s'écrit :

$$SCMF(X_v^j) = \sum_{r \in R} CI(r)$$

où R est l'ensemble des règles pour lesquelles la partie de l'antécédent impliquant la variable X_v est définie par le sous-ensemble flou j.

Pour pouvoir comparer les sous-ensembles flous de différentes variables, l'indice de controverse est normalisé par le produit du nombre des sous-ensembles flous de l'ensemble des autres variables :

$$\widehat{CI}(X_v^j) = \frac{CI(X_v^j)}{\prod_{\substack{m=1 \\ m \neq v}}^{p} n_m}$$

n_m est le nombre de sous-ensembles flous de la variables m.
La variance de la distribution des indices de controverses, pour la variable v est :

$$\sigma_v^2 = \frac{\sum_{j=1}^{n_v} \left(\widehat{CI}(X_v^j) - \overline{\widehat{CI}(X_v^j)} \right)^2}{n_v}$$

Soit : $M_{\sigma^2} = \max_v(\sigma_v^2)$

Les variables sur lesquelles les sous-ensembles seront rajoutés sont celles dont l'indice de variance est élevé, soit :

$$\sigma_v^2 > \rho M_{\sigma^2}, \qquad 0 < \rho < 1$$

La position du nouveau centre, dans la partition floue forte triangulaire, est :

$$c_v^* = \frac{\sum_{j=1}^{n_v} c_v^j \; CI(X_v^j)}{\sum_{j=1}^{n_v} CI(X_v^j)}$$

L'originalité de cette méthode, par rapport à la précédente, est que le critère d'affinement n'est plus calculé sur des régions de l'espace d'entrée mais au niveau des règles.

Sous forme d'arbre : (Glorennec, 1999) propose également un affinement de partition, sous forme d'arbre. La racine est constituée d'un SIF comprenant deux fonctions d'appartenance par entrée. Ensuite, on développe autant de sous-arbres que d'entrées. Chaque nœud définit un SIF qui est initialisé par un algorithme de prototypage rapide (la conclusion de chacune des règles est la valeur observée pour le point qui active le plus cette règle) puis optimisé (placement des fonctions d'appartenance et conséquences) et enfin caractérisé par l'erreur quadratique moyenne. L'heuristique consiste à développer, à chaque niveau, seulement la branche la plus performante. L'arrêt résulte d'un compromis entre la complexité, le nombre de règles, et la précision.

La différence principale avec les exemples précédents est d'ordre stratégique. Dans le cas présent, l'apprentissage est recommencé à chaque étape sur des structures de complexité croissante, alors qu'il est incrémental dans les cas précédents : les modifications structurelles n'impliquent pas de recommencer l'apprentissage, la mémoire est conservée.

Remarque : (Bersini et al., 1997) ont comparé les deux stratégies d'apprentissage dans un formalisme de réseau multi-expert, chaque expert agissant localement. La meilleure performance sur une série chronologique est obtenue par le modèle incrémental, même quand le modèle global est initialisé avec le nombre de règles trouvé par le modèle incrémental. L'erreur associée aux différentes structures de la stratégie globale présente un comportement erratique.
Ce résultat, obtenu dans un contexte situé, nous semble difficilement généralisable à l'ensemble des formalismes auxquels peuvent s'appliquer les deux stratégies.

A l'aide d'un algorithme génétique

Depuis qu'ils ont été proposés par Holland (Holland, 1975) et Goldberg (Goldberg, 1989), les algorithmes génétiques ont été largement utilisés pour apprendre des relations entre des entrées et des sorties.

(Ishibuchi et al., 1995) s'intéressent à des exemples à classer. Il s'agit donc de générer des règles floues qui réalisent une partition de l'espace en C zones disjointes, C étant le nombre de classes.

L'idée originale consiste à générer, de façon automatique, plusieurs partitions de l'espace avec différentes valeurs de K_i pour une variable i donnée. Les plus grossières correspondent aux petites valeurs de K_i. Toutes les règles, pour une partition donnée, sont implémentées. L'algorithme génétique est utilisé pour choisir le niveau de résolution, de granularité, le plus adapté à chacune des régions de l'espace. Pour cela, la fonction de coût vise à minimiser le nombre de règles tout en maximisant le nombre d'exemples bien classés.

A chaque système S est associée la performance :

$$f(S) = W_c C_S - W_r R_S$$

où C_S représente le nombre d'exemples correctement classés par le système S et R_S le nombre de règles présentes dans le système S. W_c et W_r sont des poids.

2.1.3 Les règles sont initialisées par les exemples

Pour (Wang & Mendel, 1992b) le nombre maximal de règles ne dépend plus, comme ci dessus, de la résolution du partitionnement. Après avoir partitionné l'espace de façon arbitraire, ils génèrent une règle par exemple : celle pour laquelle le degré de vérité pour l'exemple est le plus fort.

La procédure est décomposée en 5 étapes :

1. Un partitionnement en triangle est généré automatiquement pour chacune des variables d'entrée.
2. Une règle est générée pour chacun des exemples, elle s'écrit pour l'exemple i :

   $$SI\ x_1\ est\ A_1^i\ ET\ x_2\ est\ A_2^i\ \ldots ET\ x_p\ est\ A_p^i\ ALORS\ y\ est\ C^i$$

   Les sous-ensembles flous A_j^i sont ceux pour lesquels le degré d'appartenance des x_j^i sont les plus forts. De même, C_i est l'ensemble flou pour lequel le degré d'appartenance de y_i est maximum.
3. Un degré est assigné à chaque règle. Il est égal au degré de vérité de la règle pour l'exemple. Il peut être ensuite multiplié par un coefficient d'utilité, ou masse de croyance, fixé par l'utilisateur pour chaque exemple.

   C'est ici que sont gérés les conflits d'une façon un peu primaire : si deux règles ont les mêmes prémisses, une seule, celle dont le degré est le plus fort, est conservée. Cette gestion nous prive d'une information qui peut être précieuse, mais présente l'avantage de réduire notablement la base de règles.
4. Les règles de type ET précédemment décrites peuvent être combinées avec des règles de type OU, issues par exemple de l'expertise. Dans ce cas, le coefficient d'appartenance à tous les sous-ensembles flous des variables manquantes est égal à un, élément neutre pour le produit. La même gestion des conflits est appliquée ici.
5. La sortie est calculée par la défuzzification suivant le centre de gravité.

Remarquons que cette méthode ne dispense pas du choix de la résolution, elle repose aussi sur l'hypothèse implicite d'une bonne couverture de l'espace par la base d'apprentissage. En revanche, elle est évolutive : quand une nouvelle donnée est disponible elle entre en concurrence avec les règles existantes.

2.1.4 Les arbres de décision flous

Les arbres de décision flous sont une extension des arbres de décision binaires (Breiman et al., 1984; Quinlan, 1986). L'objectif est de définir des chemins conduisant à des classes homogènes. Chaque feuille de l'arbre correspond à une règle impliquant un nombre variable d'entrées. L'arbre est donc un sous-espace des règles possibles. Le partitionnement des entrées doit être fixé au départ. Le principal avantage de cette technique réside dans le fait que les règles générées sont incomplètes, comme la plupart des règles formulées par les experts, car elles n'impliquent pas toutes les variables disponibles. Leur interprétabilité dépend de celle du partitionnement.

(Ichihashi et al., 1996) proposent une implémentation flou de l'algorithme ID3 (pour Interative Dichotomizer) de Quinlan.

La construction de l'arbre se fait de façon itérative : à chaque étape un nouveau nœud est ajouté. Un nœud de l'arbre correspond à une variable d'entrée et donne naissance à autant de sous-arbres qu'il y a de sous-ensembles flous possibles, également appelés attributs, pour cette entrée. La procédure est répétée jusqu'à l'obtention de feuilles pures, contenant seulement des éléments qui appartiennent à la même classe.

Le choix de l'attribut s'effectue suivant le critère de maximisation de l'entropie à une étape donnée. L'arbre est alors vu comme un objet qui délivre un message, l'information contenue dans ce message étant la somme des entropies. La règle associée à un nœud donné, b, s'écrit :

$$SI\ x_{i_1}\ est\ A_{i_1}^{j_1}\ ET\ x_{i_2}\ est\ A_{i_2}^{j_2}\ \ldots ALORS\ y\ est\ C_b$$

$A_{i_1}^{j_1}$ est le premier nœud situé sur le chemin qui part de la racine et qui mène au nœud b, signifiant que la première variable sélectionnée est i_1 et que le sous-arbre qui conduit au nœud b est issu du sous-ensemble flou j_1 de cette variable ; C_b est la classe la plus représentée au sein du nœud b. Un exemple d'arbre est illustré sur la figure 2.4.

La prémisse de la règle correspondant au nœud b est définie par l'ensemble, Q, des couples (i, j), attribut j de la variable i, situés sur le chemin allant de la racine au nœud b.

L'entropie d'un nœud, b, est définie comme :

$$H_b = -\sum_k p_k^b * log(p_k^b)$$

p_k^b étant la densité de la classe k au sein du nœud b, c'est-à-dire la proportion d'éléments qui appartiennent à la classe k.

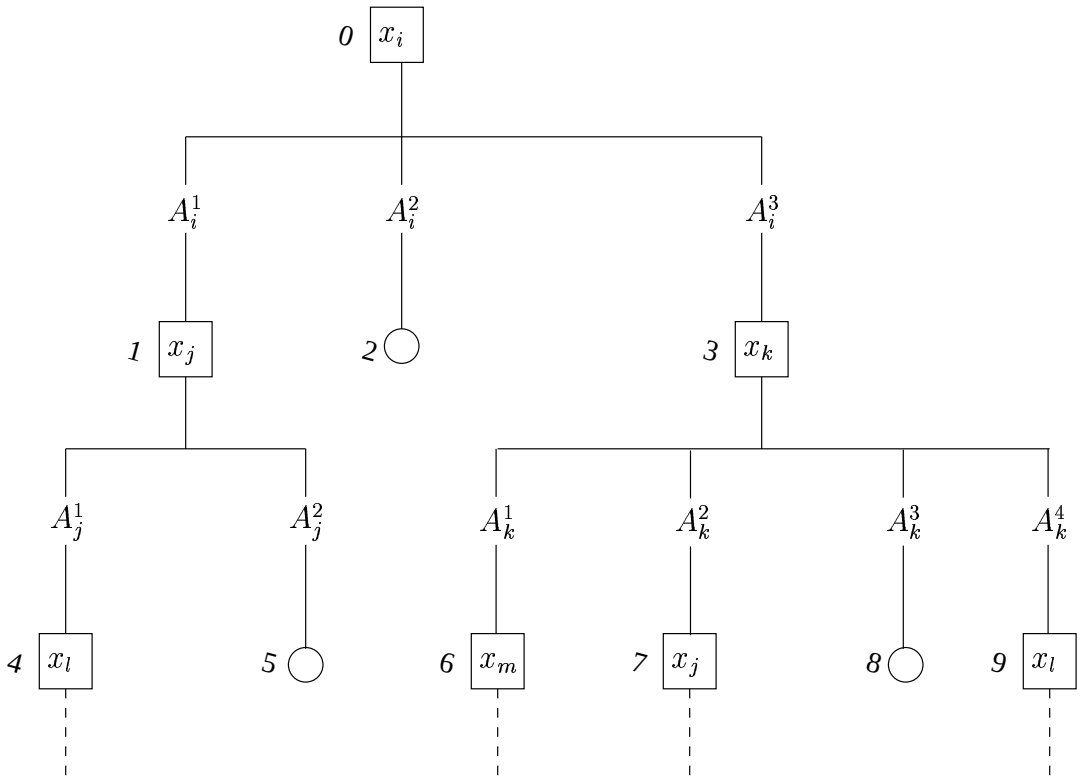
Les cardinalités sont floues et calculées comme la somme des degrés de vérité de la règle pour chacun des éléments.

$$p_k^b = \frac{|D_k^b|}{|D^b|} \qquad avec \qquad |D^b| = \sum_{x \in D^b} \left(\prod_{(i,j) \in Q} \mu_{i,j}(x) \right)$$

$|D_k^b|$ est définie de façon analogue mais à partir des seuls éléments de $x \in D^b$ qui appartiennent à la classe k.

Si H est l'entropie du nœud, et que l'attribut sélectionné à cette étape est décrit par V sous-ensembles flous, alors la nouvelle entropie est

$$E = \sum_{v=1}^{V} q^{b_v} * H_v \qquad o \qquad q^{b_v} = |D^{b_v}|/|D^b|$$



Règle noeud 2 : Si x_i est A_i^2 Alors y est C_2

Règle noeud 5 : Si x_i est A_i^1 et x_j est A_j^2 Alors y est C_5

FIG. 2.4 – Un exemple d'arbre de décision flou

Le gain d'entropie associé au choix de l'attribut est $G = H - E$.

La sélection d'attributs s'effectue jusqu'à l'obtention de secteurs homogènes.

Pour prendre en compte le caractère partiel de l'expertise, l'incertitude des experts, l'algorithme est adapté à la manipulation des fonctions de croyances de Dempster et Shafer(Shafer, 1976).

(Marsala, 1998), dans un système appelé Salammbô, propose différents choix pour les paramètres de construction des arbres de décision, incluant ceux présentés ci-dessus. Mais surtout, il implémente une méthode de partitionnement automatique des entrées basée sur la morphologie mathématique et la théorie des langages.

2.2 Groupage flou

Les algorithmes de groupage flou forment une famille bien identifiée des techniques d'induction de règles. L'objectif initial de ces algorithmes est d'organiser des données en groupes homogènes, pour, notamment dans le cadre de la classification, rendre compte de l'appartenance d'un individu à plusieurs classes. L'idée d'associer une règle à chaque groupe (ou *cluster*), et ainsi d'utiliser ces techniques pour l'induction de règles, est apparue dans un deuxième temps. Pour ce faire, le partitionnement des entrées correspondant à la prémisse de la règle est déduit des caractéristiques du groupe : prototype et zone d'influence pour chacune des entrées.

Contrairement à l'approche précédente, les sous-ensembles flous ne sont pas partagés par les règles, chacun d'entre eux correspond à une seule règle dans une dimension donnée. Leur interprétation, en termes linguistiques, est souvent difficile comme illustré sur la figure 2.5.

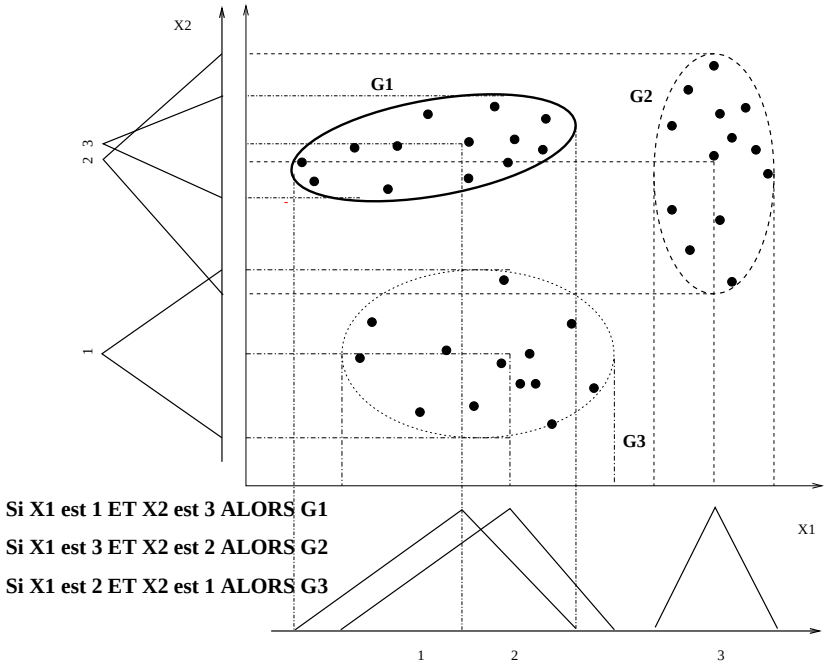


FIG. 2.5 – Un exemple de groupage

2.2.1 Les C-moyennes floues

La méthode de base, les C-moyennes floues (FCM en anglais), a été proposée par Dunn en 1973 (Dunn, 1973). Bezdek (Bezdek, 1981) en a démontré les propriétés de convergence et a proposé les premiers indices de mesure de la qualité des partitions obtenues. Chacun des n individus appartient à chacun des c groupes avec un coefficient d'appartenance u, u_{ik} étant le degré d'appartenance de l'exemple k au groupe i. Appelons D_{ki}^2 la distance entre l'exemple k et le groupe i. Elle peut être définie simplement comme la norme euclidienne, mais aussi de façon plus générale comme :

$$D_{ki}^2 = \|x_k - v_i\|_A^2 = (x_k - v_i)\ A\ (x_k - v_i)^T$$

A étant une matrice symétrique définie positive et v_i étant le prototype du groupe i.

Soient U la matrice des coefficients u_{ik} et V celle des coordonnées des centres. L'objectif de l'algorithme est de trouver U et V qui minimisent la fonction de perte suivante :

$$J_{FCM} = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m D_{ki}^2$$

sous la contrainte probabiliste :

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall k = 1, \ldots, n.$$

$m \geq 1$, est l'exposant flou.

Dans un article récent (Runkler & Bezdek, 1999), Bezbek rappelle que le groupage est défini par la fonction de perte à minimiser et pas par la méthode d'optimisation de cette fonction. La méthode d'optimisation la plus employée, dite optimisation par alternance, est présentée ci-après.

L'algorithme travaille par itérations. Au départ, les coefficients u_{ik} sont initialisés de façon aléatoire. Puis, chaque étape est composée des deux opérations suivantes :

1. Calculer les v_i en considérant les u_{ik} comme des constantes :

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m \, x_k}{\sum_{k=1}^{n} u_{ik}^m} \tag{2.3}$$

2. Calculer les u_{ik} en considérant les v_i fixés :

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{D_{ik}}{D_{jk}}\right)^{\frac{2}{m-1}}} \tag{2.4}$$

Ces opérations sont répétées jusqu'à ce que les coordonnées des centres soient stables, à une tolérance donnée près.

L'algorithme des FCM est bien adapté lorsque les groupes sont bien séparés ou de de taille et de forme comparables (sphérique dans le cas de l'utilisation de la matrice identité). Les prototypes sont dans ce cas les centres de gravité des groupes.

2.2.2 Variations autour de l'algorithme de base

Plusieurs améliorations ont été apportées à l'algorithme de base ((Gomez-Skarmeta et al., 1999) en font une revue dans une perspective semblable à la nôtre).

(Krishnapuram & Keller, 1993; Krishnapuram & Kim, 1999) ont proposé une version possibiliste en levant la contrainte probabiliste et en ajoutant un terme qui pénalise les degrés d'appartenance trop petits. La fonction de perte des PCM (Possibilistic C-Means) s'écrit :

$$J_{PCM} = \sum_{k=1}^{n} \sum_{i=1}^{c} \left(u_{ik}^m \|x_k - v_i\|^2 + \eta_i \left(1 - u_{ik}^m\right) \right)$$

$$avec \ \eta_i = \frac{\sum_{k=1}^{n} u_{ik}^m \|x_k - v_i\|^2}{\sum_{k=1}^{n} u_{ik}^m}$$

Dans l'algorithme de Gustafson-Kessel (Gustafson & Kessel, 1979) la matrice A dépend des données. Elle est fonction, pour chaque groupe, de l'inverse de la matrice de covariance.

La matrice de covariance du groupe i est calculée comme :

$$C_i = \frac{\sum_{k=1}^{n} u_{ik}^m \ \ (x_k - v_i)^T \ (x_k - v_i)}{\sum_{k=1}^{n} u_{ik}^m}$$

Et la distance du point k au groupe i devient :

$$D_{ki}^2 = (x_k - v_i) \left[det(C_i)^{1/n} \; C_i^{-1} \right] (x_k - v_i)^T$$

L'algorithme des FCRM (Fuzzy C-Regression Model) se différencie des FCM par le fait que les groupes produits ne sont pas de la forme d'une hypersphère, mais de celle d'un hyperplan (Hathaway & Bezdek, 1993; Kim et al., 1997; Kim et al., 1998). De plus, les prototypes, correspondant aux centres des groupes dans les FCM, ne sont pas des points mais des hyperplans. Soit pour le groupe i :

$$y_i = X^T P^i \quad avec \quad X = [1 \; x_1 \cdots x_m]^T \quad et \quad P^i = [a_0^i \; a_1^i \cdots a_m^i]^T$$

Les fonctions d'appartenance aux sous-ensembles flous des entrées sont des gaussiennes généralisées définies comme suit :

$$A(x) = e^{-\left(\frac{x-p_1}{p_2}\right)^2}$$

Les paramètres p_1 et p_2, ainsi que les coefficients a_j^i, sont ajustés par une descente de gradient.

2.2.3 Quelle partie des données utiliser pour le groupage ?

Le groupage flou peut se faire à partir des données complètes, entrées et sortie, ou bien des entrées ou des sorties seulement. Selon la partie des données utilisées, les règles induites seront complètement définies ou pas. Certains auteurs (Chiu, 1994; Babuska & Verbruggen, 1996) veulent tirer parti de la totalité de l'information disponible et travaillent dans l'espace produit $X \times Y$. Les règles sont alors complètement définies : la prémisse correspond aux entrées, la conclusion à la sortie. Cet avantage se double d'un inconvénient : certains éléments peuvent se trouver dans le même groupe alors qu'ils ne sont proches ni du point de vue des entrées, ni du point de vue des sorties, mais seulement parce que les distances correspondantes se compensent.

Sugeno et Emami (Sugeno & Yasukawa, 1993; Emami et al., 1998) travaillent dans l'espace des sorties seulement. La prémisse des règles correspondantes est obtenue en projetant les groupes sur l'espace des entrées. Cette opération est loin d'être triviale : le résultat est souvent bruité. D'autre part, la projection de certains groupes peut conduire à la définition de plusieurs sous-ensembles flous pour une entrée donnée. Cette situation peut correspondre à une réalité : il existe plusieurs prémisses conduisant à la même conclusion.

Si le groupage est effectué dans le seul espace des entrées il est nécessaire de prévoir une procédure de gestion des conflits : des exemples dont les conclusions sont différentes peuvent appartenir au même groupe car leurs descriptions sont voisines.

L'algorithme des FCM et ses dérivés utilisent deux paramètres : le nombre de groupes et l'exposant flou.

2.2.4 Nombre de groupes optimal

Comment trouver le nombre optimal de groupes ? Cette question a occupé nombre d'équipes de recherche. Deux types de méthodes permettent d'y répondre : soit exécuter un algorithme de type FCM avec un nombre de groupes croissant ($c = 2, \ldots, n-1$) et caractériser les partitions obtenues par des indices, soit utiliser un algorithme qui détermine le nombre de groupes.

Des indices de qualités pour les partitions obtenues par groupage flou

Xie et Beni (Xie & Beni, 1991) définissent la meilleure partition comme celle qui minimise le rapport de la compacité, $C(c)$, à la séparabilité, $S(c)$. Ces notions sont définies comme :

$$C(c) = \frac{1}{n} \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} \|x_k - v_i\|_A^2$$

$$S(c) = \min_{i \neq j} \|v_i - v_j\|_A^2$$

Sugeno (Sugeno & Yasukawa, 1993) suggère de retenir le nombre de groupes qui minimise le critère suivant :

$$V(c) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \ (\|x_k - v_i\|_A^2 \ - \|v_i - \overline{x}\|_A^2)$$

$\overline{x}$ est le centre de gravité des données. Le premier terme est la variance intra-groupe, le second correspond à la variance inter groupes.

(Emami et al., 1998) utilisent une formule voisine : le point de référence pour le calcul de la variance inter groupes n'est pas le centre de gravité, $\overline{x}$, mais son extension floue, $\overline{v}$. La différence sera d'autant plus significative que m sera grand.

(Burrough et al., 2000) utilisent un coefficient de partition, F, et un coefficient d'entropie, H, pour caractériser les différentes partitions. Ils sont définis comme :

$$F = \frac{1}{n} \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^2, \qquad \frac{1}{c} < F < 1$$

$$H = \frac{1}{n} \sum_{k=1}^{n} \sum_{i=1}^{c} -u_{ik} * ln(u_{ik}), \qquad 1 - F < H < ln(c)$$

Les valeurs de F et de H dépendent du nombre de groupes. Pour rendre ces coefficients indépendants du nombre de groupes, on effectue la normalisation suivante :

$$F_s = \frac{F - \frac{1}{c}}{1 - \frac{1}{c}} \qquad et \qquad H_s = \frac{H - (1 - F)}{ln(c) - (1 - F)}$$

Une *bonne* partition sera caractérisée par un coefficient de partition grand et un coefficient d'entropie petit. Le choix du compromis revient à l'utilisateur.

Groupage soustractif

(Chiu, 1994) a proposé un algorithme soustractif inspiré de la méthode de Yager (Yager & Filev, 1994). Chaque exemple est considéré comme un centre de groupe éventuel et est affecté d'un potentiel d'attraction qui est lui-même fonction d'un voisinage. Le voisinage est paramétré par un rayon, r_a. Pour l'exemple i, le potentiel d'attraction s'exprime par :

$$P_i = \sum_{j=1}^{n} e^{\frac{-4\|x_i - x_j\|^2}{r_a^2}} \qquad avec \qquad r_a > 0$$

L'exemple ayant le plus fort potentiel, P_1^*, noté x_1^*, devient le centre du premier groupe. Puis, le potentiel de chacun des exemples est diminué d'une valeur qui dépend de sa distance à x_1^*. La nouvelle valeur du potentiel pour l'exemple i est :

$$P_i = P_i - P_1^* e^{\frac{-4\|x_i - x_1^*\|^2}{r_b^2}} \qquad avec \qquad r_b \approx 1.5 r_a$$

Le processus est réitéré. Afin d'éviter l'introduction du bruit contenue dans les données dans le modèle, deux seuils sont définis, $s+$ et $s-$, typiquement 0.5 et 0.15, et un centre candidat, x_k^* à l'étape k, et son potentiel associé P_k^*, est géré de la façon suivante :

si $P_k^* > P_1^* * s+$, x_k^* est accepté comme centre d'un nouveau groupe ;
si $P_k^* < P_1^* * s-$, x_k^* est rejeté comme centre d'un nouveau groupe et l'algorithme s'arrête ;
sinon
    soit d_{min} la distance entre x_k^* et le centre du groupe le plus proche
        si $\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1$ x_k^* est accepté comme centre d'un nouveau groupe ;
            (il est suffisamment loin du groupe le plus proche)
        sinon x_k^* est rejeté, son potentiel est mis à 0 et l'algorithme continue.

Cet algorithme nous semble être sensible aux différents paramètres, rayons de voisinage et seuils. Malheureusement, le choix de ces valeurs demeure empirique.

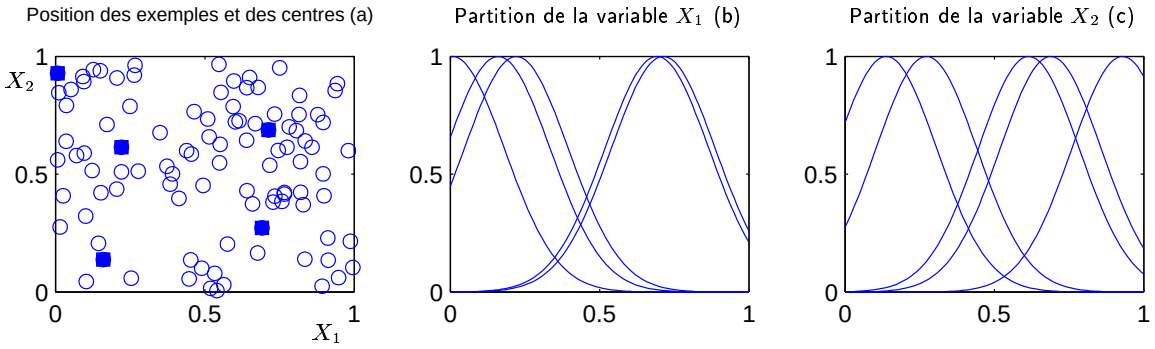


FIG. 2.6 – Un exemple de groupage soustractif

La figure 2.6 montre les résultats du groupage soustractif sur un échantillon aléatoire de 100 individus. Les 5 centres trouvés sont marqués sur la sous-figure (a). Les 5 règles qui leur correspondent sont définies par les sous-ensembles flous gaussiens dessinés sur les sous-figures (b) et (c). Ils sont obtenus par projection sur chacun des axes des points qui appartiennent à chacun des groupes.

2.2.5 Valeur de l'exposant flou

Cette valeur contrôle le niveau de flou du groupage. Plus elle est importante, plus la partition est floue. Lorsqu'elle tend vers l'infini les centres des groupes tendent tous vers le centre de gravité des données. La plupart des auteurs recommandent une valeur fixe de m, habituellement 1.5 ou 2.

(Chen & Wang, 1999) proposent une procédure d'optimisation de cette valeur. Ils travaillent avec l'algorithme des FCM, la fonction d'appartenance associée au groupe i est :

$$MF_i(x) = \frac{1}{1 + \left(\frac{x - v_i}{\sigma_i}\right)^{b_i}}$$

v_i est le centre de la fonction, σ_i est la largeur au point d'ordonnée 0.5, la pente au point $v + \sigma_i$ vaut $\frac{-b_i}{2\sigma_i}$.

La valeur de σ_i est égale à la racine carrée de la trace de la matrice de covariance du groupe i.

Les paramètres b_i sont calculés pour chaque groupe et pour chaque dimension comme le double du rapport de σ_i à une distance D. Cette distance est choisie comme la distance maximum entre le dernier centre et son précédent d'une part, le dernier centre et la valeur maximum de l'univers du discours d'autre part et les mêmes valeurs pour le premier centre et le minimum de l'univers du discours. Ce calcul assure un recouvrement suffisant pour permettre des transitions douces et garantir que l'univers est totalement couvert.

La procédure consiste à calculer les centres des groupes et les coefficients d'appartenance suivant l'algorithme des FCM. La valeur de m est initialisée à 1.5, et est augmentée, par pas de 0.1, jusqu'à ce que dans chaque dimension de l'entrée, il y ait au moins un groupe dont la valeur σ_i soit supérieure ou égale à l'écart type des données d'apprentissage dans cette dimension. Ainsi, pour la phase d'initialisation, les zones de l'espace sont suffisamment couvertes et on peut penser qu'il n'y a pas de région pour laquelle aucune règle ne s'applique.

La procédure est assez coûteuse : pour chaque incrément de m il faut appliquer les FCM et calculer les matrices de covariance. Nous serions tentés d'adopter une alternative plus économique qui consiste à vérifier la sensibilité du modèle à la valeur de m.

La méthode proposée par (Li & Mukaidono, 1999) n'utilise pas d'exposant flou. La fonction de perte à minimiser est :

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik} D_{ki}^2 \qquad avec \qquad \sum_{i=1}^{c} u_{ik} = 1 \quad \forall k = 1, \ldots, n$$

Cette méthode maximise l'entropie pour chaque exemple k sous deux contraintes : (a) minimiser la fonction de perte pour l'exemple considéré, et (b) normalisation de la somme des degrés d'appartenance. L'expression du problème est donc :

$$\begin{aligned} &maximiser \quad \left\{ -K \sum_{i=1}^{c} u_{ik}\ log(u_{ik}) \right\} \qquad K > 0 \\ &sous \qquad (a) \quad \sum_{i=1}^{c} u_{ik} D_{ki}^2 = \kappa \quad \kappa > 0\ ,\ \kappa \approx 0 \\ &et \qquad (b) \quad \sum_{i=1}^{c} u_{ik} = 1 \end{aligned}$$

L'algorithme est similaire à celui des FCM, les centres des groupes sont calculés suivant l'équation 2.3, en revanche, pour le calcul des degrés d'appartenance, l'équation 2.4 est remplacée par la solution du problème d'optimisation 2.5, soit l'équation 2.5.

$$u_{ik} = \frac{e^{-\frac{D_{ki}^2}{2\sigma^2}}}{\sum_{j=1}^{c} e^{-\frac{D_{kj}^2}{2\sigma^2}}} \tag{2.5}$$

La valeur $2\sigma^2$ est appelée la température, le paramètre σ détermine κ, la valeur de la fonction de perte.

2.2.6 Les centres mobiles flous

(Warwick, 1995) propose un algorithme de groupage dont la convergence est assurée et qui est moins coûteux que celui des FCM : Mean-tracking Algorithm.

L'algorithme est analogue à celui des centres mobiles (Saporta, 1990). Sur chaque axe de l'espace d'entrée, j ($j = 1, \ldots, p$), l'utilisateur prédéfinit une fenêtre de centre c_j : $[c_j - a_j; c_j + a_j]$. a_j est choisi en fonction de l'écart type de la distribution sur l'axe j.

La procédure est itérative : on calcule le centre de gravité de la fenêtre de centre c_j, puis dans un deuxième mouvement la fenêtre est centrée sur son centre de gravité. On peut alors calculer le nouveau centre de gravité, et ainsi de suite jusqu'à stabilisation.

Chaque fenêtre correspond à un groupe. On peut initialiser de manière aléatoire autant de fenêtres que l'on veut, et éventuellement fusionner les groupes associés.

La version floue consiste à associer un sous-ensemble flou, de forme radiale, à chaque centre.

Si la convergence est assurée, elle est très dépendante de la valeur initiale du centre. Pour éviter les solutions locales, les statisticiens appliquent l'algorithme des centres mobiles plusieurs fois à un même jeu de données : ils dégagent ainsi des formes fortes.

Enfin, il n'est pas prévu de faire évoluer la taille de la fenêtre en fonction des données. De fait, les groupes couvriront tous le même espace.

2.3 Méthodes hybrides

Cet ensemble de méthodes intègre toutes sortes d'outils dont les plus utilisés sont les réseaux de neurones et les algorithmes génétiques. Les réseaux de neurones ont apporté aux systèmes d'inférence floue leurs algorithmes d'apprentissage et leur précision dans l'ajustement numérique sans trop tenir compte de la sémantique. Les algorithmes génétiques sont susceptibles de trouver un optimum global et permettent d'optimiser, dans le même mouvement, la structure et les paramètres des systèmes d'inférence floue. Ce type d'outils peut être appréciable lorsque la sémantique est secondaire par rapport aux performances numériques, et pour la gestion des problèmes pour lesquels aucune connaissance experte n'est disponible.

2.3.1 Les modèles neuro-flous

Les modèles neuro-flous (Glorennec, 1991), comme l'ANFIS, "Adaptive Neuro Fuzzy Inference Systems" (Jang, 1993; Jang, Sun & Mizutani, 1997), sont des systèmes d'inférence floue implémentés comme des réseaux de neurones. Chacune des couches du réseau correspond à une partie du système d'inférence floue : fuzzification pour les entrées, calcul du degré de vérité des règles et inférence, défuzzification pour la sortie. L'avantage principal de ce type de représentation réside dans le fait que les paramètres du système d'inférence floue sont codés sous la forme de poids synaptiques et peuvent donc être optimisés par les méthodes d'apprentissage bien connues des réseaux de neurones : règles de Hebb, rétro-propagation, ...

Nous nous intéressons à un type particulier de ces réseaux, appelé RBF, pour "Radial Basis Function networks" : chacun des neurones de la couche cachée correspond à un modèle local. Cette architecture a été introduite par Moody et Darken (Moody & Darken, 1989), et depuis, un travail important a permis de relier les réseaux de neurones et les systèmes d'inférence floue. Jang a montré que les RBF sont strictement équivalent à des systèmes d'inférence flous, avec quelques conditions

restrictives (Jang & Sun, 1993). La plus importante étant que la conclusion des règles est un scalaire. Plus récemment, Cho et Wang (Cho & Wang, 1996) ont proposé des améliorations pour travailler avec des conclusions floues ou polynomiales.

Un RBF comporte 3 couches : une couche d'entrée de la dimension du vecteur d'entrée, une couche de sortie de la dimension du vecteur de sortie et une couche cachée dont la dimension maximale est égale au nombre d'exemples.

Chaque cellule de la couche cachée est une unité de réception (ou d'attraction) à structure radiale (symétrie par rapport au centre) qui sera ajustée localement. Les champs d'attraction se recouvrent partiellement. L'analogie avec le groupage est évidente : une unité de réception correspond à un prototype. Le but de l'apprentissage est de configurer le réseau pour que chacune de ces cellules reconnaisse un et un seul type d'exemples.

A chaque neurone caché est connecté l'ensemble des neurones d'entrée, le neurone réalise alors l'opération $R_i(x) = e^{-\frac{(x-c_i)^2}{\sigma_i^2}}$. Où c_i et σ_i sont les centres et écart type des gaussiennes[d] dans chacune des dimensions de l'espace d'entrée.

Chaque neurone de sortie est connecté à l'ensemble des neurones cachés. Dans la configuration la plus simple, les conclusions des règles sont des scalaires, la défuzzification est facile : la sortie est la somme pondérée des connexions, un poids de connexion représentant le degré de vérité de la règle correspondante.

Quand les conclusions des règles sont de forme polynomiales, $y_i = b_o^i + b_1^i x_1 + b_2^i x_2 + \cdots + b_p^i x_p$, les poids entre la couche d'entrée et la couche cachée ne sont pas constants. Ils correspondent aux coefficients $b_1^i, \ldots, b_p^i$. Le poids b_o^i est matérialisé par un biais.

L'apprentissage consiste à déterminer le nombre minimum de neurones dans la couche cachée, soit le nombre de règles, les paramètres correspondants, c_i et σ_i, ainsi que les poids.

Le nombre de neurones de la couche cachée est initialisé à zéro. Ensuite, pour chacune des itérations, l'ensemble des exemples est présenté. Pour chaque exemple i, à chaque étape k, où $r_k(j)$ est la rayon d'attraction du neurone j, et ϵ une tolérance fixée :

si $|y_i - \widehat{y}_{ik}| > \epsilon$ alors
  s'il existe un neurone j tel que $\frac{(x_i - c_j)^2}{\sigma_j^2} < r_k(j)$ alors
    modifier c_j et σ_j par une descente de gradient.
  sinon créer un nouveau neurone centré sur x_i[e].
sinon ajuster les paramètres des neurones de la couches cachée par une descente de gradient.

Lorsque tous les exemples ont été présentés, à la fin de chacun des cycles, le rayon seuil d'attraction est diminué.

L'algorithme s'arrête lorsque la somme des carrés des erreurs est inférieure à un seuil ou bien lorsque le nombre d'itérations maximal a été atteint.

Notons que dans cet algorithme, l'ordre de présentation des exemples peut influer de manière sensible sur le résultat. Remarquons aussi que cette technique s'apparente à celle du groupage soustractif présentée dans la section 2.2.4.

[d] N'importe quelle fonction radiale peut être utilisée, la gaussienne n'est qu'une illustration.

[e] Les valeurs initiale de σ sont une fonction de l'écart type des données.

D'autres auteurs essaient de travailler avec des réseaux de neurones dans un souci d'interprétabilité. (Mitra et al., 1997; Mitra & Hayashi, 2000) s'intéressent à un problème de classification. Chacune des entrées est découpée à l'aide de trois sous-ensembles flous, et chacun d'eux est lui même divisé en trois par des modificateurs linguistiques. La couche de sortie est formée de deux neurones par classe ou sous-classe convexe. Le premier apprend à reconnaître les exemples positifs pour la classe, ceux qui appartiennent à la classe, le second les exemples négatifs. L'interprétabilité consiste à générer une règle pour chacun des nœuds de sortie. Cette opération est réalisée par un examen des poids du réseau : la règle correspond au chemin entre l'entrée et la sortie pour lequel les poids sont les plus forts.

Pour d'autres (Roy, 2000), ces tentatives sont vouées à l'échec : le formalisme des réseaux de neurones n'autorise pas l'extraction de règles.

2.3.2 Les algorithmes génétiques

Les algorithmes génétiques, introduits dans la section 2.1.2, ont été mobilisés pour la conception de contrôleurs flous (Karr, 1991; Hoffmann & Pfister, 1996; Magdalena, 1997; Chiang et al., 1997; Leitch & Probert, 1998; Gonzalez & Perez, 1999; Wong & Her, 1999)[f]. Un ouvrage récent synthétise l'ensemble de leurs applications pour l'apprentissage de bases de règles floues (Cordon et al., 2001).

Le modèle proposé par (Russo, 1998) est présenté comme une illustration. Il vise à combiner les avantages de la logique floue, interprétabilité des règles, des réseaux de neurones, algorithmes d'apprentissage, et des algorithmes génétiques, aptitude à trouver un optimum global.

Son algorithme d'évolution considère une population de réseaux de neurones. L'apprentissage consiste à optimiser les poids des connexions, des neurones peuvent être enlevés. Une fois l'apprentissage terminé, l'individu est codé comme un chromosome pour participer à l'évolution grâce à des opérations de sélection, croisement, mutation.

Le réseau, qui correspond à un système d'inférence floue de r règles, est formé de 4 couches :

1. La couche d'entrée : elle comprend au plus p neurones, p étant la dimension de l'espace d'entrée.
2. La couche de fuzzification : le nombre de neurones maximum est rp.
   Il y a un sous-ensemble flou par entrée et par règle. La fonction d'appartenance est gaussienne, paramétrée par c et g, g étant l'inverse de l'écart type[g]. Les valeurs c et g sont codées comme des poids de connexion, et optimisées par rétropropagation. Un choix important est matérialisé dans cette couche : les ensembles flous sont propres à chacune des règles. La complexité s'en trouve notablement réduite, mais le partitionnement induit est moins interprétable.
3. La couche d'inférence : le degré de vérité de chacune des règles est calculé par un opérateur *min*. Les poids de connexion sont tous fixés à 1.
4. La couche de sortie : les conclusions des règles sont des scalaires. La défuzzification peut être soit de type "Weighted mean", la formule est donnée dans l'équation 2.1 ; soit de type "Weighted sum", la même que la précédente sans le dénominateur. Dans ce dernier cas, la sortie est relative. L'auteur indique qu'une sortie de type "Weighted sum" est suffisante pour des problèmes d'interpolation, tandis que l'autre est préférable pour des problèmes de classification.

[f] Bien que les algorithmes génétiques soient très populaires, d'autres techniques stochastiques peuvent être utilisées comme le recuit simulé (Guély et al., 1999).

[g] Cette astuce permet d'optimiser les temps de calcul en multipliant au lieu de diviser, et d'éviter les singularités au voisinage de $\sigma = 0$.

La fonction de coût est bien construite ! Elle intègre bien évidemment les erreurs entre sortie désirée et sortie inférée, mais de plus elle favorise les règles incomplètes et les systèmes les plus simples, ceux dont le nombre d'entrées est minimal.

2.4 Classification des méthodes

Nous proposons une classification des principales méthodes d'induction de règles dans les trois familles décrites dans cette partie. Le tableau 2.2 récapitule les principales conclusions se rapportant à chacune d'elles.

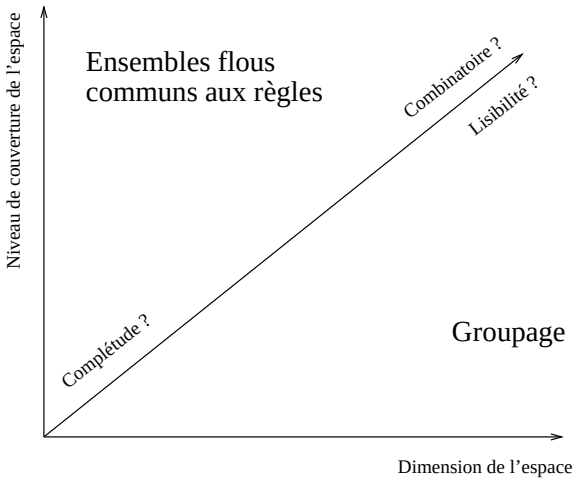


FIG. 2.7 – Choix des méthodes d'induction en fonction des données

Les conditions d'utilisation préférentielles de deux grandes familles de méthodes sont présentées dans la figure 2.7. La bissectrice des axes partage l'espace d'utilisation de celles-ci en fonction des caractéristiques de la base de données : la dimension de l'espace de travail et le niveau de couverture de cet espace par la base d'apprentissage.

Les méthodes pour lesquelles les ensembles flous sont communs à l'ensemble des règles, s'utilisent de préférence pour un petit nombre de variables et lorsque la base d'exemples couvre bien l'espace de travail. Sinon, si le niveau de couverture est trop faible, la complétude n'est plus garantie et, si la dimension devient importante, le nombre de combinaisons à prendre en compte devient trop élevé.

A l'inverse, le groupage s'utilise pour un grand nombre de variables et un faible nombre d'exemples. Toutefois, la lisibilité des règles induites se dégrade lorsque la dimension augmente.

Du fait de leur hétérogénéité, il est difficile de positionner, de manière globale, les méthodes hybrides sur ce graphique. Leurs performances dépendent de leur implémentation, et particulièrement de la façon de coder le problème. C'est la raison pour laquelle il est difficile de les évaluer globalement. L'usage de ces méthodes est motivé par leurs performances numériques et par le fait qu'elles peuvent, dans le même mouvement, optimiser à la fois la structure et les paramètres du système. Toutefois, si l'optimisation est guidée par le seul indice de performance numérique, le partitionnement induit peut s'avérer illisible rendant toute tentative d'interprétation difficile.

Quelle que soit la technique utilisée pour générer des règles, les différentes parties du système peuvent être optimisées pour accroître l'interprétabilité.

Famille	Méthode	Inconvénient	Réponses
Partition partagée	Toutes les règles	Explosion combinatoire	Affinement de partition
	Une règle par exemple	Complétude non garantie	
	Arbres de décision	Partitionnement requis *a priori*	
Groupage	FCM	Nombre de groupes ?	Groupage soustractif
Méthodes hybrides	Neuro-flou (incluant RBF) Algos génétiques	Dépend de l'implémentation, du codage.	

TAB. 2.2 – Principales méthodes d'induction de règles

B - Optimiser le système

Les différents niveaux d'optimisation possibles sont habituellement classés en deux catégories : optimisation paramétrique et optimisation structurelle.

L'optimisation paramétrique a trait principalement à la valeur de la conclusion des règles et à l'ajustement de la position des sous-ensembles flous. Les méthodes disponibles sont éprouvées, leurs avantages et inconvénients respectifs sont bien connus (Jang et al., 1997; Glorennec, 1999).

Nous nous intéressons dans les deux sections qui suivent à l'optimisation structurelle : sélection des variables d'entrée et réduction de la base de règles. Comme les bases de données disponibles deviennent de plus en plus importantes en volume, la conception automatique des systèmes d'inférence floue doit intégrer la sélection de variables et la réduction de la base de règles pour satisfaire les besoins croissants d'extraction de connaissance à partir des données. Définir le système avec les seules variables nécessaires contribue à améliorer à la fois l'interprétabilité et la stabilité. En effet, la diminution du nombre de variables entraîne la diminution du nombre de règles, et chacune des règles gagne en lisibilité. De plus l'optimisation paramétrique devient plus facile sans les variables secondaires qui sont susceptibles d'apporter davantage de bruit que d'information. Quelques méthodes pour l'optimisation paramétrique sont présentées dans la section 2.7.

2.5 Sélection des variables

La sélection de variables peut être réalisée au niveau global ou bien au niveau local. Dans le premier cas, la variable est enlevée et aucune des règles ne peut l'utiliser, tandis que dans la seconde configuration, la sélection est réalisée au niveau des règles et conduit donc à des règles incomplètes.

Certaines méthodes déjà présentées dans la section précédente effectuent une sélection de variables. Par définition, les arbres de décision ne s'appuient que sur les variables nécessaires pour constituer des feuilles homogènes. La fonction de coût de l'algorithme génétique utilisée par (Russo, 1998) tend à minimiser le nombre de variables. Les variables d'entrée sont parfois sélectionnées en fonction de leur effet sur la sortie, les moins influentes peuvent être retirées du modèle. Ceci peut s'appliquer dans divers formalismes comme les réseaux de neurones (Ruck et al., 1990; Pal, 1999), ou des méthodes relationnelles (Huang & Chu, 1999).

2.5.1 Indice de régularité

(Sugeno & Yasukawa, 1993) proposent d'effectuer la sélection, en minimisant un critère de régularité, par une procédure de validation.

Les données, n exemples décrits par p variables, sont partagées aléatoirement en deux groupes, A et B. Le critère de régularité, RC, est défini comme :

$$RC = \frac{1}{2}\left[\frac{1}{k_A}\sum_{i=1}^{k_A}\left(y_i^A - y_i^{AB}\right)^2 + \frac{1}{k_B}\sum_{i=1}^{k_B}\left(y_i^B - y_i^{BA}\right)^2\right]$$

y_i^A (respectivement y_i^B) représente la sortie désirée pour l'exemple i du groupe A (respectivement B), et y_i^{AB} (respectivement y_i^{BA}) représente la sortie, pour l'exemple i du groupe A (respectivement B) inférée par le système construit avec les données du groupe B (respectivement A).

Les variables sont introduites selon une procédure ascendante. La première étape consiste à bâtir p modèles de une variable, et à retenir la variable qui minimise le critère RC. A l'étape suivante, il faudra évaluer $p-1$ modèles de 2 variables constitués de la variable sélectionnée et de chacune des variables candidates restantes. La procédure s'arrête lorsque la valeur du critère RC augmente à nouveau.

Remarquons que le nombre de modèles à évaluer est borné : $\frac{p\ (p+1)}{2}$. Ce nombre est à comparer au nombre total de modèles possibles : $2^p - 1$.

2.5.2 Critères géométriques

Pour l'induction de règles (Emami et al., 1998) effectuent un groupage sur l'espace des sorties suivie d'une projection sur l'espace des entrées. Les sous-ensembles flous résultent du lissage de cette projection. Pour une règle donnée, un indice de significativité est calculé pour chacune des variables d'entrée : c'est le rapport de la largeur du noyau à celle du support. En effet, 1 est l'élément neutre de l'opérateur ET. Cependant, les auteurs utilisent une combinaison globale de ces indices locaux, leur produit, pour effectuer la sélection sur l'ensemble de la base de règles.

Une autre méthode statique a été proposée dans (Lin & Cunningham, 1994). Elle est d'une complexité linéaire en le nombre de variables et le nombre d'exemples. Autour de chaque point k et dans chaque dimension j, on définit un sous-ensemble flou par la fonction d'appartenance suivante :

$$\mu_{jk}(x) = e^{-\left(\frac{x_{jk}-x}{b}\right)^2} \qquad avec \qquad b \approx \frac{x_j^{max} - x_j^{min}}{10}$$

Une règle floue est associée à chacun des exemples. Pour chaque variable j, la sortie de chacun des exemples d'apprentissage l est calculée comme :

$$c_{jl} = \frac{\sum_{k=1}^{n} \mu_{jk}(x_{jl}).y_k}{\sum_{k=1}^{n} \mu_{jk}(x_{jl})}$$

L'ensemble des valeurs c_{jl} forme la courbe floue de l'entrée j. Les auteurs utilisent la dynamique de la courbe $(c_j^{max} - c_j^{min})$ comme un indice d'influence de la variable j.
Le processus de construction d'une courbe floue est illustré sur la figure 2.8.

Nous ne sommes pas convaincus par le bien fondé de cette méthode. Lorsqu'elle est appliquée à un cas d'école dans lequel sont introduites des variables fictives, les dynamiques correspondantes aux variables actives sont comparables entre elles ainsi que celles associées aux variables fictives. La sélection est possible. Sur des données réelles, seuls les résultats de prédiction sont montrés, les dynamiques ne sont pas données.

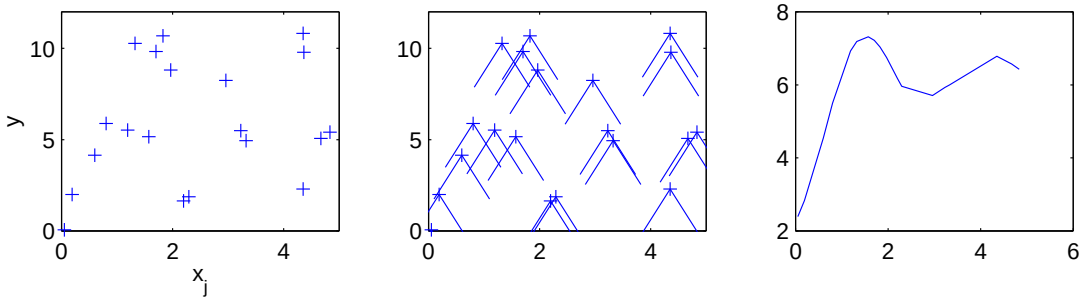


FIG. 2.8 – Construction de la courbe floue

D'autre part, la dénomination de courbe floue cache un estimateur à noyau multidimensionnel projeté sur une dimension. Ces estimateurs ont été largement étudiés par les mathématiciens, et aucun de leurs résultats n'indique que les projections sur un axe ne sont comparables ou significatives.

Enfin, considérer les variables de manière isolée, hors de tout contexte, revient à faire l'hypothèse, très forte, de leur indépendance. Cette hypothèse n'est que rarement vérifiée dans les problèmes réels.

2.5.3 Pouvoir discriminant individuel

L'originalité de la méthode proposée par (Hong & Chen, 1999) réside dans le fait d'effectuer la sélection des variables avant de définir le partitionnement. Notons cependant qu'elle ne s'applique qu'à des problèmes de classification. La procédure se déroule en 5 étapes :

1. Notons q_i le nombre de valeurs uniques de l'entrée i : A_{ij}, $j = (1, \ldots, q_i)$
2. Soit n_{ij} le nombre d'instances dont la valeur pour l'entrée i est A_{ij} ; soit n_{ijk} le nombre de ces instances qui appartiennent à la classe k, $n_{ijk} \leq n_{ij}$. Le pouvoir de discrimination est basé sur le nombre d'instances pour lesquelles une valeur unique correspond à une et une seule classe. Ce nombre, t_i, est calculé comme :

$$t_i = \sum_j n_{ijk} \quad \{j = 1, \ldots, q_i | \exists k,\ n_{ijk} = n_{ij}\}$$

3. Calculons un indice de pouvoir discriminant pour chacune des variables d'entrée à l'aide d'un des deux formules suivantes :

$$f_i = \frac{t_i}{n}$$

$$f_i = 1 - \left\{ -\frac{1}{q_i} \sum_{j=1}^{q_i} \sum_{k=1}^{c} \left[\left(\frac{n_{ijk}}{\sum_{k=1}^{c} n_{ijk}} \right) log_p \left(\frac{n_{ijk}}{\sum_{k=1}^{c} n_{ijk}} \right) \right] \right\}$$

   La seconde correspond à une définition classique de l'entropie, c étant le nombre de classes.
4. Les variables sont classées par ordre de pouvoir discriminant décroissant.
5. Les variables sont introduites dans le système, suivant l'ordre défini ci-dessus, tant que l'erreur est inférieure à un seuil donné, 0.1 par exemple. La variable err est initialisée à 1 et actualisée comme $err = err * (1 - f_i)$ quand l'entrée i est sélectionnée.

Les auteurs ne soulignent pas que cette approche fait implicitement l'hypothèse de l'indépendance des variables en considérant leurs contributions individuelles comme additives.

2.5.4 Indice de variation d'entropie

Cette approche, proposée par (Pal, 1999) est similaire à la précédente : elle adresse les problèmes de classification et fait l'hypothèse implicite de l'indépendance des variables.

L'entropie est une mesure du niveau de flou. Pour un ensemble A donné, elle peut s'exprimer par :

$$H(A) = \frac{1}{nlog2} \sum_{i=1}^{n} -\mu_A(x_i)log(\mu_A(x_i)) - (1-\mu_A(x_i))log(1-\mu_A(x_i))$$

$H(A)$ est maximum lorsque A est le plus flou, c'est-à-dire lorsque $\mu_A(x_i) = 0.5\ \forall i$, et est minimum si $\mu_A(x_i) = 0\ or\ 1\ \forall i$.

L'auteur utilise une sigmoïde pour modéliser la fonction μ qui est définie de la manière suivante sur l'intervalle $[a, c]$, b étant le point de croisement tel que la fonction en ce point vaut 0.5, $b = \frac{a+c}{2}$:

$$\mu_A(x_i; a, b, c) = \begin{cases} 0, & x_i \leq a \\ 2\left[\frac{x_i - a}{c-a}\right]^2, & a \leq x_i \leq b \\ 1 - 2\left[\frac{x_i - c}{c-a}\right]^2, & b \leq x_i \leq c \\ 1, & x_i \geq c \end{cases}$$

Soient x_{qj} les valeurs de l'attribut q pour l'ensemble des individus qui appartiennent à la classe j. Considérons l'ensemble flou caractérisé par la fonction d'appartenance μ définie par les paramètres suivants :

$$\begin{cases} b = (x_{qj})_{av} \\ c = b + max(|(x_{qj})_{av} - (x_{qj})_{max}|\ ,\ |(x_{qj})_{av} - (x_{qj})_{min}|) \\ a = 2b - c \end{cases}$$

Dans ces expressions, av, min et max désignent respectivement la moyenne, le minimum et le maximum des valeurs de x_{qj}.

Les plus grandes valeurs de la fonction H pour cet ensemble, H_{qj}, se rencontrent lorsque un grand nombre d'exemples ont des degrés d'appartenance proches de 0.5, c'est-à-dire lorsqu'ils sont regroupés autour de leur valeur moyenne. Autrement dit, la valeur de H_{qj} sera inversement proportionnelle à la variance de la variable q pour la classe j.

Si l'on fusionne deux classes, j et k, et que l'on recalcule les paramètres a, b, c en tenant compte de l'effectif des deux classes, alors l'entropie du nouvel ensemble H_{qjk} sera d'autant plus petite que les moyennes des 2 classes sont distantes. H_{qjk} est donc inversement proportionnel à la variance de la variable q entre les classes j et k.

La variable q la plus discriminante pour les classes j et k est celle qui minimise l'indice d'évaluation de variable suivant :

$$IEV_q = \frac{H_{qjk}}{H_{qj} + H_{qk}}$$

Méthode	**Technique**	**H**[a]	**Avantage ou *inconvénient***
Indice de régularité	Par validation croisée		Méthode incrémentale- *Coûteux*
Algo génétique	Sélection globale ou règle par règle		*Peu informatif*
Critères géométriques			
Largeur noyau des sous-ensembles flous	Mesure locale à une règle	•	*Combinaison globale d'indices locaux*
Courbe floue	Estimateur à noyau	•	*Interprétation du critère ?*
Restriction à la classification :			
Arbres de décision			Règles incomplètes
Variation d'entropie	Par paire de classes	•	*Sélection sur l'ensemble des règles*
Pouvoir discriminant individuel	Ensemble des classes	•	Sélection avant partitionnement

TAB. 2.3 – Sélectionner les variables

Pour travailler avec plus de deux classes, l'auteur propose la généralisation suivante :

$$IEVG_q = \frac{\sum\limits_{\substack{j,k=1 \\ j \neq k}}^{N_c} H_{qjk}}{\sum\limits_{j=1}^{N_c} H_{qj}}$$

Remarquons que cet indice travaille en moyenne. Un descripteur qui permet de séparer nettement une classe d'un ensemble d'autres n'aura pas forcément un indice fort. Sa valeur est indépendante de la population de chacune des classes, ce qui peut être considéré comme un avantage ou un inconvénient suivant le contexte.

Les caractéristiques des principales méthodes présentées sont rappelées dans le tableau 2.3.

2.6 Optimisation de la base de règles

Rappelons qu'une base de règles doit posséder les trois propriétés définies dans l'introduction du présent chapitre : continuité, cohérence et complétude. Si, de plus, on s'intéresse à l'interprétabilité des règles, il est nécessaire d'éliminer la redondance.

Nous avons déjà vu que les algorithmes génétiques, (Russo, 1998), peuvent prendre en compte le nombre de règles dans les critères d'optimisation. De même, les RBF, (Cho & Wang, 1996), ou les techniques de groupage (Chiu, 1994), en créant de nouveaux centres au fur et à mesure des besoins, tendent à minimiser le nombre de règles du système.

La démarche inverse est également possible : générer un grand nombre de règles, au maximum une par exemple d'apprentissage, puis réduire cet ensemble. Les méthodes de réduction de la base de règles sont aussi utiles si l'on souhaite fusionner deux ou plusieurs bases de règles, par exemple l'une issue de l'expertise et l'autre issue des données.

Deux grandes familles de techniques sont disponibles. La première consiste à fusionner des éléments de même type : groupes, sous-ensembles flous ou variables. La seconde famille est basée sur des techniques statistiques.

2.6.1 Fusion d'éléments compatibles

Générer un grand nombre de règles par une méthode de groupage permet d'être moins dépendant de l'initialisation des centres des groupes. L'équipe de l'université de Delft (Kaymak & Babuska, 1995; Babuska & Verbruggen, 1995; Babuska & Verbruggen, 1996; Babuska et al., 1998) propose une amélioration de la méthode de fusion de groupes compatibles initialement développée par Krishnapuram et Freg (Krishnapuram & Freg, 1992). La forme des groupes est déterminée par les vecteurs propres (directions de l'ellipsoïde) et les valeurs propres associées (longueur des axes). Le groupage est effectué avec l'algorithme de Gustafson/Kessel, la distance d'un point à un groupe est fonction de la matrice de covariance. Pour le groupe i, l'hyperplan est défini par l'équation $(x - v_i).\phi_{is} = 0$, dans laquelle ϕ_{is} représente la plus petite valeur propre du groupe i.

Les deux critères de fusion, pour deux groupes i et j, sont :

1. Les hyperplans sont presque parallèles : $|\phi_{is}.\phi_{js}| \geq k_1$, k_1 voisin de 1.
2. Les centres sont proches : $\|c_i - c_j\| \leq k_2$, k_2 positif et voisin de 0.

Deux matrices symétriques $C1$ et $C2$ sont calculées. Chaque élément $c1_{ij}$ (respectivement $c2_{ij}$) indique le degré de similarité entre les groupes i et j pour le premier (respectivement second) critère.

Ces valeurs sont fuzzifiées dans un espace de dimension 2, les coordonnées du candidat idéal sont maintenant $(1, 1)$. Les fonctions d'appartenance sont de type exponentiel. Les matrices $\tilde{C}1$ et $\tilde{C}2$ contiennent les degrés d'appartenance de chacun des éléments. Pour prendre en compte le fait que les deux critères peuvent partiellement se compenser, deux groupes pas tout a fait parallèles mais très proches peuvent être fusionnés, et vice-versa, le degré de comptabilité global entre les groupes i et j est défini comme la moyenne géométrique des deux critères :

$$\mu_{ij} = \sqrt{\tilde{c}1_{ij} \ . \ \tilde{c}2_{ij}}$$

La matrice de ces degrés de compatibilité peut être seuillée (à 0.7 dans l'exemple). Avant de fusionner les groupes candidats, il faut vérifier que leur voisinage ne contient pas de groupe incompatible au sens des critères de fusion. Soit M l'ensemble des groupes candidats à la fusion, les groupes i et j seront effectivement fusionnés si :

$$min \max_{\substack{c_i \in M \\ c_k \notin M}} d_{ik} > \max_{c_i, c_j \in M} d_{ij}$$

d_{ij} est la distance entre les groupes i et j dans l'espace des prémisses, le groupage étant effectué dans l'espace produit, entrée plus sortie.

La fusion de groupes compatibles est strictement équivalente à la fusion de règles car à chaque groupe est associée une règle.

Dans une autre méthode, également proposée par l'équipe de Delft (Setnes et al., 1998), les éléments fusionnés sont des ensembles flous. Ces fusions ont pour conséquence une réduction de la base de règles. Selon les auteurs, les effets indésirables de l'induction automatique peuvent être caractérisés par les trois similarités suivantes :

- similarité entre deux sous-ensembles flous d'une même entrée ;
- similarité entre un sous-ensemble flou et l'ensemble universel $(\mu(x) = 1, \quad \forall x)$;
- similarité entre un sous-ensemble flou et un singleton.

Ils proposent une méthode de gestion des deux premiers cas et restent prudents sur le troisième. Bien que les règles définies par cet ensemble ont peu de chance d'être activées, ils préfèrent consulter un expert avant de les enlever. En effet, elles peuvent traduire des exceptions.

La similarité entre deux ensembles flous, A et B, peut être calculée, par exemple, comme :

$$S_{(A,B)} = \frac{|A \cap B|}{|A \cup B|}$$

$|.|$ représente la cardinalité floue, $\cap$ et $\cup$ sont les opérateurs d'intersection et d'union.

L'algorithme consiste donc à fusionner les deux sous-ensembles flous les plus similaires, puis à mettre à jour la base de règles pour tenir compte de la nouvelle situation. L'opération est répétée tant que des fusions sont possibles. A la fin, les sous-ensembles flous universels sont enlevés.

Deux seuils sont nécessaires : un seuil de similarité entre sous-ensembles flous et un seuil différent, supérieur au précédent, de similarité avec l'universel. Bien évidemment, les résultats sont très dépendants de ces valeurs. La baisse des seuils augmente la lisibilité du modèle.

Si les sous-ensembles flous sont modélisés par des trapèzes, a_1, a_2, a_3, a_4 étant les paramètres du sous-ensemble A, le nouvel ensemble C est défini à partir de $A \cup B$ comme :
- $c_1 = min(a_1, b_1)$
- $c_2 = \lambda_2 a_2 + (1 - \lambda_2) b_2$
- $c_3 = \lambda_3 a_3 + (1 - \lambda_3) b_3$
- $c_4 = max(a_4, b_4)$

λ_2, $\lambda_3 \in [0, 1]$, et valent 0.5 dans l'exemple.

La fusion mathématique consiste à former une variable adjointe par combinaison linéaire des variables initiales. Le danger de cette opération est la perte de signification physique. (Lacrose, 1997), dans une problématique de contrôle, combine l'erreur et ses dérivées successives.

L'utilisation de fonctions d'appartenance multidimensionnelles permet également de diminuer le nombre de règles. Le découpage de l'espace d'entrée est assuré par un maillage de Delaunay, soit une triangulation pour un espace de dimension 2. (Benoit & Foulloy, 1993; Foulloy et al., 1994) ont utilisé cette technique pour la conception de capteurs symboliques de couleur. Concevoir des ensembles flous multidimensionnels qui ont réellement un sens pour le problème considéré n'est pas une tâche triviale.

2.6.2 A partir de méthodes statistiques

Ces méthodes initialisent également un grand nombre de règles, au plus une par exemple, et sélectionnent les plus influentes par des méthodes issues de la statistique. Ces méthodes sont éprouvées, et fondées d'un point de vue mathématique. Notons toutefois, que celles qui travaillent sur un domaine d'entrée transformé conduisent à une perte de sémantique.

Orthogonalisation des moindres carrés

Cette famille de méthodes, (Chen et al., 1989; Chen et al., 1991), appelée Orthogonal Least Square (OLS), effectue la sélection par régression linéaire[h]. Pour utiliser des techniques linéaires dans un cadre d'optimisation non linéaire, il est nécessaire de reformuler le problème. Un système d'inférence floue peut être décomposé en deux niveaux : tout d'abord les entrées sont projetées, de manière non linéaire, dans un nouvel espace; puis la sortie est calculée comme une combinaison linéaire des composantes de ce nouvel espace.

Pour (Wang & Mendel, 1992a), un système d'inférence floue peut être représenté comme une combinaison linéaire de fonctions floues de base, FBF en anglais. Chacune de ces FBF assurant

[h]Le développement complet de la méthode est donné en annexe E.

la projection non linéaire des entrées. Une règle par exemple est générée comme indiqué dans la section 2.1.3. La fonction d'appartenance correspondant à l'entrée j de la règle i est une gaussienne centrée sur x_j^i :

$$\mu_{A_j^i}(x_j) = a_j^i \; e^{-\frac{1}{2}\left(\frac{x_j - x_j^i}{\sigma_j^i}\right)^2} \quad avec \quad 0 < a_j^i \leq 1$$

La sortie inférée pour l'entrée x est :

$$f(x) = \frac{\sum_{i=1}^{r} c^i \left(\prod_{j=1}^{p} \mu_{A_j^i}(x_j) \right)}{\sum_{i=1}^{r} \left(\prod_{j=1}^{p} \mu_{A_j^i}(x_j) \right)}$$

La valeur de la FBF, $f_i(x)$, est la contribution relative de la règle i pour la sortie calculée pour l'exemple x :

$$f_i(x) = \frac{\prod_{j=1}^{p} \mu_{A_j^i}(x_j)}{\sum_{i=1}^{r} \left(\prod_{j=1}^{p} \mu_{A_j^i}(x_j) \right)}$$

Le système peut être maintenant réécrit sous la forme : $\widehat{y} = \sum_i f_i(x)\ \theta_i$, où $\theta_i \in R$ représentent les paramètres, scalaires, à optimiser. Soit encore $y = F\theta + E$; y étant le vecteur des sorties observées dans la base d'apprentissage et E le vecteur d'erreur.

Chaque régresseur, f_i, est un vecteur de la dimension, r, du nombre de règles, le terme général f_{ji} étant le degré de vérité de la règle i pour l'exemple j.

L'algorithme d'apprentissage de l'OLS transforme les f_i vecteurs originaux en un ensemble de vecteurs orthogonaux par la méthode de Gram-Schmidt. La matrice F est décomposée en une matrice orthogonale, W, et une matrice triangulaire supérieure, A. L'espace engendré par l'ensemble des vecteurs orthogonaux étant identique à celui engendré par les vecteurs originaux, f_i, le problème s'écrit : $y = Wg + E$. La solution des moindres carrés est $\widehat{g_i} = \dfrac{w_i^T y}{w_i^T w_i}$, $1 \leq i \leq r$. Les paramètres θ s'obtiennent facilement par la relation : $A\widehat{\theta} = \widehat{g}$. Les vecteurs w_i étant orthogonaux, leurs contributions individuelles sont additives, il n'y a aucune covariance. A chaque étape l'algorithme sélectionne le vecteur w_i qui maximise la variance expliquée sur la sortie y, c'est-à-dire le critère suivant :

$$[err]_i = \frac{g_i^2 \; w_i^T w_i}{y^T y}$$

L'algorithme s'arrête lorsque la sortie est suffisamment reconstruite. Ce qui se produit à l'étape r_f telle que : $1 - \sum_{i=1}^{r_f} [err]_i < \epsilon$, ϵ étant un seuil donné.

Une fois les règles sélectionnées, Hohensohn et Mendel (Hohensohn & Mendel, 1994) proposent d'effectuer une deuxième passe de l'algorithme dans le seul objectif d'optimiser les conclusions des règles, sans faire aucune sélection. Cette opération a pour but de reconstruire des vecteurs w_i à partir des seules règles sélectionnées, et ainsi, d'éviter qu'ils contiennent de l'information provenant de règles éliminées par la procédure.

Méthodes d'analyse des données multidimensionnelles

L'analyse des données multidimensionnelles propose des outils qui réalisent une réduction de l'espace de travail. Le plus connu est sans aucun doute l'analyse en composantes principales, ACP. L'ensemble de ces méthodes est basé sur une propriété des matrices rectangulaires, appelée décomposition en valeurs singulières, SVD en anglais.

Cette décomposition s'écrit :

$$X = \sum_{i=1}^{l} \sqrt{\lambda_i}\ u_i\ v_i' \qquad ou\ bien \qquad X = U\Sigma V^T$$

l, $l \leq min(n,p)$, est le rang de la matrice, c'est-à-dire nombre de valeurs propres non nulles de la matrice XX' ou de la matrice $X'X$. Les valeurs propres sont notées λ_i et sont classées par ordre décroissant. Les vecteurs u_i, de dimension n, sont les vecteurs propres associés aux valeurs propres du même indice dans l'espace des lignes de la matrice X. Les vecteurs v_i, dimension p, sont les vecteurs propres associés dans l'espace des colonnes. Les vecteurs u_i et v_i sont normés.

Des travaux récents mobilisent cette technique pour la conception de systèmes d'inférence floue (Yen et al., 1998; Yam et al., 1999). Le modèle de Yen (Yen et al., 1998) est de la forme $Y = Xb$. La matrice X est initialisée à partir des exemples, comme dans la section précédente. Il y a donc n règles initialisées à partir de chacun des exemples. La conséquence de la règle construite à partir de l'exemple i est : $y_i(x) = b_i^0 + b_i^1\ x_1 + \cdots + b_i^p\ x_p$. La ligne j de la matrice comprend n blocs, un par règle, de $p+1$ valeurs correspondant aux coordonnées du point j pondérées par le degré de vérité de chacune des règles pour l'exemple j. Soit pour le bloc correspondant à la règle i :

$$w_i(j) \quad w_i(j)\ x_1(j) \quad w_i(j)\ x_2(j) \quad \cdots \quad w_i(j)\ x_p(j)$$

L'examen des valeurs singulières permet de déterminer la dimension de l'espace final r, tel que $r \leq rang(X)$. Il faut ensuite partitionner V pour obtenir V_{11} de dimension $r * r$ de la façon suivante : $V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$. On note $\overline{V}^T = [V_{11}^T \ \ V_{21}^T]$, et on effectue une décomposition QR[i] de $\overline{V}$, Q est une matrice orthogonale de dimension $r \times r$ et R_{11} une matrice triangulaire supérieure de la même dimension, pour obtenir la matrice de permutation Π : $Q^T\overline{V}^T\Pi = [R_{11} \ \ R_{12}]$. Les r plus importantes partitions floues sont dans les r premières colonnes de Π.

L'ACP peut également être utilisée pour réduire l'espace de travail. (Kim et al., 1998) effectuent une opération de groupage pour initialiser les règles. Ils appliquent l'ACP, au niveau de chaque groupe, pour décorréler les variables.

Pour chaque groupe, donc pour chaque règle, on calcule tout d'abord la moyenne m_i, puis la matrice de covariance C^i. Enfin, les valeurs propres et les vecteurs propres de la matrice de covariance définissent un nouvel espace pour chaque règle dans lequel les p variables d'entrée sont des combinaisons des p variables initiales.

Nous voulons souligner que, contrairement aux techniques de fusion qui préservent la sémantique, les méthodes qui travaillent dans un espace transformé ne sont pas conseillées pour la coopération homme machine car les règles produites sont difficilement interprétables. Les caractéristiques des principales méthodes d'optimisation de la base de règles présentées sont résumées dans le tableau 2.4.

[i] Cette méthode est similaire à celle de Gram-Schmidt.

Famille	**Principales approches**	**Technique**	**Avantage ou *inconvénient***
Procédures incrémentales	Groupage RBF Affinement de partition	Ajout de règles	*Choix du nombre de règles* *Risque d'introduction du bruit dans le modèle.*
Fusion	de groupes	Heuristique	Moins dépendant des conditions initiales
	de sous-ensembles flous	Mesures de similarité	Arithmétique floue (cadre formel)
	mathématique	Recombinaison de variables	*Attention à la sémantique*
	symbolique	Sous-ensembles flous multidimensionnels	*Conception difficile*
Méthodes statistiques	OLS SVD ACP	Régression après une projection non linéaire Réduction de la dimension de l'espace de travail	Mathématiquement robuste *Lisibilité ?* *A déconseiller*

TAB. 2.4 – Optimiser la base de règles

2.7 Optimisation paramétrique

Sans avoir la prétention d'être exhaustif, nous allons présenter dans cette section les travaux récents relatifs à l'optimisation de la forme des sous-ensembles flous et rappeller les principales méthodes pour l'optimisation des conséquences des règles.

2.7.1 Optimiser la forme des sous-ensembles flous

Cette phase d'optimisation peut être considérée comme secondaire pour nombre d'approches, car elle est de plus en plus intégrée dans la démarche même de conception. Elle est en revanche indispensable, lorsque la conception s'est résumée à une initialisation grossière. La forme des sous-ensembles flous dépend à la fois de l'allure de la fonction d'appartenance, elle peut être triangulaire ou gaussienne par exemple, et des paramètres nécéssaires pour la définir. Ces paramètres, qui induisent les bornes qui délimitent les ensembles, contribueront, plus que l'allure, à l'efficacité et à la lisibilité du modèle. Le nombre de sous-ensembles flous sur chacune des entrées influera directement sur le nombre de règles.

Les algorithmes génétiques sont capables d'optimiser, dans le même mouvement, non seulement les conséquences des règles, les paramètres, mais encore le nombre et la forme des sous-ensembles flous, c'est-à-dire la structure (Leitch & Probert, 1998). Il suffit pour cela que le chromosome contienne le codage correspondant aux paramètres nécessaires. Un sous-ensemble flou de forme triangulaire est codé par trois valeurs (il en faut quatre pour un trapèze, mais seulement deux pour une gaussienne).

(Wong & Her, 1999) présentent un codage qui permet d'atteindre les deux objectifs, forme et nombre des sous-ensembles flous, dans un même mouvement. Chaque entrée (sortie), i, est divisée en un nombre maximal de sous-ensembles flous, $2n_i + 1$ ($2n_o + 1$). n_i (n_o) représente le nombre de sous-ensembles flous de part et d'autre du sous-ensemble flou central. Chaque sous-ensemble flou est de forme triangulaire. La position des sommets est repérée par la distance au sommet précédent. Le sommet central est fixe et les distances sont notées $d(i,1)$, $d(i,2)$, dans une direction et $d(i,-1)$, $d(i,-2)$ dans l'autre. Les pieds des triangles correspondent aux sommets des voisins (partition floue forte).

Quand les valeurs $d(i,j)$ sont déterminées, on retient $p_i + q_i + 1$ sous-ensembles flous par entrée et sortie. p_i et q_i satisfont les conditions suivantes (u_i et l_i représentent les limites hautes et basses du domaine de variation de l'entrée i) :

$$d_{i,1} + d_{i,2} + \cdots + d_{i,q_i-1} < u_i \leq d_{i,1} + d_{i,2} + \cdots + d_{i,q_i}$$

$$d_{i,-1} + d_{i,-2} + \cdots + d_{i,-p_i-1} < |l_i| \leq d_{i,-1} + d_{i,-2} + \cdots + d_{i,-p_i}$$

Le système est formé de l'ensemble des règles correspondant à toutes les combinaisons possibles.

La fonction à optimiser tient compte des performances du système à évaluer et de sa complexité, c'est-à-dire du nombre de règles.

$$f(R_i) \;=\; g_1(J(R_i)) \;*\, g_2(m(R_i))$$

$J(R_i)$ mesure la performance du système R_i, $m(R_i)$ est le nombre de règles, soit le produit sur l'ensemble des entrées des quantités $q_i + p_i + 1$. Les fonctions g_k sont des fonctions d'appartenance gaussiennes.

Les méthodes qui génèrent des règles à partir d'une opération de groupage sur les sorties ont besoin d'une étape de lissage. La projection brute des coordonnées des points du groupe sur les entrées est inutilisable, car fortement bruitée. (Emami et al., 1998) travaillent avec des triangles, l'algorithme suivant, proposé par (Sugeno & Yasukawa, 1993), calcule les quatre paramètres du trapèze le mieux ajusté :

1. Pour une entrée donnée, le pas d'ajustement f est fixé à une valeur voisine de 5% de l'univers du discours.
2. Notons p_j^k le k^{ime} paramètre du j^{ime} sous-ensemble flou de l'entrée.
3. Calculer $p_j^k + f$ et $p_j^k - f$
   si k = 2, 3, 4 et si $p_j^k - f < p_j^{k-1}$ alors $\widehat{p}_j^k = p_j^{k-1}$ sinon $\widehat{p}_j^k = p_j^k - f$
   si k = 1, 2, 3 et si $p_j^k + f < p_j^{k+1}$ alors $\widehat{\widehat{p}}_j^k = p_j^{k+1}$ sinon $\widehat{\widehat{p}}_j^k = p_j^k + f$
4. Calculer l'indice de performance (somme des carrés des écarts entre sorties observées et sorties inférées) pour $p_j^k, \widehat{p}_j^k, \widehat{\widehat{p}}_j^k$ et remplacer p_j^k par le meilleur.
5. Retourner à l'étape c) tant que ça évolue.
6. Répéter l'ensemble de la procédure un nombre fixé de fois, 20 par exemple.

Pour (Hong & Chen, 1999), la conception des fonctions d'appartenance, modélisées par des triangles, se réalise à travers les étapes suivantes :

- Le nombre initial de sous-ensembles flous pour chacun des attributs dépend du nombre d'exemples : $G = 1 + 3.3\ log\ n$, où log désigne le logarithme népérien.
- Le pas de mouvement, pour l'ajustement des limites des triangles, dépend de la dynamique de l'entrée i :
$$H_i = \frac{x_i^{max} - x_i^{min}}{G - 1}$$
- La valeur minimale de l'univers du discours est prise en dehors de cette plage : $V_i = x_i^{min} - \frac{H_i}{2}$
- L'univers est divisé en G groupes. La limite basse du groupe j correspond à la limite haute du groupe $j-1$, la borne basse du premier est V_i. Les limites hautes des G groupes sont : $V_i + j\ H_i, \quad j = 1, \ldots, G$
- A chacun de ces groupes, est associée une fonction d'appartenance triangulaire, paramétrée par a, b et c. b étant le sommet, a et c les limites basse et haute. Le sommet du groupe j de l'attribut i vaut :
$$b_{ij} = \sum_{s=1}^{r_{ij}} \frac{A'_{ij}(I_{ijs})}{r_{ij}}$$
  r_{ij} est le nombre d'exemples dont la valeur est dans le groupe j pour l'attribut i, $A'_{ij}(I_{ijs})$ est la valeur de l'entrée i pour l'exemple I_{ijs}.
- Enfin, on choisit $a_{ij} = b_{i(j-1)}$ et $c_{ij} = b_{i(j+1)}$

L'optimisation finale se fait par une opération de fusion. Les auteurs travaillent en classification : à chaque exemple est associée une classe. L'espace d'entrée est partagé en cellules. Les cellules doivent être homogènes : elles indiquent une, et une seule, classe de sortie ou bien elles sont vides.

En cas de conflit, la classe assignée à une cellule est celle pour laquelle le degré d'appartenance est le plus fort. Celui-ci est calculé comme la somme des degrés d'appartenance à la cellule des exemples étiquetés de la classe.

La procédure de fusion permet une redéfinition des sous-ensembles flous : il s'agit de construire, pour chaque région les ensembles les plus grands rassemblant des cellules homogènes. Il est possible de fusionner des lignes, des colonnes ou des cellules adjacentes homogènes, ou bien séparées par des

zones de cellules vides. Le résultat, conduit, sur le problème des iris, à un ensemble de 10 règles, et une performance en classification, mesurée par validation croisée, de 96%.

(Chow et al., 1999; Altug et al., 1999) introduisent des contraintes de forme pour l'optimisation de systèmes d'inférence floue dans le but de combiner la précision et la lisibilité.

Les contraintes formulées sont relativement "basiques". Elles indiquent que le noyau doit être non vide, que les fonctions doivent être convexes (au sens de la définition de Zadeh) :

$$x_a \leq x_b \leq x_c \Longrightarrow min\{\mu_i^j(x_a), \mu_i^j(x_b)\} \leq \mu_i^j(x_c) \qquad \forall x_a, x_b, x_c, j.$$

Tous les points de l'univers du discours doivent avoir une appartenance non nulle pour au moins un sous-ensemble flou, les sous-ensembles flous doivent se recouvrir partiellement. Enfin, pour le sous-ensemble flou *petit*, le plus à gauche sur l'axe, on souhaite :

$$x_a \geq x_b \Longrightarrow \mu_{petit}(x_a) \leq \mu_{petit}(x_b)$$

2.7.2 Optimiser la conséquence des règles

L'optimisation de la conséquences des règles est une phase essentielle de la conception des systèmes d'inférence floue. Elle détermine la précision du modèle. Les techniques, dont les avantages et les limites sont connues, sont éprouvées et largement diffusées.

(Ishibuchi et al., 1994) proposent une descente de gradient stochastique pour optimiser les les conclusions, des constantes, des règles générées par l'heuristique présentée dans la section 2.1.1.

Soit E le signal d'erreur défini pour un point i comme $E_i = \frac{(\widehat{y}_i \ - \ y_i)^2}{2}$. $\widehat{y}_i$ représente la sortie inférée et y_i la sortie observée.

Pour chacune des règles, la nouvelle conséquence est calculée comme l'ancienne diminuée de la dérivée partielle de l'erreur observée pour ce point multipliée par le degré de vérité de la règle pour cet exemple ($\mu_r(i)$) et multipliée par le pas du gradient, β :

$$b_r^{new} = b_r^{old} + \beta(-\partial E_i/\partial b)$$
$$b_r^{new} = b_r^{old} - \beta \ . \ \mu_r(i) \ . \ (\widehat{y}_i \ - \ y_i)$$

La descente de gradient peut être appliquée pour optimiser les conclusions des règles des modèles de Sugeno, des polynômes d'ordre 1, ou bien pour l'optimisation de fonctions d'appartenance sous la réserve que celles-ci soient dérivables.

Les algorithmes génétiques, plus lourds à mettre en œuvre et plus coûteux, présentent l'avantage de fournir un optimum global. (Chiang et al., 1997) ont utilisé cette méthode pour la conception de contrôleurs capables d'apprendre en ligne. Ils sont souvent utilisés pour une optimisation à la fois structurelle et paramétrique.

(Bortolet, 1998) propose une méthode pour obtenir des conséquences qui soient des fonctions linéaires des entrées à partir de conséquences constantes. Cette méthode ne change ni les prémisses des règles, ni la valeur des sorties pour les valeurs modales. Les coefficients de la combinaison sont définis à partir des conséquences des règles voisines.

Les règles initiales sont de la forme : *si* x_1 *est* A_1^i *et* x_2 *est* A_2^j *alors* y *est* B_{ij}. Les sous-ensembles flous A_k^l sont des triangles. La règle, après linéarisation, s'écrit :

si x_1 *est* A_1^i *et* x_2 *est* A_2^j *alors* $y = a_0 + a_1 * x_1 + a_2 * x_2$.

Les valeurs des coefficients sont :

$$\begin{aligned} a_1 &= \frac{\left(\frac{B_{(i+1)j}-B_{ij}}{A_1^{i+1}-A_1^i} + \frac{B_{ij}-B_{(i-1)j}}{A_1^i-A_1^{i-1}}\right)}{2} \\ a_2 &= \frac{\left(\frac{B_{i(j+1)}-B_{ij}}{A_2^{j+1}-A_2^j} + \frac{B_{ij}-B_{i(j-1)}}{A_2^j-A_2^{j-1}}\right)}{2} \\ a_0 &= B_{ij} - a_1 * A_1^i - a_2 * A_2^j \end{aligned}$$

2.8 Conclusion

Les méthodes de conception de systèmes d'inférence floue à partir des données sont souvent caractérisées par un indice de performance tel que l'erreur quadratique moyenne. L'amélioration aveugle de ce critère risque de se produire au détriment de l'originalité de la logique floue : son interprétabilité.

Le tableau 2.5 compare les différentes familles de méthodes d'induction de règles et d'optimisation de la base de règles en terme d'interprétabilité. Les méthodes qui génèrent ou utilisent un partitionnement partagé produisent des règles interprétables, du fait de la lisibilité du partitionnement. Toutefois nombre d'entre elles ne sont pas utilisables pour des systèmes importants, comme le montre la figure 2.7. L'explosion combinatoire interdit l'emploi des méthodes qui génèrent toutes les règles ; tandis que celles qui ne génèrent qu'une règle par exemple conduisent à un niveau de couverture trop faible, ou bien, si le nombre de points d'apprentissage est important, à un nombre de règles très grand. La seule méthode de cette famille utilisable dans des grands espaces est l'arbre de décision flou.

Les techniques de groupage sont efficaces particulièrement dans les grandes dimensions et lorsqu'il y peu d'exemples. Néanmoins, comme les sous-ensembles flous induits sont propres à chacune des règles, l'interprétabilité du système est fortement limitée par le fait que les règles ne sont pas comparables.

La dernière famille des méthodes d'induction est caractérisée par un niveau d'interprétabilité variable, dû à l'hétérogénéité de cette famille. Notons qu'historiquement l'objectif de ces méthodes n'était pas la production d'un système interprétable. Toutefois, des travaux récents tentent d'améliorer cet aspect.

La seconde condition à satisfaire pour rendre le système interprétable est la réduction de la base de règles. La sélection de variables en constitue la première étape. Ensuite, d'autres méthodes d'optimisation, dont le niveau d'interprétabilité est indiqué dans le tableau 2.5, peuvent être appliquées.

Les méthodes statistiques conduisent à un niveau d'interprétabilité très faible si le partionnement est effectué dans un domaine d'entrée transformé. Les techniques de fusion sont préférables, même si les résultats dépendent des types d'éléments qui sont fusionnés : ils sont meilleurs lorsqu'il s'agit d'ensembles flous que lorsqu'il s'agit de groupes.

Enfin, la seule méthode qui génère des règles incomplètes, et qui est donc susceptible de prendre en compte notre troisième critère, est l'arbre de décision flou. Les autres méthodes de sélection de variables sont globales : les variables écartées ne sont plus disponibles pour aucune des règles.

Famille	Interprétabilité
Génération automatique	
Partitionnement partagé	*Bon* : Les ensembles flous peuvent s'interpréter comme des labels linguistiques
Arbre de décision flou	*Bon* : Les règles générées sont incomplètes
Groupage	*Faible* : Par construction les ensembles flous sont propres à chacune des règles, ce qui rend la comparaison des règles et l'interprétabilité difficiles
Méthodes hybrides	*Faible à moyen* : Faible si les fonctions d'appartenance sont optimisées, moyen lorsque la génération est effectuée avec un partitionnement fixé
Optimisation de la base de règles	
Fusion	*Moyen à bon* : Varie en fonction du type d'éléments fusionnés
Méthodes statistiques	*Bon à faible* : Bon pour l'OLS, faible si transformation du domaine d'entrée

TAB. 2.5 – Interprétabilité des méthodes d'induction de règles floues

Les arbres de décision flous apparaissent, au terme de cette étude, comme la meilleure méthode utilisable. Le partitionnement des entrées peut être réalisé simplement à l'aide de triangles dont les sommets sont calculés par un algorithme de groupage tel que les centres mobiles ou $k - means$ (Saporta, 1990).

Si le niveau de couverture de l'espace de travail est suffisant, une sélection des règles par régression linéaire (OLS) peut s'avérer intéressante. La sélection s'effectue sur les centres des sous-ensembles flous, déterminés dans chaque dimension par les valeurs des exemples d'apprentissage. Il est alors possible d'imposer simplement aux sous-ensembles flous un recouvrement tel qu'ils puissent s'interpréter comme des labels linguistiques.

La méthode que nous proposons s'efforce de satisfaire ces critères d'interprétabilité. Elle est présentée dans le deux chapitres suivants. Le premier traite de la génération d'une famille de partitions floues au sein de chaque dimension ; le second, utilise ces partitions pour construire un système multidimensionnel.

Chapitre 3

Partitionnement ascendant hiérarchique (PAH)

La simplification est nécessaire, mais elle doit être relativisée. J'accepte la réduction consciente qu'elle est réduction, et non la réduction arrogante qui croit posséder la vérité simple, derrière l'apparente multiplicité et complexité des choses.

Edgar Morin

(Morin, 1990)

Sommaire

La première condition d'interprétabilité des règles est la lisibilité du partitionnement : les sous-ensembles flous doivent pouvoir s'interpréter comme des labels linguistiques. L'étude bibliographique du chapitre précédent a montré d'une part, que cette condition n'était absolument pas garantie par l'utilisation du formalisme flou ; et d'autre part, que pour pouvoir comparer les règles, il est nécessaire que celles-ci utilisent les mêmes sous-ensembles flous.

La méthode d'induction de règles que nous proposons opère en deux étapes. La première est présentée dans ce chapitre : elle a pour but de définir un partitionnement des entrées qui satisfasse la condition de lisibilité. Elle est conduite indépendamment au sein de chacune des dimensions. La deuxième étape, présentée dans le chapitre suivant, utilise ces partitions pour induire des règles floues par composition multidimensionnelle.

Revenons à l'objectif de ce chapitre, qui est de construire une famille de partitions floues pour chacune des entrées. Chacune de ces partitions sera caractérisée par un indice de qualité. L'utilisateur pourra alors choisir le nombre de sous-ensembles flous le plus adapté.

La démarche que nous proposons, appelée partitionnement ascendant hiérarchique, s'inspire de deux types de méthodes : l'une issue de la classification ascendante hiérarchique utilisée en statistique, l'autre dérivée du groupage (clustering) adapté au formalisme flou, largement utilisée en reconnaissance de forme.

La classification ascendante hiérarchique est un processus qui permet de regrouper des individus de façon itérative suivant un critère. Le point de départ est une partition en classes dans laquelle chaque classe n'est formée que d'un seul individu, la variance intra-classe est nulle et la variance totale est égale à la seule variance inter-classes. Le point d'arrivée est une partition formée d'une seule classe correspondant à l'ensemble des points du nuage. La variance totale correspond alors à la seule variance intra-classe.
A chaque étape de l'algorithme, deux classes sont regroupées. Le critère d'agrégation le plus communément employé est le critère de Ward : les deux classes qui sont regroupées sont celles qui minimisent l'augmentation de la variance intra-classe. En effet, lorsque l'on passe d'une partition de $k+1$ classes à une partition à k classes, la variance intra-classe ne peut qu'augmenter car les groupes deviennent plus hétérogènes.

Les procédures habituelles de groupage flou regroupent les individus selon un critère de proximité. A chaque groupe correspond une règle, le partitionnement des entrées et des sorties est obtenu par la projection des membres du groupe sur chacune des dimensions. La distance utilisée ne tient pas compte du partionnement, elle tient compte au mieux de la forme des groupes (distance de Mahalanobis, algorithme de Gustafson-Kessel).

Notre propos est différent : nous souhaitons regrouper des sous-ensembles flous, et non des individus, de façon itérative en choisissant le regroupement à l'aide des exemples de la base d'apprentissage. Ainsi, contrairement à ce qui se passe avec le groupage flou, le partitionnement n'a pas à être dérivé de l'opération de groupage des individus. Pour caractériser à chaque étape la partition et la valider à l'aide des exemples d'apprentissage, il nous faut un indice analogue à la variance dans la classification ascendante hiérarchique. Nous utiliserons la somme des distances entre toutes les paires de points. Pour cela nous avons besoin d'une distance entre individus qui prenne en compte la structure de la partition floue. Cette fonction de distance n'est pas disponible.

Un examen de la littérature montre que la notion de distance, en logique floue, est principalement définie entre ensembles (Dubois & Prade, 1980), pas entre points. Même si, régulièrement, des auteurs rappellent l'importance de la distance pour les opérations de groupage (Bezdek, 1981;

Hammah & Curran, 1999), aucun ne prend en compte le partitionnement dans le calcul de distance. Dans (Fan & Xie, 1999), les auteurs utilisent une distance de Hamming floue, définie pour les ensembles A et B comme $l(A,B) = \sum_{i=1}^{n} |\mu_A(x_i) - \mu_B(x_i)|$, pour générer une mesure d'entropie floue. Leur objectif est de déterminer la quantité d'information nécessaire pour discriminer deux ensembles flous. Plusieurs auteurs utilisent la distance entre ensembles dans un but de raisonnement qualitatif par interpolation. Dans une problématique d'analyse d'images, Lowen et Peeters (Lowen & Peeters, 1998) utilisent une pseudo métrique, qui ne satisfait pas l'inégalité triangulaire, tandis que Chaudhuri et Rosenfeld (Chaudhuri & Rosenfeld, 1996; Chaudhuri & Rosenfeld, 1999) utilisent une distance entre alpha-coupes, basée sur la distance de Hausdorff, qui respecte l'inégalité triangulaire. Hirota (Koczy & Hirota, 1993; Subasic & Hirota, 1998) propose une fonction de distance qui rend compte du jugement humain. La distance entre deux ensembles flous est définie comme la distance entre deux lignes, les courbes médianes de chacun des ensembles. La courbe médiane d'un ensemble flou est l'ensemble des milieux des segments des alpha-coupes. Bien que cette fonction de distance ne soit pas bijective, elle est interprétable et présente des propriétés intéressantes pour le raisonnement.

La fonction de distance qui prend en compte le partitionnement présentée dans ce chapitre est, comme nous allons le montrer, une semi-distance. Pour la conception des hiérarchies de partition le respect de l'inégalité triangulaire n'est pas nécessaire. Seules les dissimilarités entre des paires de points sont utilisées. Néanmoins, nous avons choisi de montrer que cette fonction garantit le respect de l'inégalité triangulaire pour ne pas limiter son usage à cette application.

Nous allons commencer par exposer le principe du partitionnement ascendant hiérarchique, en précisant la phase d'initialisation de la partition ainsi que le critère de fusion. La distance adaptée au partitionnement est décrite dans la section 3.2. La section 3.3 propose une amélioration de la fusion intra-dimensionnelle par la prise en compte des valeurs de sortie. Un indice qui caractérise la qualité d'une partition est introduit dans la section 3.4. La section 3.5 étudie la sensibilité de la méthode aux différents paramètres sur plusieurs jeux de données. Les principales conclusions sont rappelées en section 3.6.

3.1 PAH : la méthode

Le partitionnement ascendant hiérarchique est une procédure intra-dimensionnelle. Pour l'entrée considérée, à partir d'une partition initiale, deux sous-ensembles flous sont fusionnés à chaque étape. La procédure s'arrête lorsque la partition n'est plus formée que d'un seul sous-ensemble flou correspondant à la totalité du domaine de variation de l'entrée.

La méthode présentée dans ce chapitre est générique, seul l'opérateur de fusion dépend de la forme des sous-ensembles flous. Nous l'avons défini pour les ensembles triangulaires car ce sont les plus utilisés du fait de leurs propriétés (Pedrycz, 1994), ainsi que pour les ensembles de forme trapézoïdale.

3.1.1 Formation de la partition initiale

En toute rigueur il pourrait y avoir au départ un sous-ensemble flou par exemple d'apprentissage. En pratique, les premiers regroupements peuvent être accélérés sans dégrader les performances de la procédure. Deux approches complémentaires, et de complexité croissante, peuvent être envisagées.

1. Regroupement par valeurs uniques :
   Le test d'égalité de deux valeurs est paramétré par une tolérance. Il est important de garder

celle-ci petite pour que les groupes puissent être considérés comme des unités, la valeur de la tolérance fixe en quelque sorte le niveau de résolution.

Les valeurs des individus sont triées par ordre croissant, la valeur initiale de chacun des groupes est la première de la série, c'est-à-dire la plus petite. Un nouveau groupe est formé dès que la valeur testée est supérieure à la valeur du groupe en cours augmentée de la tolérance. La valeur associée au groupe maintenant constitué est la moyenne des valeurs des individus qui le composent. Le nombre d'occurrences dans la distribution de chacune des valeurs dites uniques est mémorisé.

2. Groupage :
   Si le nombre de valeurs après regroupement est encore important, une opération de groupage est nécessaire. L'algorithme des centres mobiles (Saporta, 1990) peut être utilisé. L'objectif est ici de former des groupes très homogènes, la somme des variances intra-groupe doit rester inférieure à un seuil donné. Cette étape éventuelle ne doit être vue que comme une phase de préparation.

Il est maintenant possible d'initialiser la partition avec les valeurs uniques ou les centres des groupes. Chacune de ces valeurs est le sommet d'un ensemble flou triangulaire. Le sous-ensemble flou f est défini par trois points, $gauche^f, c^f, droite^f$. Les pieds du triangle sont les sommets des triangles voisins, nous obtenons ainsi une partition floue forte[a]. Les relations liant trois ensembles voisins f, g, h sont :

$$\begin{cases} gauche^g = c^f \\ gauche^h = c^g \\ droite^g = c^h \end{cases}$$

Chacune des étapes de la procédure de fusion peut être décomposée en deux phases : il convient de calculer un critère de fusion pour chacune des combinaisons possibles, puis de regrouper les deux sous-ensembles flous qui satisfont au mieux ce critère. La procédure s'arrête lorsque tous les sous-ensembles flous de l'entrée ont été regroupés en un seul.

3.1.2 Fusion de deux ensembles flous

Seuls deux ensembles adjacents peuvent être fusionnés. Les coordonnées du nouvel ensemble flou sont les suivantes : la limite gauche (ou inférieure) du support est celle de l'ensemble le plus à gauche, la limite droite (ou supérieure) est celle de l'ensemble le plus à droite, la position du sommet est la moyenne pondérée des sommets des ensembles à fusionner. Les coefficients de pondération sont les masses des ensembles flous.

La masse d'un ensemble flou, f, est sa cardinalité floue, définie comme la somme des coefficients d'appartenance, pour tous les exemples d'apprentissage, à l'ensemble considéré : $w^f = \sum \mu_j^f(x)$.

Sur la figure 3.1, les ensembles A_2 et A_3 sont fusionnés. L'ensemble résultant, A_{23}, est défini par les points suivants :

$$\begin{cases} gauche^{A_{23}} = gauche^{A_2} \\ c^{A_{23}} = \dfrac{w^{A_2} c^{A_2} + w^{A_3} c^{A_3}}{w^{A_2} + w^{A_3}} \\ droite^{A_{23}} = droite^{A_3} \end{cases}$$

[a]Une partition floue est dite forte, ou standardisée, si pour chaque point de l'univers, la somme des coefficients d'appartenance de ce point, pour tous les sous-ensembles flous de la partition, vaut 1.

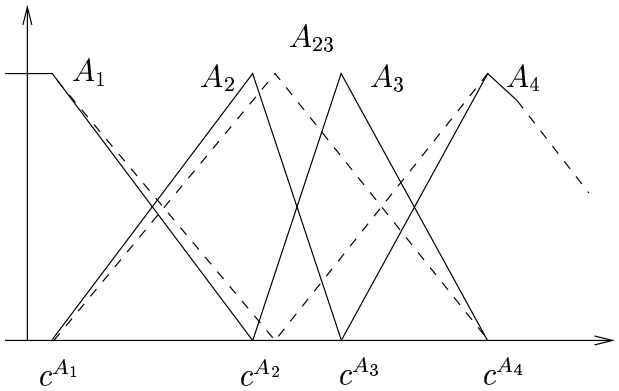


FIG. 3.1 – Processus de fusion de A_2 et A_3 en A_{23}

Les ensembles voisins du résultat de la fusion sont modifiés pour maintenir la propriété d'une partition floue forte : la limite droite de l'ensemble précédent et la limite gauche de l'ensemble suivant sont toutes deux égales à la position du nouveau sommet.

D'un point de vue sémantique, et pour maintenir la propriété d'une partition floue forte, les ensembles flous extrêmes devraient être modélisés par des demi-trapèzes. Pour des raisons pratiques d'implémentation, qui seront détaillées dans la section 3.5, ils sont modélisés, dans la partition initiale, par des triangles isocèles.

Les trapèzes constituent une alternative aux triangles pour la modélisation des ensembles flous. Dans ce cas, les deux points définissant le noyau du trapèze sont confondus lorsqu'il s'agit des valeurs uniques, ce qui correspond à un triangle. Les noyaux s'élargissent ensuite du fait du groupage et des opérations de fusion : le nouvel ensemble est défini par le pied gauche de l'ensemble précédent, le pied droit de l'ensemble suivant, le nouveau noyau est le segment délimité par la valeur minimale du noyau de l'ensemble précédent et la valeur maximale de l'ensemble suivant. La modélisation par des triangles devrait toutefois conduire à des ensembles plus progressifs.

3.1.3 Critère de fusion

Le critère de fusion est basé sur la distance entre observations que nous allons présenter dans la prochaine section. Soulignons dès à présent une de ses caractéristiques essentielles.

Pour respecter la sémantique attachée à la partition floue, nous imposons que la distance entre deux points qui appartiennent au même sous-ensemble flou, dite distance interne, soit inférieure à la distance entre deux points qui appartiennent à deux sous-ensembles flous différents, appelée distance externe. Cette condition reflète la plus grande proximité de deux points qui appartiennent à un même ensemble par rapport à deux points qui appartiennent à des ensembles distincts. La mise en œuvre de cette contrainte, rendue délicate par le fait que les points peuvent appartenir à plusieurs ensembles de la partition, sera étudiée en détail dans la section 3.2.4.

Ainsi, la somme des distances pour toutes les paires d'exemples dépend à la fois du nombre de sous ensembles flous et de leurs positions au sein de la partition. Elle caractérise une partition par rapport à un ensemble d'apprentissage.

Soient s la taille de la partition, c'est-à-dire le nombre d'ensembles flous qui la composent, n le cardinal de l'ensemble d'apprentissage, et $d(q,r)$ une fonction de distance entre les points q et r, la somme des distances, D_s, est définie comme :

$$D_s = \frac{1}{n(n-1)} \sum_{q,r=1,2,..,n,\ q \neq r} d(q,r) \tag{3.1}$$

Au cours du processus de fusion de deux sous-ensembles flous, certaines distances externes deviennent des distances internes induisant une variation de la somme des distances entre toutes les paires de points.

Comme dans la classification ascendante hiérarchique, nous voulons maintenir au mieux la structure construite, comme étant la plus homogène, à l'étape précédente. Nous proposons donc de fusionner la paire d'ensembles flous qui minimise la variation de la somme des distances. Du fait de la contrainte imposée à la distance interne d'être inférieure à la distance externe, cette variation est, sauf quelques cas particuliers en début de procédure, négative, ce qui correspond à une baisse de la somme des distances. La combinaison de fusion retenue, f, lorsqu'on passe d'une partition de taille s à une de taille $s-1$, sera celle, parmi les $s-1$ possibles, telle que :

$$D_{s-1} = argmax \left(\frac{D^f_{s-1}}{D_s} \right) \qquad \forall f = 1, \ldots, s-1 \tag{3.2}$$

Cet algorithme de fusion est de complexité réduite. Considérons que la variable i est découpée en f_i ensembles flous à une étape donnée. Comme seuls deux ensembles flous adjacents peuvent être fusionnés, le nombre de cas à étudier est $f_i - 1$ pour l'entrée i. Le calcul des distances peut être fait sur les prototypes issus du pré-traitement, à savoir les valeurs uniques à la tolérance près. Ce nombre peut être raisonnablement borné, et ne dépend plus du cardinal de l'ensemble d'apprentissage.

La section suivante présente la fonction de distance entre points qui tient compte du partitionnement.

3.2 Une distance adaptée au partitionnement

Comme les points peuvent appartenir à plusieurs sous-ensembles flous de la partition, la distance entre deux points, q et r, notée $d(q,r)$, sera une combinaison de distances partielles. Deux cas, non exclusifs, sont à envisager. Lorsque l'on considère les degrés d'appartenance à un même ensemble flou nous parlerons de distance partielle interne. Dans le cas contraire, la distance partielle sera dite externe.

La distance interne est notée d_{int}. Elle est définie dans la section 3.2.1.

Lorsque les points appartiennent à des ensembles différents, il faut considérer la distance externe. Celle-ci, notée d_{ext}, résulte d'une combinaison, l'opérateur est noté $\odot$, entre la distance interne et la distance entre les noyaux des deux ensembles, encore appelée distance entre prototypes et notée d_{prot} :

$$d_{ext}(q,r) = d_{int}(q,r) \odot d_{prot}(A_q, A_r)$$

Les distances interne et externe, ainsi que leur combinaison, doivent vérifier les propriétés mathématiques d'une distance. Xavier Perrier (Perrier, 1998) utilise les définitions suivantes :

Étant donnés trois points, $q,\ r,\ s$, une fonction d est une dissimilarité si :

$$\forall\ q, r \qquad \begin{cases} d(q,r) \geq 0 \\ d(q,q) = 0 \\ d(q,r) = d(r,q) \end{cases} \tag{3.3}$$

Une dissimilarité est dite semi-propre si :

$$d(q,r) = 0 \quad \Rightarrow \quad \forall\ s \qquad d(q,s) = d(r,s) \tag{3.4}$$

Elle est dite propre si :

$$d(q,r) = 0 \quad \Rightarrow \quad q = r \tag{3.5}$$

Une semi-distance est une dissimilarité qui respecte l'inégalité triangulaire :

$$\forall\ q, r, s \qquad d(q,r) \leq d(q,s) + d(r,s) \tag{3.6}$$

Une distance est une semi-distance propre.

Nous pouvons maintenant présenter les différentes distances et vérifier leurs propriétés. Cette étude est réduite aux ensembles flous convexes et normaux[b], tels que ceux usuellement utilisés dans les systèmes d'inférence floue. Les fonctions triangulaires illustreront nos propos.

3.2.1 Distance interne

Le degré d'appartenance du point q à l'ensemble flou considéré, μ_q, indique le degré de similitude de q avec les prototypes, le noyau, de cet ensemble. La distance d'un point q à un ensemble flou peut être définie par la quantité $1 - \mu_q$, c'est-à-dire le complément de son degré d'appartenance à l'ensemble. La distance entre deux points, est dans ce contexte, la différence entre leur distance aux prototypes, soit tout simplement la différence de leur degrés d'appartenance à l'ensemble. La distance interne est définie par la fonction d_{int} suivante :

$$d_{int}(q,r) = |\mu_q - \mu_r| \qquad donc \quad \forall q, r \quad 0 \leq d_{int}(q,r) \leq 1$$

Il est aisé de vérifier que c'est une dissimilarité semi-propre. La condition mentionnée dans l'équation 3.5 n'est pas toujours vérifiée. Il existe de nombreuses paires de points distincts qui ont un même degré d'appartenance, et donc pour lesquels la distance interne est nulle. Cette situation, illustrée par la figure 3.2, se rencontre également au sein du noyau des ensembles flous de forme trapézoïdale par exemple.

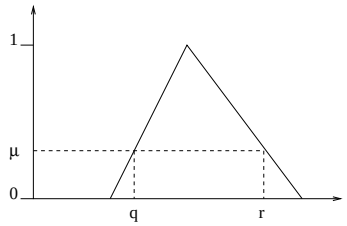


FIG. 3.2 – La distance interne est nulle

Le respect de l'inégalité triangulaire est à vérifier dans les trois cas suivants :

[b]Un ensemble flou est dit normal, ou normalisé, si la borne supérieure de la fonction d'appartenance vaut 1.

1. Cas trivial : les trois points sont tels que $\mu_q = \mu_r = \mu_s$.
2. Deux d'entre eux ont le même degré d'appartenance, soit $\mu_q = \mu_r$ et $\mu_q \neq \mu_s$, avec par exemple $\mu_q > \mu_s$. Il faut dans ce cas vérifier les trois inégalités possibles :

$$\begin{array}{lll} d_{int}(q,r) \leq d_{int}(q,s) + d_{int}(r,s) & soit & 0 \leq 2(\mu_q - \mu_s) \\ d_{int}(q,s) \leq d_{int}(q,r) + d_{int}(r,s) & soit & \mu_q - \mu_s \leq \mu_q - \mu_s \\ d_{int}(r,s) \leq d_{int}(q,r) + d_{int}(q,s) & soit & \mu_q - \mu_s \leq \mu_q - \mu_s \end{array}$$

3. Les degrés d'appartenance sont tous différents, soit par exemple $\mu_q < \mu_r < \mu_s$. Les inégalités s'écrivent alors :

$$\begin{array}{lll} \mu_r - \mu_q \leq \mu_s - \mu_q + \mu_s - \mu_r & soit & \mu_r \leq \mu_s \\ \mu_s - \mu_q \leq \mu_r - \mu_q + \mu_s - \mu_r & soit & \mu_s - \mu_q \leq \mu_s - \mu_q \\ \mu_s - \mu_r \leq \mu_r - \mu_q + \mu_s - \mu_q & soit & \mu_q \leq \mu_r \end{array}$$

Les inégalités étant vraies dans tous les cas, la fonction de distance interne proposée, d_{int}, est une semi-distance au sein d'un ensemble flou.

3.2.2 Distance entre prototypes

Cette mesure, pour être indépendante des unités des différentes variables, doit être normalisée, Ainsi, $\forall q, r \quad 0 \leq d_{prot}(A_q, A_r) \leq 1$.

Considérons d'abord les variables pour lesquelles les sous-ensembles flous correspondent à un ordre : typiquement de *Petit* vers *Grand*. La relation d'ordre est la relation *prcde* telle que $Petit \prec Moyen \prec Grand$. Le domaine de variation ramené dans un espace [0 1], deux options sont alors possibles :

1. Définir la distance comme la distance euclidienne des noyaux. Il s'agit de la distance entre les sommets pour des triangles, ou bien de la distance entre les deux extrémités des noyaux les plus proches pour des trapèzes.
2. Cette distance peut être rendue encore plus fidèle à la représentation symbolique : à chacun des sous-ensembles flous est associé un numéro d'ordre, et la distance est définie comme la différence entre ces numéros divisée par la distance maximum, c'est-à-dire le nombre de sous-ensembles flous de la partition diminué de un.

La première sera notée distance numérique par opposition à la seconde qui est appelée distance symbolique. Cette dénomination constitue un abus de langage car la distance proposée n'est pas floue. Elle reflète néanmoins le rôle important de la représentation symbolique dans son mode de calcul.

Dans l'exemple de la figure 3.3, la distance numérique entre les sous-ensembles flous triangulaires A_1 et A_4 vaut : $d_{prot}(A_1, A_4) = \dfrac{x_4 - x_1}{x_5 - x_1}$ et la distance symbolique vaut $d_{prot}(A_1, A_4) = \dfrac{4-1}{5-1}$. Selon cette dernière fonction de distance l'ensemble A_3 est aussi proche de l'ensemble A_2 que de l'ensemble A_4.

Pour les ensembles flous de forme trapézoïdale, comme sur la figure 3.4, la distance numérique entre A_1 et A_3 vaut $x_{31} - x_1$ et celle entre A_2 et A_3 vaut $x_{31} - x_{22}$, avec la normalisation $x_4 - x_1$.

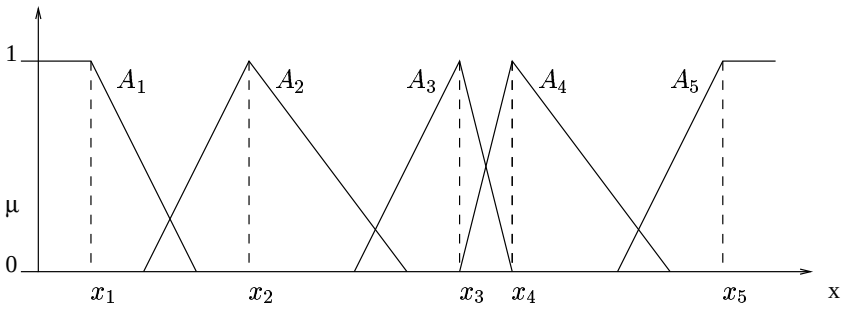


FIG. 3.3 – Position des noyaux dans une partition formée de triangles

Il est aisé de vérifier que les distances entre prototypes proposées vérifient l'inégalité triangulaire. Notons que dans le cas des triangles, cette inégalité est une égalité lorsque les ensembles sont ordonnés au sens de la relation *prcde* :

$$\text{si } A_q \prec A_r \prec A_s \text{ alors } d_{prot}(A_q, A_s) = d_{prot}(A_q, A_r) + d_{prot}(A_r, A_s).$$

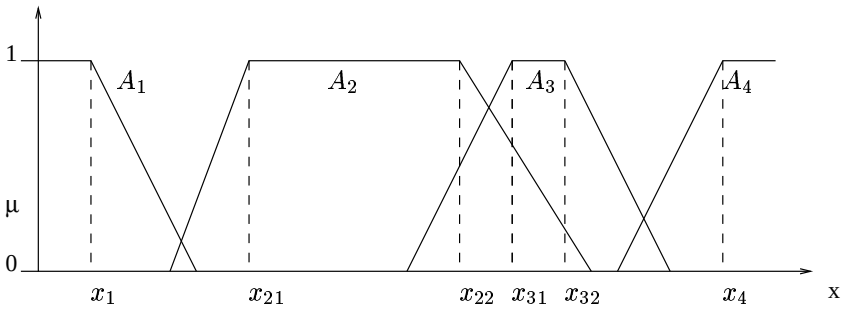


FIG. 3.4 – Position des noyaux dans une partition formée de trapèzes

Dans le cas particulier des variables qualitatives non ordonnées, la distance entre deux sous-ensembles flous différents est une constante à déterminer, indépendante des sous-ensembles flous et de leur nombre dans la partition. Dans ce cas aussi, la vérification de l'inégalité triangulaire est triviale.

3.2.3 Distance externe

La distance externe doit tenir compte de deux éléments : la position de chacun des points au sein de son ensemble flou, et la position relative des deux ensembles flous dans la partition. Nous proposons d'étudier deux opérateurs de combinaison, notés $\odot$, définissant deux distances externes. Le premier est additif, le second basé sur un produit.

Dans un premier temps, par souci de simplification, nous étudions la distance externe à l'aide de points qui appartiennent à un seul sous-ensemble flou de la partition.

Soit la fonction : $d_{ext}(q,r) = |\mu_q^{A_q} - \mu_r^{A_r}| + d_{prot}(A_q, A_r)$.
La distance externe est bornée : $\forall q, r \quad 0 \leq d_{ext}(q,r) \leq 2$, et vérifie l'inégalité triangulaire.

Dans l'exemple de la figure 3.5, les distances s'écrivent :

$$d_{ext}(q,r) = \mu_r^{A_3} - \mu_q^{A_2} + d_{prot}(A_2, A_3)$$
$$d_{ext}(r,s) = \mu_r^{A_3} - \mu_s^{A_4} + d_{prot}(A_3, A_4)$$
$$d_{ext}(q,s) = \mu_s^{A_4} - \mu_q^{A_2} + d_{prot}(A_2, A_4)$$

La vérification de l'inégalité triangulaire est immédiate.

Lorsque les deux coefficients d'appartenance sont égaux, la distance externe est égale à la distance entre les prototypes des ensembles. C'est un résultat intéressant, qui a un sens : lorsque deux points présentent la même dissimilarité au prototype de l'ensemble flou auquel ils appartiennent, leur distance externe est égale à la distance entre ces prototypes.

Une deuxième fonction peut être envisagée : $d_{ext} = |\mu_q^{A_q} - \mu_r^{A_r}| \times \Big(d_{prot}(A_q, A_r) + 1\Big)$. Cette fonction conduit à une distance nulle, quelle que soit la distance entre les prototypes, si les degrés d'appartenance des deux points sont égaux. De ce fait, elle ne vérifie pas l'inégalité triangulaire dans tous les cas, comme le montre l'exemple suivant.

Sur la figure 3.6, les distances s'écrivent :

$$d_{ext}(q,r) = (\mu_r^{A_3} - \mu_q^{A_2}) \times \Big(d_{prot}(A_2, A_3) + 1\Big)$$
$$d_{ext}(r,s) = (\mu_r^{A_3} - \mu_s^{A_4}) \times \Big(d_{prot}(A_3, A_4) + 1\Big)$$
$$d_{ext}(q,s) = (\mu_s^{A_4} - \mu_q^{A_2}) \times \Big(d_{prot}(A_2, A_4) + 1\Big)$$

Comme $\mu_s^{A_4} = \mu_q^{A_2}$, $d_{ext}(q,s) = 0$ et donc pour vérifier l'inégalité triangulaire, $d_{ext}(q,r)$ doit être inférieure ou égale à $d_{ext}(r,s)$. Cependant, la différence des degrés d'appartenance étant la même, ceci n'est vérifié que lorsque $d_{prot}(A_2, A_3) \leq d_{prot}(A_3, A_4)$. L'inégalité $d(q,s) \leq d(q,r) + d(r,s)$ n'est, quant à elle, vérifiée que lorsque les deux autres distances sont nulles aussi.

Ces deux éléments, distance nulle lorsque les degrés d'appartenance des deux points sont égaux et non respect de l'inégalité triangulaire, nous interdisent l'utilisation de cette fonction basée sur le produit comme fonction de distance.

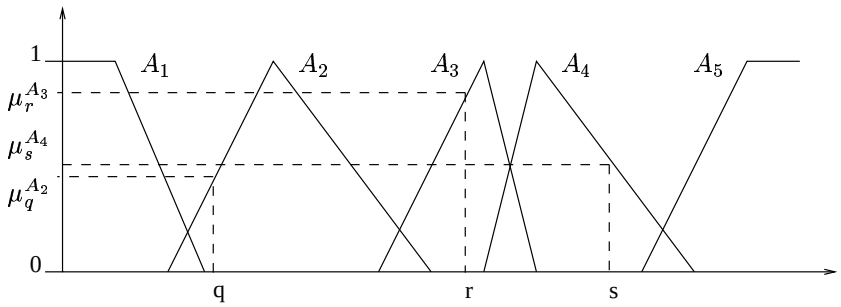


FIG. 3.5 – Une illustration de distances externes

3.2.4 Harmonisation

Pour rendre compte du fait que deux individus qui appartiennent principalement au même sous-ensemble flou sont toujours plus proches l'un de l'autre que deux qui appartiennent principalement à des ensembles distincts, la distance externe doit être supérieure à la distance interne. Il est nécessaire de corriger l'une ou l'autre des distances, interne ou externe, afin de garantir cette assertion. Deux corrections peuvent être envisagées : la première est multiplicative, la seconde additive.

Correction multiplicative

La correction multiplicative peut s'appliquer à la distance interne ou bien à la distance externe.

La distance interne peut être normalisée pour qu'elle soit toujours plus petite que la distance externe. Dans le cas de la distance symbolique entre prototypes, la distance interne pourrait être divisée par le nombre d'ensembles flous de la partition, f :

$$d_{int_s} = \frac{|\mu_q - \mu_r|}{f}$$

Cette correction présente l'avantage de rendre compte d'une autre réalité : la distance interne entre deux points est d'autant plus petite que le nombre d'ensembles flous dans la partition est grand, c'est-à-dire que le fait d'appartenir à un même ensemble est, en soi, un critère de proximité important lorsqu'il y a beaucoup d'ensembles.

Ainsi la valeur maximale de la distance interne est $\frac{1}{f}$, et la valeur minimale de la distance externe, qui correspond à une différence des degrés d'appartenance nulle pour deux ensembles flous adjacents est également de $\frac{1}{f}$. La normalisation proposée garantit dans tous les cas que la distance interne est inférieure ou égale à la distance externe.

En contrepartie, comme la distance dépend du nombre d'ensembles flous, il est impossible d'interpréter l'évolution de ces distances au cours du processus de fusion. Les valeurs absolues n'ont de sens qu'à une étape donnée. Cette remarque est également valable pour la distance symbolique entre prototypes.

La correction est moins simple pour la distance dite numérique. En effet, il faut garantir que la distance interne normalisée maximum soit toujours inférieure à la plus petite distance entre les prototypes des deux ensembles flous les plus proches. Or celle-ci est impossible à borner *a priori*. Il en résulte que le coefficient de normalisation ne peut être calculé que de façon dynamique, pour une

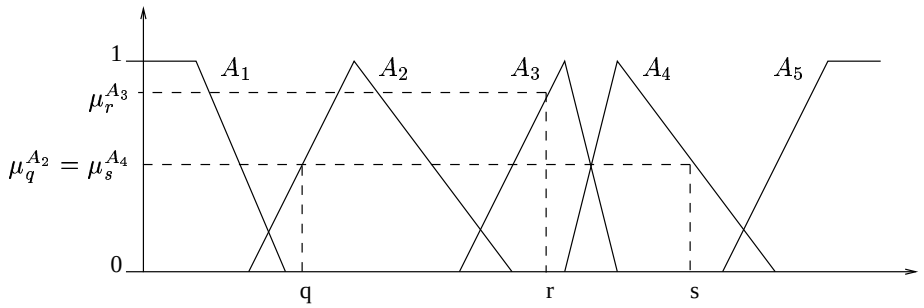


FIG. 3.6 – Inégalité triangulaire et distance externe : deux points ont le même degré d'appartenance

partition donnée. Ce coefficient peut tendre vers l'infini si la distance entre prototypes tend vers zéro.

La multiplication de la distance externe présente le même inconvénient numérique. Elle pourrait s'écrire :

$$d_{ext}(q,r) = |\mu_q^{A_q} - \mu_r^{A_r}| + (d_{prot}(A_q, A_r) * m)$$

Mais le coefficient m devrait aussi tendre vers l'infini si la distance entre deux prototypes tend vers zéro.

Ces considérations numériques nous conduisent à rejeter la correction multiplicative pour la distance dite numérique.

Correction additive

Elle ne peut s'effectuer que sur la distance externe. En effet, il faut garantir que la distance interne soit inférieure à la plus petite distance externe, c'est-à-dire la plus petite distance entre prototypes qui peut tendre vers zéro. Si on appliquait cette correction sur la distance interne, sa positivité ne serait plus garantie.

La proposition consiste à ajouter à la distance externe la valeur maximale que peut prendre la distance interne, m. Cette valeur vaut 1 dans le cas le plus général. La distance entre les points q et r devient :

$$d_{ext}(q,r) = |\mu_q^{A_q} - \mu_r^{A_r}| + d_{prot}(A_q, A_r) + m$$

La présence de ce terme additif induit une perte de généralité : $d_{ext}(q,q) = m$, ce qui est différent de 0. La correction ne peut donc s'appliquer qu'à des ensembles différents. De plus l'inégalité triangulaire est toujours stricte. Ce fait s'illustre simplement en considérant la distance entre les noyaux :

$$\begin{aligned} & m + d_{prot}(A_1, A_3) \leq m + d_{prot}(A_1, A_2) + m + d_{prot}(A_2, A_3) \\ & comme\ d_{prot}(A_1, A_3) = d_{prot}(A_1, A_2) + d_{prot}(A_2, A_3) \\ & m \leq 2 * m \end{aligned}$$

La correction additive n'est pas sensible à la répartition des sous-ensembles flous dans la partition et, bien qu'inélégante, elle respecte les propriétés mathématiques de la semi-distance.

La correction sera donc additive, appliquée à la distance externe.

3.2.5 Continuité et combinaison

Dans une partition floue les points peuvent avoir des degrés d'appartenance non nuls pour plusieurs sous-ensembles de la partition. Dans ce cas, la distance entre deux points, $d(q,r)$, est une combinaison de distances internes et externes, en fonction des ensembles pour lesquels les degrés d'appartenance μ_q et μ_r sont non nuls.

Les fonctions de distance présentées font simplement l'hypothèse que les sous-ensembles flous de la partition sont normaux (le degré d'appartenance maximum vaut 1) et convexes. Pour préciser la combinaison des distances dans toutes les configurations possibles, nous allons tenir compte du contexte particulier de notre travail.

Nous maintenons tout au long du processus de fusion une partition floue forte, comme illustrée sur la figure 3.7. Ainsi, pour tout point, la somme des degrés d'appartenance vaut un et le nombre

maximal de degrés d'appartenance non nuls est limité à deux. Les propriétés d'une telle partition sont présentées en détail dans des publications récentes (Glorennec, 1999; de Oliveira, 1999; Espinosa & Vandewalle, 2000).

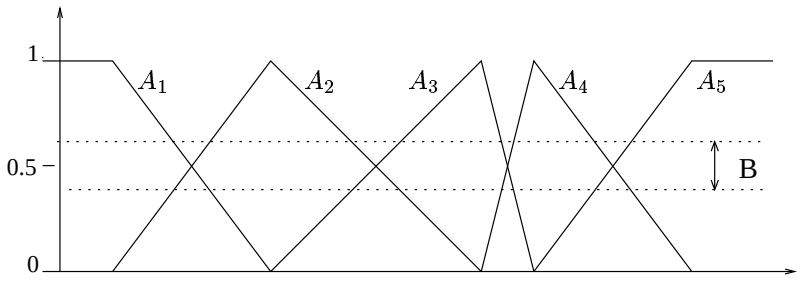


FIG. 3.7 – Une partition floue forte

Considérons donc le cas le plus compliqué dans lequel chacun des deux points possède une double appartenance :

$$\begin{cases} \mu_q^f > 0, & \mu_q^g > 0 \\ \mu_r^h > 0, & \mu_r^l > 0 \end{cases}$$

Notons $d_{f,h}(q,r)$ la part de distance pour la paire (q,r) qui représente leurs degrés d'appartenance respectifs à f et h. Si $f = h$ il s'agit d'une distance interne, sinon d'une distance externe.

$d(q,r)$ résulte de la combinaison de quatre distances partielles :

$$d(q,r) = \frac{1}{\mu_q^f + \mu_q^g} \left(\mu_q^f * \frac{\mu_r^h * d_{f,h}(q,r) + \mu_r^l * d_{f,l}(q,r)}{\mu_r^h + \mu_r^l} + \mu_q^g * \frac{\mu_r^h * d_{g,h}(q,r) + \mu_r^l * d_{g,l}(q,r)}{\mu_r^h + \mu_r^l} \right) \tag{3.7}$$

Dans le cas particulier d'une partition floue forte, les dénominateurs sont tous égaux à 1.

Remarquons que la combinaison des distances partielles, établie pour rendre compte de l'appartenance de chacun des points à deux sous-ensembles flous, se généralise facilement à un nombre quelconque d'ensembles. Elle peut être étendue au cas d'une partition floue quelconque. Son expression est dans ce cas :

$$d(q,r) = \frac{1}{\sum_{f=1}^{m} \mu_q^f} \sum_{f=1}^{m} \left[\mu_q^f \frac{1}{\sum_{h=1}^{m} \mu_r^h} \sum_{h=1}^{m} \left[\mu_r^h \; d_{f,h}(q,r) \right] \right] \tag{3.8}$$

Nous dirons qu'un point q appartient principalement à un ensemble flou f si son degré d'appartenance est tel que $\mu_q^f \geq 0.5$. En conséquence, la distance entre deux points sera principalement interne si les deux points appartiennent principalement au même sous-ensemble flou. Pour imposer que de tels points soient toujours plus proches que deux points dont la distance est principalement externe, il suffit de considérer la plus grande des distances principalement internes, soit 0.5. C'est cette valeur qui constituera le terme additif, m, pour corriger la distance externe, et non la valeur de 1 comme dans le cas général indiqué en section 3.2.4.

La dissimilarité d ainsi définie est une combinaison de semi-distances dont les coefficients sont positifs ou nuls. C'est donc une semi-distance. Dans un soucis d'allégement des notations, d sera désormais appelée une distance.

Remarque : La gestion de la multi appartenance peut être simplifiée. Nous pouvons considérer que tous les points appartiennent principalement à un ensemble flou, celui pour lequel leur degré d'appartenance est maximum, à l'exception de ceux dont les degrés sont situés dans une bande centrée sur 0.5, comme illustré sur la figure 3.7. Ceci revient à ne pas tenir compte des degrés inférieurs à la limite de la bande. Dans les formules qui suivent, ces degrés non significatifs sont égaux à zéro.

Considérant que $x_q < x_r$, les sous-ensembles flous pour lesquels le degré d'appartenance de x_r est non nul ne sont pas situés *avant* ceux pour lesquels le degré de x_q est non nul, au sens de la relation *prcde* introduite en section 3.2.2, trois cas sont à envisager :

1. Chacun des deux points appartient à un seul ensemble flou, la distance est alors soit purement interne, soit purement externe :

$$d(q,r) = \begin{cases} d_{int}(q,r) = |\mu_q^f - \mu_r^g| & si \quad f = g \\ d_{ext}(q,r) = |\mu_q^f - \mu_r^g| + d_{prot}(f,g) + 0.5 & si \quad f \neq g \end{cases} \tag{3.9}$$

2. L'un des points, q, appartient de façon significative à deux sous-ensembles flous, f et g. L'autre appartient seulement à h. Sachant que $x_q < x_r$, la distance devient :

$$d(q,r) = \begin{cases} |\mu_q^g - \mu_r^g| + \frac{1}{2}\Big(d_{prot}(f,g) + 0.5\Big) & si \quad g = h \\ d_{ext}(q,r) = |\mu_q^g - \mu_r^h| + d_{prot}(g,h) + 0.5 & si \quad g \neq h \end{cases} \tag{3.10}$$

   Les formules de l'équation 3.10 constituent une approximation de la moyenne des distances externe et interne pour la paire de points (q,r). Cette approximation provient de fait que les degrés d'appartenance du même point aux deux ensembles sont considérés comme voisins, $\mu_q^f \approx \mu_q^g$. Cette hypothèse est d'autant plus vraie que la bande est étroite, car l'erreur est majorée par $\dfrac{B}{2}$, B étant la largeur de la bande.

3. Les deux points présentent une double appartenance significative :

$$\begin{cases} \mu_q^f > \alpha, \quad \mu_q^g > \alpha \\ \mu_r^h > \alpha, \quad \mu_r^l > \alpha \end{cases} \qquad avec \qquad \alpha = 0.5 - \tfrac{B}{2} \tag{3.11}$$

   Au lieu de combiner les quatre distances, une alternative économique consiste à ne conserver que la plus petite. La distance est alors calculée suivant l'équation 3.9 et correspond soit à une distance interne, soit à une distance externe.

3.3 Hétérogénéité d'une partition

L'application du critère de fusion défini dans l'équation 3.2 revient à considérer que les entrées sont indépendantes des autres entrées et de la sortie. Pour tenir compte du contexte, nous proposons de modifier la valeur de ce critère en prenant appui sur la sortie seulement. Ce choix permet de conserver le caractère atomique, et donc rapide, de la fusion intradimensionnelle et évite une procédure multidimensionnelle qui n'aurait pu être que simpliste, avec comme conséquence la réduction voire la non prise en compte des interactions, ou bien très lourde.

L'idée proposée consiste à caractériser chacun des sous-ensembles flous d'une entrée i donnée par l'homogénéité des sorties[c]. L'homogénéité d'un sous-ensemble flou sera mesurée au sein de l'ensemble des exemples d'apprentissage dont la valeur de la variable d'entrée i appartient à ce sous-ensemble flou. Cette mesure présente l'avantage de traiter de la même manière les problèmes de classification et de régression.

3.3.1 Mesure de l'hétérogénéité

La mesure de l'hétérogénéité proposée est tout simplement la variance de la sortie pondérée par le degré d'appartenance des exemples à l'ensemble flou.

L'hétérogénéité d'un sous-ensemble flou s, μ_k^s représentant le degré d'appartenance de l'exemple k à l'ensemble s, s'écrit :

$$H_s = \frac{1}{\sum_{k=1}^{n} \mu_k^s} \sum_{k=1}^{n} \mu_k^s (y_k - \overline{y})^2$$

Comme :

$$\overline{y} = \frac{\sum_{k=1}^{n} \mu_k^s y_k}{\sum_{k=1}^{n} \mu_k^s}$$

La formule programmée est :

$$H_s = \frac{1}{\sum_{k=1}^{n} \mu_k^s} \left(\sum_{k=1}^{n} \mu_k^s y_k^2 - \frac{\left(\sum_{k=1}^{n} \mu_k^s y_k \right)^2}{\sum_{k=1}^{n} \mu_k^s} \right)$$

Pour rester indépendant des unités de la sortie, les valeurs, comme celles des entrées, seront normalisées dans l'intervalle unité.

Un sous-ensemble flou, s, sera qualifié d'hétérogène si son hétérogénéité, H_s, est supérieure à un seuil. En classification[d], la valeur du seuil, H_p^c, dépend du nombre de classes, c, et de la proportion d'impureté tolérée, p.

La variance minimale d'un sous-ensemble flou hétérogène se rencontre lorsque les exemples regroupés au sein de l'ensemble flou appartiennent à deux classes dont les effectifs sont p et $1-p$ et dont les numéros sont consécutifs.

Soient y_i et y_j les numéros des deux classes normalisés dans l'espace unité, s'ils sont consécutifs ils sont liés par la relation $|y_i - y_j| = \frac{1}{c-1}$. La position du centre de gravité s'écrit simplement : $\overline{y} = p\ y_i\ +\ (1-p)\ y_j$.

Pour le calcul du seuil, nous considérons que les degrés d'appartenance sont voisins. Ainsi l'expression de l'hétérogénéité devient :

[c] Dans le cadre de l'apprentissage supervisé la valeur de sortie fait partie de la description des exemples.
[d] Nous supposons que les valeurs des sorties sont les numéros des classes : $1, \ldots, c$.

$$H = \sum_{l} (y_l - \overline{y})^2 = p\ (y_i - \overline{y})^2 + (1-p)\ (y_j - \overline{y})^2$$

Or :

$$y_i - \overline{y} = (1-p)\ (y_i - y_j)$$
$$y_j - \overline{y} = p\ (y_j - y_i)$$
$$(y_i - y_j)^2 = (y_j - y_i)^2 = \frac{1}{(c-1)^2}$$

Soit :

$$H = p\ (1-p)^2\ \frac{1}{(c-1)^2} + (1-p)\ p^2\ \frac{1}{(c-1)^2}$$
$$H = \Big((1-p) + p\Big) \left(p\ (1-p)\ \frac{1}{(c-1)^2} \right)$$

Et finalement, l'expression de l'hétérogénéité seuil en classification est :

$$H_p^c = p\ (1-p)\ \frac{1}{(c-1)^2}$$

Lorsque la variable de sortie est continue, le domaine de variation, normalisé dans l'espace unité, est découpé en sous-ensembles flous. La valeur prise en compte pour le calcul de l'hétérogénéité est le numéro d'ordre du sous-ensemble flou auquel appartient la valeur de sortie. Lorsque celle-ci est située à la frontière de deux ensembles[e], cette valeur est la moyenne des deux numéros d'ordre correspondants.

L'hétérogénéité d'une partition est définie comme la somme de l'hétérogénéité de chacun des sous-ensembles flous pondérée par leur masse. Notons w^s la cardinalité du sous-ensemble flou s, l'hétérogénéité s'écrit :

$$H = \frac{\sum_{s=1}^{f} w^s\ H_s}{\sum_{s=1}^{f} w^s} \tag{3.12}$$

Comme dans la classification ascendante hiérarchique, la courbe d'évolution de l'hétérogénéité des partitions, analogue à celle d'évolution de la variance intra-classe, est susceptible de fournir des informations relatives à la qualité du partitionnement.

3.3.2 Utilisation de la mesure d'hétérogénéité pour pénaliser certaines fusions

L'objectif est de retarder la fusion de deux ensembles si le degré d'hétérogénéité de l'ensemble résultant de la fusion est différent des degrés d'hétérogénéité de chacun des ensembles flous fusionnés.

Nous illustrerons notre propos à l'aide d'un jeu de données artificielles appelé $Test2D$, représenté sur la figure 3.8. Il s'agit d'un ensemble de données bi-dimensionnelles reparties en 3 classes. Deux classes sont partiellement mélangées, tandis que la troisième est bien isolée. Les lignes pointillées indiquent une partition possible pour refléter la structure des données. La minimisation de la taille des zones hétérogènes conduit à un découpage en 12 zones dont une seule, la zone $(3,3)$, est hétérogène. Certaines de ces zones peuvent aussi être regroupées.

[e] Dans le cas particulier d'une partition floue forte cette condition est vérifiée dès lors que le degré d'appartenance appartient à la bande centrale, soit par exemple $0.4 \leq \mu \leq 0.6$.

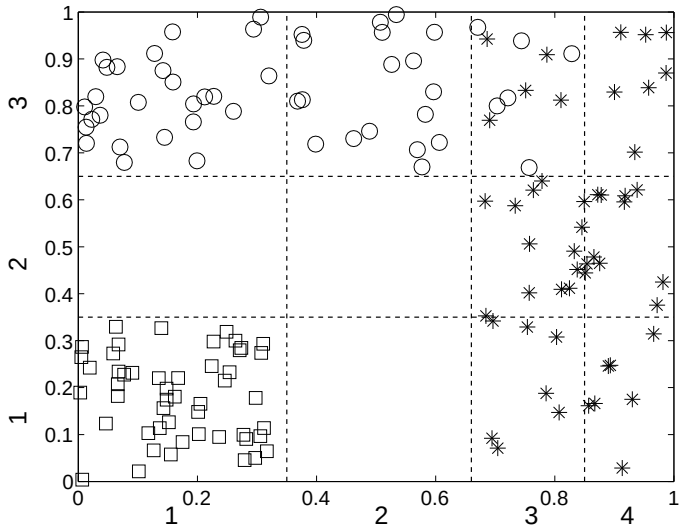


FIG. 3.8 – Le jeu de données Test2D

Par exemple la fusion des sous-ensembles flous centrés sur 0.24 et sur 0.49 sur l'axe des abscisses conduirait à la perte de l'homogénéité de l'ensemble centré sur 0.49. La fusion de deux ensembles également hétérogènes, comme ceux centrés sur 0.12 et 0.16 sur l'axe des abscisses, ne sera pas pénalisée.

Quatre cas sont donc à prendre en compte pour la fusion :

1. Les deux candidats sont hétérogènes et le résultat de la fusion l'est aussi. Cette fusion ne dégrade pas la structure existante, elle correspond par exemple à la fusion des ensembles centrés sur 0.12 et 0.16 sur l'axe des abscisses.
2. Les deux ensembles sont homogènes et la fusion l'est aussi. Pas de problème non plus, cette situation correspond par exemple aux ensembles centrés sur 0.45 et 0.49 sur le même axe.
3. Les deux ensembles sont homogènes et la fusion ne l'est pas. Dans ce cas, le processus conduit à mélanger deux sous-ensembles flous qui étaient jusque là bien identifiés. Avant de l'envisager, il vaut mieux, même si le critère des distances n'est plus respecté, combiner d'autres sous-ensembles flous pour préserver cette représentation.
4. L'un des ensemble est hétérogène, l'autre pas. Pour les mêmes raisons que précédemment, il est souhaitable de retarder cette fusion.

Pour retarder la fusion de deux sous-ensembles flous, il suffit de pénaliser la valeur du critère de fusion présenté dans l'équation 3.2. La pénalité consiste à multiplier le rapport des distances par une valeur paramétrable.

Comme le principe de la méthode réside dans la conservation à chaque étape de la structure de l'étape précédente, il convient d'éviter d'user prématurément du critère d'hétérogénéité. Les premiers pas de la fusion intradimensionnelle seront donc réalisés suivant le seul critère de la variation minimale de la somme des distances, les derniers, dix par exemple, pourront prendre en compte d'éventuelles pénalités.

3.4 Un indice de qualité

Comment définir une bonne partition ? La question, largement étudiée, demeure sans réponse dans le cas général. Dans le cadre de l'apprentissage supervisé, il est possible de construire les systèmes d'inférence floue correspondants aux partitions induites et d'évaluer leurs performances quantitatives respectives. Il faudrait commencer par le plus simple, une seule règle correspondant à un seul ensemble flou par entrée, et augmenter la complexité jusqu'à ce que l'erreur augmente.

Nous voulons aussi essayer de nous affranchir de cette contrainte et tenter de qualifier la partition elle-même, indépendamment d'un éventuel usage ultérieur. Nous allons utiliser l'homogénéité des densités des sous-ensembles flous de la partition. La densité d'un sous-ensemble flou est définie comme le rapport de sa cardinalité à son aire. Pour prendre en compte le fait que les limites des sous-ensembles flous sont imposées par la propriété de partition floue forte, l'aire sera diminuée des parties communes aux sous-ensembles flous voisins comme le montre la figure 3.9. L'homogénéité des densités sera mesurée par l'écart type des densités des sous ensembles flous de la partition, comme l'indique l'équation 3.13.

$$\sigma_d = \sqrt{\frac{1}{m}\sum_{f=1}^{m}(d_f - \overline{d})^2} \tag{3.13}$$

$\overline{d}$ étant la densité moyenne. Une bonne partition est susceptible d'être homogène, équilibrée, c'est-à-dire d'avoir un écart type plutôt petit. Les effets de bords, l'aire est plus importante aux extrémités, sont intégrés dans la statistique.

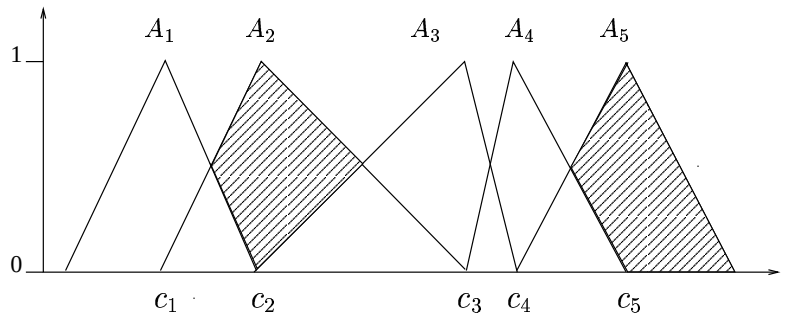


FIG. 3.9 – Aires des ensembles A_2 et A_5

Ce critère n'est pas significatif au début de la procédure car les ensembles sont tous très bien remplis. Il prend tout son sens au cours des dernières étapes de la fusion. La meilleure partition, au sens de ce critère, sera celle pour laquelle sa valeur est minimum. Nous prêterons également attention aux éventuels points singuliers de sa courbe d'évolution au cours du processus.

Nous allons examiner ce critère sur l'ensemble $Test2D$ présenté dans la section précédente.

Le tableau 3.1 montre l'évolution des centres des sous-ensembles flous et des indicateurs pour le jeu de données $Test2D$. Dans ce tableau σ_d, équation 3.13, correspond à l'écart type des densités des sous-ensembles flous de la partition et H, équation 3.12, à l'hétérogénéité de la partition.

La figure 3.10 retrace l'évolution de l'écart type des densités sur chacun des axes dans les configurations correspondant à une absence de pénalité ainsi qu'aux proportions d'impuretés tolérées de 10% et 5%, $p = 0.90$ et $p = 0.95$. La pénalité appliquée consiste à doubler la valeur du critère de

	Abscisse						Ordonnée					
	σ_d	H	Centres des SEF				σ_d	H	Centres des SEF			
p = 0	7.61	6.52	0.25	0.70			3.45	15.41	0.30	0.71		
	11.30	5.96	0.20	0.31	0.70		6.21	13.50	0.30	0.61	0.78	
	37.63	5.87	0.16	0.28	0.31	0.70	8.51	13.85	0.23	0.49	0.61	0.78
p = 0.90	10.58	7.52	0.36	0.87			4.71	16.74	0.23	0.73		
	5.55	6.41	0.22	0.60	0.87		5.80	15.68	0.23	0.64	0.81	
	6.75	5.17	0.22	0.47	0.70	0.87	12.68	14.61	0.23	0.57	0.72	0.81
p =0.95	13.37	6.39	0.24	0.60			3.99	15.36	0.29	0.71		
	7.73	5.33	0.24	0.38	0.71		5.73	13.29	0.29	0.59	0.77	
	27.20	4.88	0.20	0.31	0.38	0.71	10.18	13.27	0.23	0.46	0.59	0.77

TAB. 3.1 – Évolution des centres des SEF et des indicateurs pour $Test2D$ en fonction des paramètres de pénalité

fusion (équation 3.2), soit un taux de pénalité, t, de 100% dans les deux cas. Ces trois illustrations conduisent à la même conclusion : les courbes présentent une décroissance rapide dans la zone 2 à 4 sous-ensembles flous. Les configurations pénalisées présentent toutes deux un minimum pour trois sous-ensembles flous sur l'axe des abscisses et deux sur l'axe des ordonnées. Ces résultats sont compatibles avec la structure des données illustrée par la figure 3.8. Considérons par exemple les centres issus de la procédure correspondant à la valeur $p = 0.90$. Les quatre valeurs pour l'axe des abscisses sont situées dans chacune des quatre zones représentées. Le passage de quatre à trois sous-ensembles flous, qui réalise la fusion des deux zones les plus petites, revient à accepter que la partie droite de la partition soit hétérogène du point de vue de la sortie et constitue une proposition acceptable. La même analyse peut être faite pour le découpage en trois et deux sous-ensembles flous de l'axe des ordonnées.

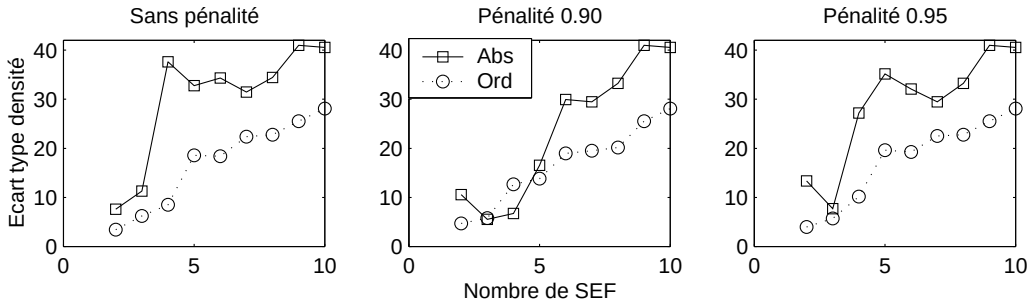


FIG. 3.10 – Évolution de l'écart type des densités sur $Test2D$ en fonction de la pénalité

3.5 Mise en œuvre : sensibilité aux paramètres

La méthode a été développée avec de nombreux paramètres. Malgré la présence d'interactions entre ceux-ci, il est nécessaire pour évaluer leur influence de les étudier séparément. La présente étude concerne la combinaison complète de la distance, définie dans l'équation 3.7 pour une partition floue forte.

Les principes exposés dans les sections précédentes doivent faire l'objet d'une validation sur

différents jeux de données. Nous avons choisi de travailler avec un jeu de données réelles, les iris de Fisher, et d'autres simulés pour mettre en évidence les caractéristiques ou propriétés de la méthode.

Afin d'éviter l'apparition de phénomènes indésirables aux extrémités de la partition, il convient d'une part de définir avec soin la forme des sous-ensembles flous extrêmes, et, d'autre part, de déterminer quels sous-ensembles flous doivent être pris en compte pour le calcul de la somme des distances. Ce sera l'objet des sections 3.5.2 et 3.5.3. Ensuite seront examinés l'effet du type de distance choisi, numérique ou symbolique, la sensibilité au regroupement en valeurs uniques et, enfin, l'influence des paramètres de pénalité. Mais commençons par présenter les principaux jeux de données utilisés.

3.5.1 Présentation des jeux de données

Le premier jeu de données qui sera utilisé est appelé *uniforme*. Il s'agit tout simplement d'un ensemble de 10 valeurs équidistantes dans l'espace euclidien (0, 1, ..., 9). L'initialisation produit donc 10 sous-ensembles flous (SEF) de masse unité, qui jouent un rôle équivalent dans la partition.

Le jeu de données *Gauss*3 est formé de trois générations aléatoires normales de cinquante valeurs chacune. Les caractéristiques observées sur les échantillons sont les suivantes : 0.10, 0.41, 0.87 pour les moyennes et environ 0.045 pour l'écart type. L'histogramme de la distribution est donné sur la figure 3.11.

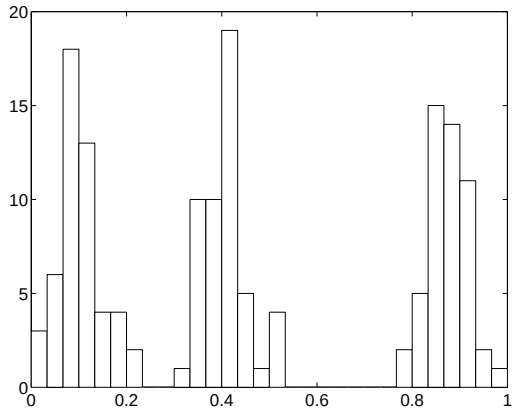


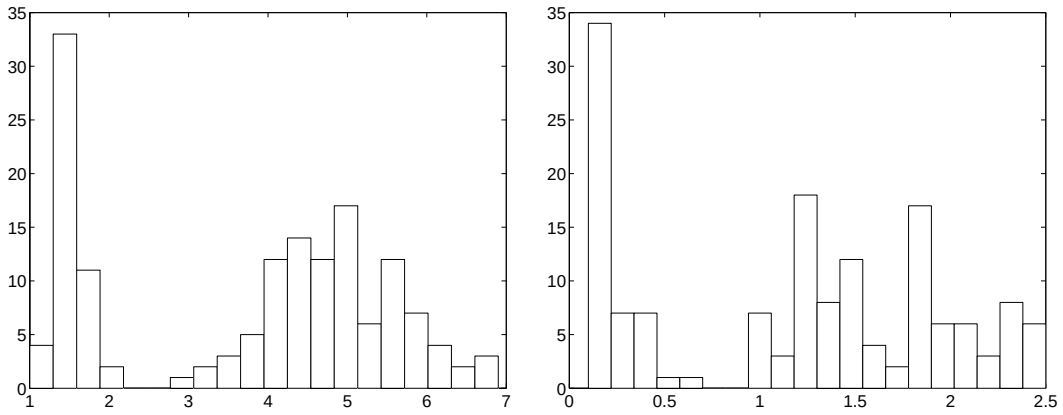
FIG. 3.11 – Histogramme du jeu de données *Gauss*3

Pour les iris il s'agit de discriminer 3 classes : Setosa, Versicolor et Virginica, les individus étant décrits par 4 variables : longueur et largeur des sépales, longueur et largeur des pétales. Les histogrammes de des variables relatives aux pétales sont représentés sur la figure 3.12.

3.5.2 Initialisation de la partition

Du point de vue de la sémantique, les sous-ensembles flous extrêmes devraient être modélisés par des demi-trapèzes, signifiant que les valeurs hors domaine appartiennent exclusivement, et avec un degré d'appartenance maximum, à l'un de ces ensembles. Ceci garantit également la propriété de partition floue forte. Cette solution sera adoptée pour l'utilisation des partitions.

Il ne peut pas en être de même pour la phase de construction car le processus de fusion donnerait incontestablement plus de poids au résultat de la fusion d'ensembles extrêmes, induisant ainsi une inégalité de traitement. Sur la figure 3.13 la partie gauche illustre la fusion des ensembles A_2 et A_3, la partie droite celle de A_1 et A_2.



(a) Longueur (b) Largeur

FIG. 3.12 – Histogrammes des caractéristiques des pétales des *iris*

C'est la raison pour laquelle nous préférons initialiser les ensembles extrêmes comme des triangles isocèles. Considérons s ensembles flous, numérotés de 1 à s, et c l'ensemble des coordonnées de leurs centres $(c_1,\ \ldots,\ c_s)$. La coordonnée du pied gauche du premier ensemble est alors : $2 * c_1 - c_2$, de même celle du pied droit de l'ensemble s est $2 * c_s - c_{s-1}$. La figure 3.14 illustre cette implémentation pour 5 sous-ensembles flous. Cette symétrie a un prix : la propriété de partition floue forte n'est maintenue que dans l'intervalle $[c_1,\ c_s]$.

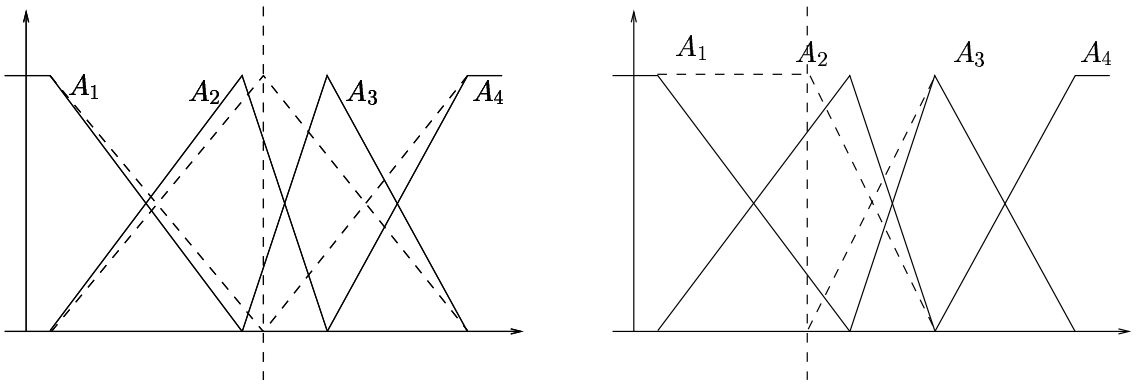


FIG. 3.13 – Fusion : les ensembles extrêmes sont des demi-trapèzes

Si le pré-traitement consiste seulement en un regroupement en valeurs uniques, il n'y a aucune valeur hors de cet intervalle lors de l'initialisation. Au cours du processus de fusion la valeur de c_1 (respectivement c_s) ne peut qu'augmenter (respectivement diminuer) faisant ainsi apparaître des valeurs pour lesquelles la somme des coefficients d'appartenance sera inférieure à l'unité.

Avant d'étudier la sensibilité de la méthode aux principaux paramètres, nous allons nous assurer que le critère de fusion est indépendant de la position des sous-ensembles flous fusionnés dans la partition, toutes choses étant égales par ailleurs. Nous allons montrer que cette indépendance ne peut être approchée, pour la distance symbolique, qu'en réduisant le nombre de sous-ensembles flous pris en compte dans le calcul de la somme des distances.

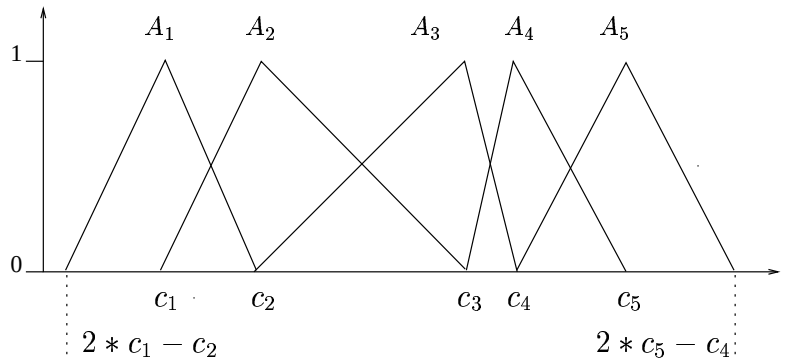


FIG. 3.14 – Fusion : les ensembles extrêmes sont des triangles isocèles

3.5.3 Nombre de sous-ensembles flous pris en compte

Le jeu de données *uniforme* a été créé pour cette étude. Comme il comprend 10 valeurs, il y a 9 combinaisons de fusion à étudier lors de la première itération. Nous voulons que, lors de la première itération, la valeur du critère de fusion soit la même pour les neuf combinaisons possibles. Chacune de ces combinaisons est étiquetée par la valeur des centres des sous-ensembles flous fusionnés. Ainsi la colonne $0-1$ correspond à la fusion des ensembles centrés sur les valeurs 0 et 1. Le nombre 1.138 est la valeur du critère de fusion associée à cette combinaison lorsque la distance est numérique. Rappelons qu'il s'agit du rapport de la somme des distances. Ce rapport n'est pas nécessairement inférieur à l'unité car le nombre de sous-ensembles flous, qui intervient dans le calcul des distances, varie : il y en a un de moins après la fusion qu'avant. La somme des distances, définie dans l'équation 3.1, est divisée par le nombre de paires de points intervenant dans la sommation, soit $n \times n - 1$, n étant le nombre de points, 10 pour la distribution *Uniforme*.

Le tableau 3.2 montre que, quelle que soit la distance utilisée, les sous-ensembles flous extrêmes sont favorisés. Ceci provient du fait que la propriété de partition floue forte n'est pas maintenue pour les bords : la somme des degrés d'appartenance est inférieure à l'unité pour ces points.

SEF fusionnés	0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9
Distance numérique									
Tous les SEF	1.138	1.172	1.172	1.172	1.172	1.172	1.172	1.172	1.138
4 SEF	1.254	1.342	1.342	1.342	1.342	1.342	1.342	1.342	1.254
Distance symbolique									
Tous les SEF	1.164	1.173	1.158	1.149	1.146	1.149	1.158	1.173	1.164
4 SEF	1.216	1.266	1.266	1.266	1.266	1.266	1.266	1.266	1.216

TAB. 3.2 – Distribution des critères de fusion du jeu *Uniforme* en fonction du nombre de SEF pris en compte dans le calcul de la somme des distances

La parabole obtenue lorsqu'on utilise la distance symbolique s'explique par le fait que la distance symbolique introduit une discontinuité dans la distance entre prototypes. Ces discontinuités sont amplifiées dans le calcul de la somme des distances car la contribution des points est directement fonction de leur position dans la partition.

Pour limiter cet effet nous pouvons envisager de travailler avec les seuls ensembles modifiés. Sur la figure 3.7, les ensembles pris en compte pour calculer le critère correspondant à la fusion de A_2

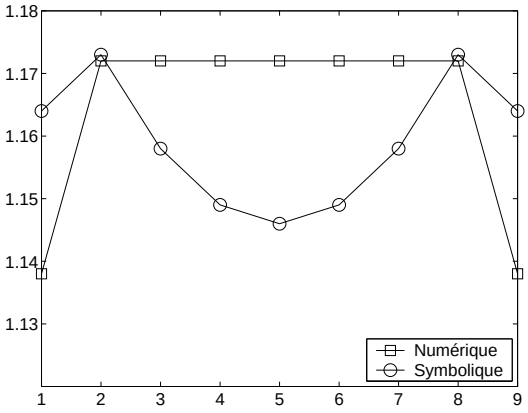


FIG. 3.15 – Distribution des critères de fusion du jeu $Uniforme$ en fonction du type de distance

et A_3 sont A_1, A_2, A_3 et A_4 pour la somme des distances *avant* la fusion ; A_1, A_{23} (le résultat de la fusion) et A_4 pour la somme des distances *aprs* la fusion.

Ce principe, s'il était appliqué aux bords, conduirait encore à des inégalités. En effet le nombre de sous-ensembles flous à prendre en compte, car modifiés par la fusion, n'est plus que de trois avant la fusion et deux après. Il n'y a pas d'ensemble *prcdent* sur le bord gauche, ni d'ensemble *suivant* sur le bord droit. Sur la figure 3.7, pour la fusion de A_1 et A_2, les ensembles concernés sont A_1, A_2, A_3 pour *avant* et A_{12}, A_3 pour *aprs*. Pour rétablir l'équité, nous ajoutons un quatrième ensemble même s'il n'est pas modifié. Nous choisissons le plus proche de ceux qui sont modifiés, soit A_4 pour la fusion de A_1 et A_2.

Les paires de points qui participent au calcul de la somme des distances sont celles pour lesquels un degré d'appartenance de chacun des points est non nul pour l'un des 4 ensembles pris en compte.

Les résultats obtenus avec la distance symbolique sont maintenant analogues à ceux obtenus précédemment avec la distance numérique comme l'indiquent les deuxièmes lignes de chacune des sections du tableau 3.2. Nous pouvons néanmoins constater que le biais, dû à l'abandon de la propriété de partition floue forte, est plus important, pour la distance numérique, lorsque la somme des distances est calculée sur les seuls quatre sous-ensembles flous modifiés.

Ceci nous conduit à préférer calculer la somme des distances sur l'ensemble de la partition si la distance est numérique, et sur les seuls sous-ensembles flous modifiés lorsqu'elle est symbolique.

3.5.4 Type de distance : symbolique ou numérique

L'effet du type de distance sera validé sur les iris, largeur et longueur des pétales, ainsi que le jeu de donnée $Gauss3$. Tous deux sont présentés dans la section 3.5.1.

Pour étudier l'effet du type de distance, nous travaillons avec le nombre de sous-ensembles flous optimal pour la distance symbolique, soit quatre, et nous testons les deux combinaisons correspondant à la distance numérique. Les résultats sont comparés à ceux obtenus par une méthode de référence, les $k-means$, dite encore méthode des centres mobiles. Le nombre de groupes, qui est fixé *a priori*, est de trois pour $Gauss3$.

Distribution Gauss3			
Numérique Tous Sef	0.06	0.41	0.93
Numérique 4 Sef	0.06	0.40	0.91
Symbolique 4 Sef	0.18	0.61	0.89
k-means	0.10	0.41	0.87

TAB. 3.3 – Influence du type de distance sur la position des centres des SEF pour le jeu de données *Gauss*3

Le tableau 3.3 montre que les résultats obtenus par la distance numérique sont voisins pour les deux configurations étudiées, et proches de ceux résultant des $k-means$. Les centres issus de la procédure utilisant la distance symbolique sont différents.

Afin d'évaluer la stabilité de ces résultats nous avons fait varier le seuil de significativité des degrés d'appartenance. Les valeurs inférieures au seuil n'interviennent pas dans le calcul. Le tableau 3.3 a été obtenu avec un seuil de 10^{-6}, qui est noté 0 dans les tableaux suivants. Le tableau 3.4 montre que seule la distance numérique la plus globale, celle qui prend en compte l'ensemble de la partition, permet d'obtenir une stabilité relative. Le seuil de rupture, pour cet exemple, se situe entre 0.10 et 0.15.

Il est important de noter que nous cherchons à obtenir une partition efficace. Nous devons, bien sûr, examiner la position des centres, mais surtout, l'impact de leurs positions respectives sur le partitionnement. La figure 3.16 illustre cet effet. Les centres $c1$ et $c2$ définissent les limites des noyaux des sous-ensembles flous mais aussi le point de croisement, $p1$. Nous pouvons observer qu'une différence sur la position des centres peut ne pas influer sur la position de ce point de croisement, mais seulement sur la largeur de la zone incertaine : la position de $p1$ n'a pas évolué malgré les déplacements de $c1$ et $c2$.

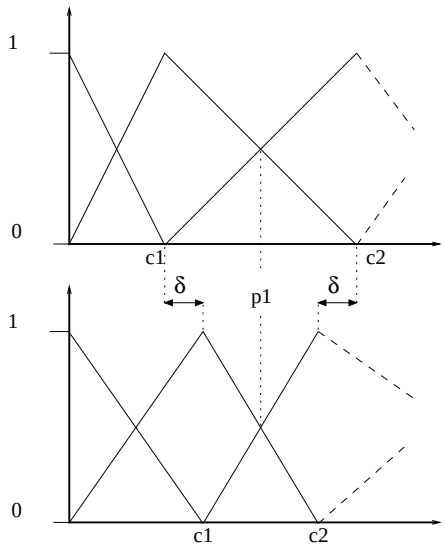


FIG. 3.16 – Impact de la position des centres sur le partitionnement

Les remarques précédentes doivent être vérifiées sur d'autres jeux de données, tels que les iris de Fisher. Nous avons choisi de travailler avec les variables relatives aux pétales car ce sont les

Distribution Gauss3				
	seuil	Centres des SEF		
Numérique Tous Sef	0.00	6.23e-02	4.14e-01	9.29e-01
Numérique Tous Sef	0.05	8.78e-02	4.42e-01	9.41e-01
Numérique Tous Sef	0.10	6.23e-02	3.89e-01	9.10e-01
Numérique Tous Sef	0.15	6.43e-02	2.99e-01	8.03e-01
Numérique Tous Sef	0.20	4.41e-02	3.81e-01	9.05e-01
Numérique Tous Sef	0.25	4.41e-02	4.12e-01	9.19e-01
Numérique Tous Sef	0.30	1.80e-02	3.88e-01	8.93e-01
Numérique Tous Sef	0.35	1.80e-02	3.92e-01	8.79e-01
Numérique Tous Sef	0.40	1.80e-02	3.29e-01	9.23e-01
Numérique Tous Sef	0.45	1.80e-02	3.92e-01	9.25e-01
Numérique Tous Sef	0.50	6.00e-02	3.67e-01	8.87e-01
	seuil	Centres des SEF		
Numérique 4 Sef	0.00	6.21e-02	4.01e-01	9.12e-01
Numérique 4 Sef	0.05	1.44e-01	5.68e-01	8.92e-01
Numérique 4 Sef	0.10	6.42e-02	4.02e-01	9.04e-01
Numérique 4 Sef	0.15	8.67e-02	2.88e-01	7.90e-01
Numérique 4 Sef	0.20	4.41e-02	4.95e-01	9.19e-01
Numérique 4 Sef	0.25	1.18e-01	5.77e-01	8.97e-01
Numérique 4 Sef	0.30	2.09e-02	3.23e-01	9.21e-01
Numérique 4 Sef	0.35	2.09e-02	3.61e-01	8.81e-01
Numérique 4 Sef	0.40	1.04e-01	6.17e-01	9.22e-01
Numérique 4 Sef	0.45	-4.12e-05	2.12e-01	9.04e-01
Numérique 4 Sef	0.50	-4.12e-05	3.24e-01	8.79e-01
	seuil	Centres des SEF		
Symbolique 4 Sef	0.00	1.81e-01	6.05e-01	8.92e-01
Symbolique 4 Sef	0.05	1.56e-01	6.11e-01	8.99e-01
Symbolique 4 Sef	0.10	8.41e-02	2.92e-01	8.09e-01
Symbolique 4 Sef	0.15	8.40e-02	2.94e-01	7.98e-01
Symbolique 4 Sef	0.20	4.41e-02	2.99e-01	7.88e-01
Symbolique 4 Sef	0.25	1.64e-01	5.46e-01	8.77e-01
Symbolique 4 Sef	0.30	1.38e-01	5.56e-01	8.91e-01
Symbolique 4 Sef	0.35	1.53e-01	5.84e-01	8.92e-01
Symbolique 4 Sef	0.40	1.50e-01	5.36e-01	8.91e-01
Symbolique 4 Sef	0.45	4.96e-02	3.85e-01	9.05e-01
Symbolique 4 Sef	0.50	5.43e-02	3.85e-01	8.75e-01

TAB. 3.4 – Stabilité des centres en fonction du type de distance et du seuil de significativité pour le jeu de données *Gauss*3

plus discriminantes. La figure 3.12 montre les histogrammes des distributions des deux variables. La largeur des pétales présente trois modes correspondant à chacune des trois classes, il y a très peu de recouvrement. La longueur des pétales est aussi intéressante car nous pouvons y voir deux modes, et ainsi tester la procédure jusqu'à son terme. La distance est numérique et la somme est calculée sur l'ensemble de la partition.

Seuil	Centres des SEF		
0.00	0.10	1.40	2.24
0.05	0.10	1.55	2.50
0.10	0.10	1.33	2.30
0.15	0.10	1.36	2.30
0.20	0.10	1.36	2.30
0.25	0.10	1.42	2.50
0.30	0.21	1.46	2.39
0.35	0.10	1.39	2.42
0.40	0.22	1.36	2.45
0.45	0.10	1.41	2.45
0.50	0.10	1.46	2.45
k-means	0.25	1.32	2.06

TAB. 3.5 – Stabilité des centres en fonction du seuil de significativité pour la largeur des pétales des iris avec la distance numérique appliquée à l'ensemble des sous-ensembles flous.

Le tableau 3.5 montre que les positions des centres de la largeur des pétales fluctuent dans des limites acceptables. Les résultats, répertoriés dans le tableau 3.6, sont différents pour la longueur des pétales. Nous avons choisi de présenter les partitions formées de trois sous-ensembles flous car les individus appartiennent à trois classes, et celles formées de deux sous-ensembles flous car la distribution fait apparaître deux modes. Deux lignes, seuils de 0.05 et de 0.10 se distinguent nettement des autres. Ces discontinuités dans la position des centres sont dues à des effets de seuil.

	Centres des SEF				
Seuil	3 valeurs			2 valeurs	
0.00	1.34	4.49	6.90	1.34	5.08
0.05	1.34	3.56	5.88	2.47	5.88
0.10	1.34	3.56	5.81	2.46	5.81
0.15	1.34	4.47	6.09	1.34	5.01
0.20	1.34	4.51	6.34	1.34	5.04
0.25	1.43	4.55	6.31	1.43	5.06
0.30	1.44	4.54	5.89	1.44	5.01
0.35	1.42	4.61	6.69	1.42	5.12
0.40	1.15	4.36	6.62	1.15	5.00
0.45	1.43	4.68	6.76	1.43	5.15
0.50	1.43	4.53	6.76	1.43	5.09
k-means	1.46	4.23	5.56	1.49	4.93

TAB. 3.6 – Stabilité des centres en fonction du seuil de significativité pour la longueur des pétales des iris avec la distance numérique.

Nous pouvons retenir de cette section que la combinaison de référence est l'utilisation de la

distance numérique calculée sur l'ensemble de la partition et prenant en compte l'ensemble des degrés d'appartenance. La distance symbolique ne semble pas concurrencer la distance numérique sur des données numériques. Elle pourrait, en revanche, se montrer plus performante sur des données de nature symbolique.

3.5.5 Sensibilité au nombre de valeurs uniques

La première étape de l'algorithme consiste en un regroupement en valeurs uniques. Deux paramètres peuvent influer sur la position des centres lors de ce regroupement. Le premier est le seuil de tolérance sur l'égalité de deux valeurs distinctes pour former une valeur unique. Le second est, pour un seuil de tolérance donné, la valeur initiale qui sert de référence pour comparer les autres valeurs.

Pour étudier leur influence respective, nous travaillons avec la distance numérique, la somme des distances étant calculée sur tous les sous-ensembles flous de la partition.

Choix de la valeur de référence pour le regroupement

Si toutes les valeurs distinctes sont utilisées comme valeurs uniques, la position des centres ne dépend pas de la valeur de départ. Les valeurs uniques résultantes étant, dans tous les cas, les mêmes.

En revanche, si la variable étudiée est réellement continue ou si la base d'apprentissage est très grande, il est probable que le nombre de valeurs uniques sera inférieur au nombre de valeurs distinctes. Dans ce cas le nombre de valeurs uniques peut varier en fonction de la valeur de départ.

Chacune des valeurs distinctes est utilisée comme valeur de référence pour le regroupement en valeurs uniques. Le tableau 3.7 rend compte des résultats pour la distribution *Gauss*3 qui comprend 43 valeurs distinctes. Lorsque la tolérance d'égalité est fixée à 1.4%, le nombre de valeurs uniques est soit 25, soit 26. La première colonne du tableau indique le nombre d'occurrences de la partition correspondante. Ce tableau indique qu'une position majoritaire, 26 occurrences, se dégage. La première et les trois dernières lignes correspondent à des partitions voisines, soit un total de 13 occurrences.

Nb Occur	Nb uniques	3 SEF		
1	26	0.08	0.44	0.94
26	25	0.15	0.61	0.91
4	26	0.01	0.53	0.91
6	26	0.06	0.42	0.91
1	26	0.06	0.41	0.91
5	26	0.06	0.42	0.93

TAB. 3.7 – Influence de l'index de départ pour le regroupement en valeurs uniques pour Gauss3 (Tolérance d'égalité 1.4%)

Le phénomène observé sur *Gauss*3 se retrouve sur la longueur des pétales des iris. Seules 4 partitions sur 43, soit le nombre de valeurs distinctes, diffèrent de la partition majoritaire comme le montre le tableau 3.8 obtenu avec une tolérance de 2%. Elles sont décrites dans les deux dernières lignes du tableau, les quatre premières conduisant à des partitions quasiment identiques.

Nb Occur	Nb uniques	3 SEF			2 SEF	
1.00	24	1.25	4.42	5.97	1.25	4.96
21.00	24	1.26	4.42	5.94	1.26	4.97
16.00	24	1.26	4.42	5.97	1.26	4.96
1.00	25	1.26	4.42	5.96	1.26	4.96
3.00	25	1.38	3.79	5.86	2.61	5.86
1.00	25	1.45	4.64	6.20	1.45	5.09

TAB. 3.8 – Influence de l'index de départ pour le regroupement en valeurs uniques pour la longueur des pétales des iris

Il est raisonnable de conclure que quelques essais, avec des indices de départ choisis aléatoirement, permettront d'identifier la partition majoritaire.

Du fait du caractère continu des données générées, les résultats de ce test pour chacune des variables du jeu de données $Test2D$ comporte autant de lignes que d'exemples, soit 150. Le nombre de valeurs uniques est de 55 ou 56 pour l'axe des abscisses et de 60 ou 61 pour l'axe des ordonnées. Nous proposons une visualisation statistique des résultats.

La figure 3.17 montre la répartition des individus dans l'espace et, pour chacun des axes, propose une illustration de la variation de la position des centres pour des partitions de 2, 3 et 4 sous-ensembles flous. L'outil de représentation choisi est appelé *boxplot*. Il fournit une représentation synthétique de l'histogramme cumulé de la distribution : les limites de la boîte sont les quartiles de la distribution, la marque intérieure est la médiane, les tirets délimitent les bornes minimum et maximum, tandis que les valeurs considérées comme éloignées de l'ensemble, les *outliers*, sont représentées par le signe $'+'$.

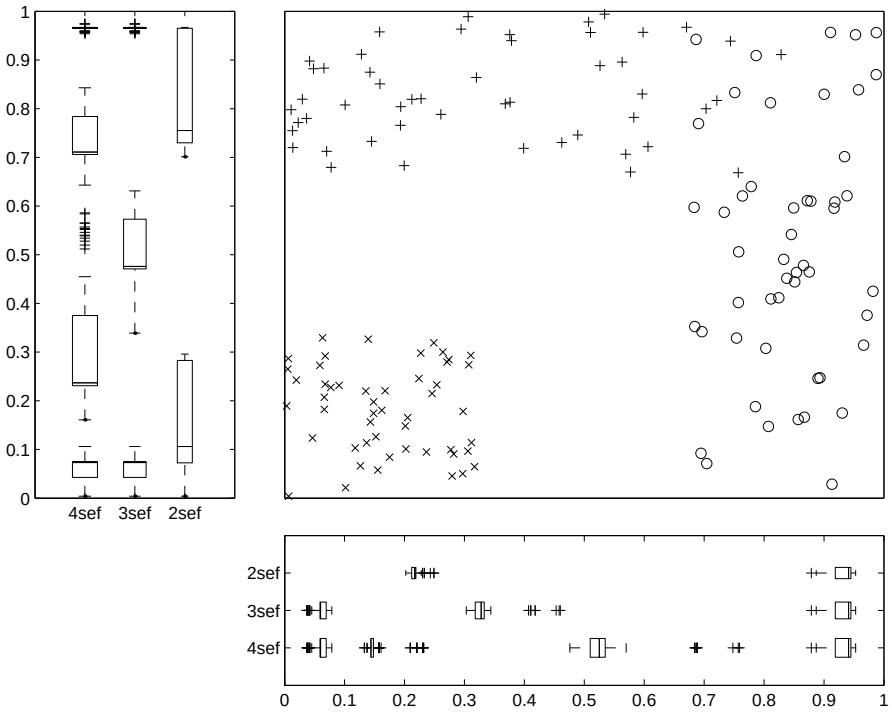


FIG. 3.17 – Influence de l'index de départ de regroupement en valeurs uniques pour Test2D

La figure 3.17 montre que les variations, bien qu'importantes, correspondent à des partitions interprétables.

Effet du seuil de tolérance

Afin de mettre en évidence l'effet du seuil de tolérance, la tolérance initiale est choisie suffisamment petite pour que le nombre de valeurs uniques soit égal à celui des valeurs distinctes. Puis, cette tolérance est augmentée, par pas de 0.001, et la procédure est conduite à son terme si le nombre de valeurs uniques est différent de celui obtenu avec la tolérance précédente.

Le tableau 3.9 illustre le fait que les résultats sont influencés par le nombre de valeurs uniques.

Seuil	Nb uniques	3 SEF		
0.001	43	6.23e-02	4.14e-01	9.29e-01
0.009	36	3.56e-02	5.58e-01	9.44e-01
0.010	34	5.54e-02	4.10e-01	9.32e-01
0.011	31	6.42e-02	4.12e-01	9.28e-01
0.012	30	6.42e-02	3.88e-01	9.19e-01
0.013	26	7.56e-02	4.35e-01	9.41e-01
0.014	25	1.49e-01	6.14e-01	9.10e-01
0.015	24	1.48e-01	6.15e-01	9.10e-01
0.019	22	7.09e-02	4.25e-01	9.29e-01
0.021	20	5.08e-03	5.48e-01	9.39e-01
0.026	18	5.08e-03	3.40e-01	8.77e-01
0.027	17	9.30e-02	4.71e-01	9.39e-01
0.029	16	5.27e-02	4.21e-01	9.18e-01
0.032	15	5.27e-02	4.23e-01	9.18e-01
0.042	13	7.36e-02	4.11e-01	8.98e-01
0.043	12	1.20e-01	5.26e-01	8.98e-01
0.056	11	8.85e-02	4.29e-01	9.27e-01
0.061	10	8.83e-02	4.16e-01	9.35e-01
0.064	9	8.83e-02	4.27e-01	9.59e-01
0.090	8	6.89e-02	2.74e-01	7.97e-01
0.092	7	6.89e-02	4.55e-01	1.00e+00
0.099	6	6.89e-02	2.96e-01	8.21e-01
0.151	5	9.20e-02	4.52e-01	1.00e+00
0.189	4	1.04e-01	6.35e-01	1.00e+00
0.200	3	1.04e-01	4.07e-01	8.71e-01

TAB. 3.9 – Influence du nombre de valeurs uniques sur la position des centres des SEF pour le jeu de données *Gauss*3

Même si les résultats correspondant à l'ensemble des valeurs de tolérance possibles sont rapportés dans ce tableau, il convient de noter que le concept de valeur unique n'est défini et porteur de sens que dans certaines limites. Nous ne pouvons néanmoins que constater que les partitions résultantes sont différentes, même pour des valeurs de tolérance faibles.

Le même phénomène peut être constaté sur les iris, comme l'indiquent les tableaux 3.10 et 3.11. Dans le tableau 3.10, la position des centres est complétée par la position des points de croisement correspondants.

Seuil	Nb uniques	3 centres			2 points de c.	
0.001	22	0.10	1.40	2.24	0.75	1.82
0.042	11	0.19	1.51	2.45	0.85	1.98
0.084	8	0.21	1.46	2.50	0.84	1.98
0.125	6	0.23	1.44	2.34	0.84	1.89
0.167	5	0.24	1.36	2.27	0.80	1.82
0.209	4	0.25	1.54	2.34	0.90	1.94
0.292	3	0.25	1.34	2.07	0.80	1.71

TAB. 3.10 – Influence du nombre de valeurs uniques sur la position des centres des SEF pour la largeur des pétales des iris

Seuil	Nb uniques	3 SEF			2 SEF	
0.001	43	1.34	4.49	6.90	1.34	5.08
0.017	24	1.26	4.42	5.94	1.26	4.97
0.034	18	1.40	4.59	6.32	1.40	5.07
0.051	13	1.45	4.54	6.26	1.45	5.05
0.068	10	1.33	4.53	6.05	3.07	6.05
0.085	9	1.39	3.64	5.49	2.38	5.49
0.102	8	1.42	4.50	6.13	1.42	5.04
0.119	7	1.44	3.90	5.58	2.57	5.58
0.153	5	1.46	4.47	6.35	1.46	5.02
0.221	4	1.46	4.66	6.19	1.46	5.13
0.323	3	1.46	4.29	5.63	1.46	4.91
0.492	2				1.84	5.07

TAB. 3.11 – Influence du nombre de valeurs uniques sur la position des centres des SEF pour la longueur des pétales des iris

La figure 3.18 illustre graphiquement le phénomène pour le jeu de données $Test2D$. Le nombre de valeurs uniques varie de 131 à 6 en 44 pas pour l'axe des abscisses, et de 133 à 7 en 47 pas pour l'axe des ordonnées. Bien que le type de représentation choisi, le *boxplot*, soit le même que pour rendre compte de l'influence du choix de la valeur initiale du regroupement en valeurs uniques, figure 3.17 , il est important de noter que, dans le cas présent, les valeurs de la distribution ne représentent pas la même expérience.

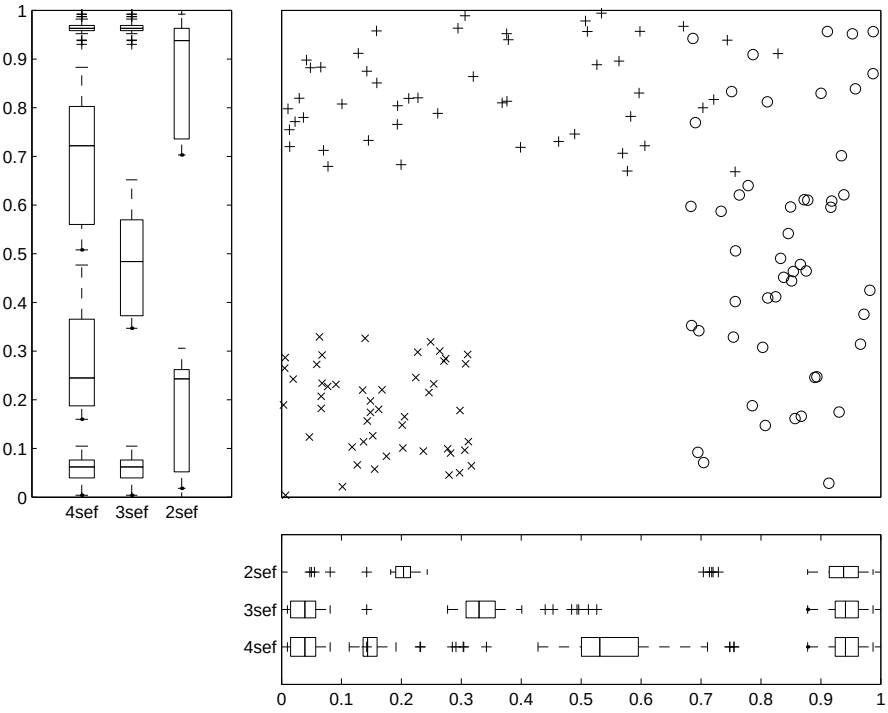


FIG. 3.18 – Influence de la tolérance sur l'égalité des valeurs uniques pour Test2D

Le choix de la tolérance d'égalité pour le regroupement en valeurs uniques a deux conséquences importantes. D'une part, il définit la partition qui sert de base à la procédure de fusion. D'autre part, pour le calcul de la somme des distances, les valeurs réelles sont assimilées aux valeurs uniques. Ces deux propositions concourent à améliorer la vitesse d'exécution de l'algorithme. Afin d'évaluer la part respective de ces conséquences sur les résultats, pour chacune des valeurs de tolérance seuil, la partition est initialisée avec les valeurs uniques comme précédemment, mais la somme des distances est calculée en utilisant l'ensemble des paires de points. Les résultats pour le jeu de données $Gauss3$ complets sont rapportés dans le tableau 3.12.

Leur représentation graphique est donnée sur la figure 3.19. La ligne P correspond aux partitions issues d'une somme des distances calculées sur l'ensemble des points, tandis que le ligne U illustre celles issues d'une somme des distances calculées sur les seules valeurs uniques. La largeur des boîtes indique que, lorsque la distance est calculée sur l'ensemble des points, la dispersion des centres résultants est moindre.

Les différences observées entre les premières lignes des tableaux 3.9 et 3.12, lorsque le nombre de valeurs uniques est identique au nombre de valeurs distinctes, sont imputables à la précision numérique de la représentation des nombres en machine et aux erreurs d'arrondi qui en découlent[f].

[f]Le nombre d'opérations dans la sommation est d'environ $\frac{n^2}{2}$. n représente dans un cas le cardinal de l'ensemble

Seuil	Nb uniques	3 SEF		
0.001	43	6.23e-02	4.07e-01	9.28e-01
0.009	36	4.41e-02	4.05e-01	9.25e-01
0.010	34	4.41e-02	4.26e-01	9.27e-01
0.011	31	1.27e-01	4.95e-01	9.35e-01
0.012	30	3.44e-02	3.97e-01	9.35e-01
0.013	26	4.81e-02	3.99e-01	9.28e-01
0.014	25	5.60e-02	4.02e-01	9.28e-01
0.015	24	5.60e-02	4.03e-01	9.28e-01
0.019	22	5.75e-02	4.09e-01	9.24e-01
0.021	20	5.95e-02	4.11e-01	9.28e-01
0.026	18	6.72e-02	4.13e-01	9.16e-01
0.027	17	4.71e-02	3.11e-01	9.46e-01
0.029	16	4.71e-02	4.25e-01	9.31e-01
0.032	15	4.71e-02	4.01e-01	9.31e-01
0.042	13	6.98e-02	4.08e-01	9.17e-01
0.043	12	6.98e-02	4.12e-01	9.17e-01
0.056	11	3.64e-02	3.84e-01	9.31e-01
0.061	10	3.64e-02	3.07e-01	9.46e-01
0.064	9	3.64e-02	3.11e-01	9.59e-01
0.090	8	6.89e-02	4.11e-01	9.17e-01
0.092	7	6.89e-02	4.14e-01	9.17e-01
0.099	6	6.89e-02	2.96e-01	8.16e-01
0.151	5	9.20e-02	4.57e-01	1.00e+00
0.189	4	1.04e-01	6.35e-01	1.00e+00
0.200	3	1.04e-01	4.07e-01	8.71e-01

TAB. 3.12 – Influence du nombre de valeurs uniques sur la position des centres des SEF pour le jeu de données *Gauss*3 lorsque la somme des distances est calculée sur l'ensemble des paires de points

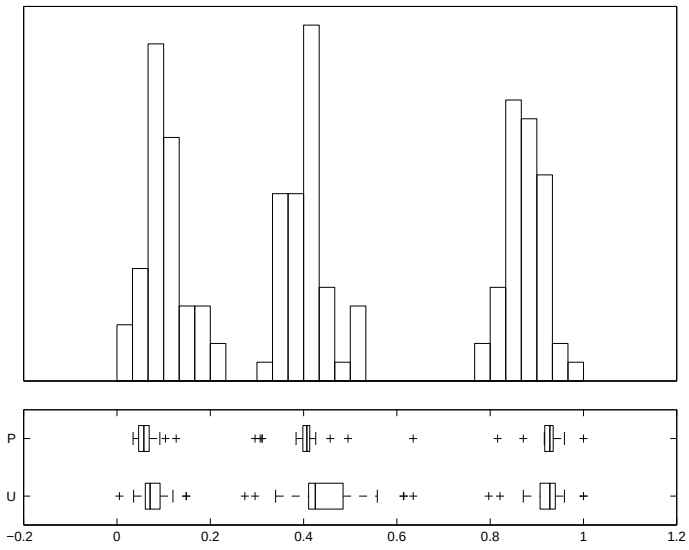


FIG. 3.19 – Influence de la tolérance sur l'égalité des valeurs uniques pour le jeu de données Gauss3

La même opération est réalisée pour la largeur des pétales des iris. Le nombre de valeurs est trop faible pour envisager une représentation graphique ou un traitement statistique. Le tableau 3.13 est à rapprocher de son homologue, le tableau 3.10, donnant la position des centres lorsque la somme des distances est calculée à partir des seules valeurs uniques. La similitude des coordonnées des centres correspondant aux valeurs de tolérance élevées se comprend bien : le petit nombre de sous-ensembles flous dans la partition initiale limite les possibilités de fusion, et conduit à des partitions très voisines.

Seuil	Nb uniques	3 SEF		
0.001	22	0.10	1.41	2.50
0.042	11	0.19	1.43	2.45
0.084	8	0.21	1.51	2.50
0.125	6	0.21	1.42	2.34
0.167	5	0.24	1.35	2.27
0.209	4	0.25	1.54	2.34
0.292	3	0.25	1.34	2.07

TAB. 3.13 – Influence du nombre de valeurs uniques sur la position des centres des SEF pour la largeur des pétales des iris lorsque la somme des distances est calculée sur l'ensemble des paires de points

Pour la longueur des pétales, les valeurs sont rapportées dans le tableau 3.14 et leur représentation graphique est donnée sur la figure 3.20.

Comme pour le jeu de données *Gauss*3, la principale source de la variation dans les résultats ne réside pas dans l'utilisation du regroupement en valeurs uniques pour former la partition initiale, mais dans le fait que le calcul de la somme des distances est réalisé à partir des valeurs uniques et non à partir des valeurs réelles de chacun des points. Ceci est particulièrement observable pour la configuration stable, celle comprenant deux sous-ensembles flous. La variabilité observée pour la

d'apprentissage, et, dans l'autre, le nombre de valeurs uniques.

Seuil	Nb uniques	3 SEF			2 SEF	
0.001	43	1.27	4.49	6.90	1.27	5.07
0.017	24	1.25	3.57	6.68	2.59	6.68
0.034	18	1.12	4.42	5.82	1.12	4.93
0.051	13	1.24	4.44	6.60	1.24	5.03
0.068	10	1.33	4.47	6.73	1.33	5.06
0.085	9	1.39	4.46	6.00	3.11	6.00
0.102	8	1.50	4.62	6.73	1.50	5.13
0.119	7	1.44	4.62	6.60	1.44	5.12
0.153	5	1.46	4.48	6.35	1.46	5.04
0.221	4	1.46	4.66	6.19	1.46	5.10
0.323	3	1.46	4.29	5.63	1.46	4.91
0.492					1.84	5.07

TAB. 3.14 – Influence du nombre de valeurs uniques sur la position des centres des SEF pour la longueur des pétales des iris lorsque la somme des distances est calculée sur l'ensemble des paires de points

configuration comportant trois sous-ensembles est davantage imputable à la structure des données qu'à l'un ou l'autre des paramètres de la méthode.

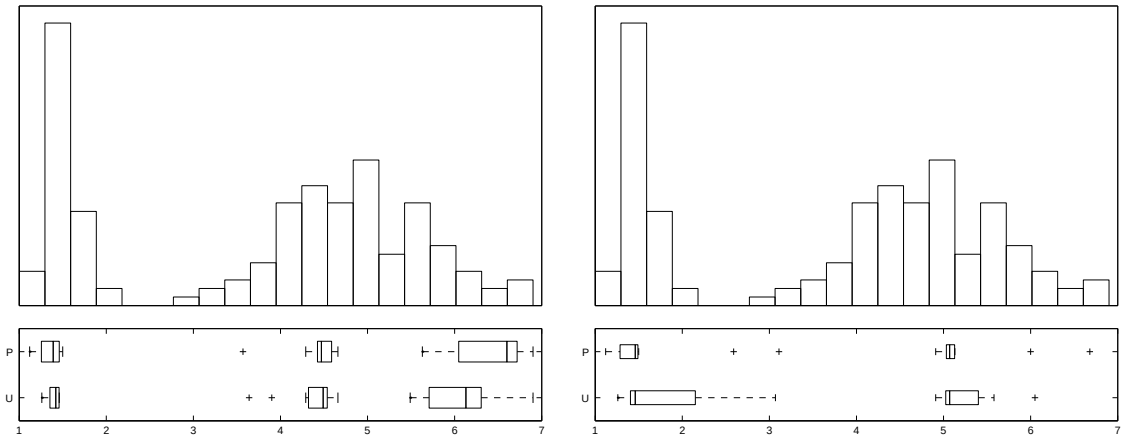


FIG. 3.20 – Influence de la tolérance sur l'égalité des valeurs uniques pour la longueur des pétales des iris

Les tests rapportés dans cette section montrent que la tolérance d'égalité pour former les valeurs uniques doit être remplacée par deux paramètres. Ces deux valeurs de tolérance d'égalité correspondent à deux fonctions de l'algorithme : le regroupement pour la formation de la partition initiale, pour la première, la transformation des valeurs réelles en valeurs uniques pour le calcul de la somme des distances, pour la seconde. La tolérance pour former la partition initiale peut être plus importante que celle qui est utilisée pour le calcul des distances. Si le nombre de valeurs distinctes n'est pas trop important, il est préférable de ne pas les regrouper. Dans le cas de données continues, la tolérance peut être compatible avec l'erreur de mesure ou bien correspondre à un niveau de résolution choisi. Si le nombre d'exemples d'apprentissage est élevé, le regroupement peut être réalisé par des méthodes statistiques, comme les $k - means$ par exemple.

3.5.6 Paramètres de pénalité

En examinant la distribution des valeurs de la longueur des pétales des iris en fonction des classes, figure 3.21, on remarque que le second mode provient du regroupement de deux classes. Il est intéressant d'observer le comportement de la procédure de fusion dans cette zone. Nous travaillons avec la distance numérique. La somme des distances est calculée sur la totalité de la partition. Toutes les valeurs distinctes sont utilisées, le seuil de significativité est fixé à 10^{-6}, l'hétérogénéité est calculée sur les dix derniers pas.

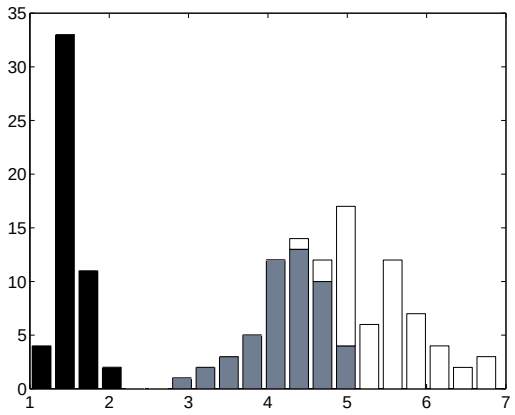


FIG. 3.21 – Histogramme de la longueur des pétales des *iris* en fonction des classes

Nous allons étudier principalement l'influence du paramètre p, la proportion de population hétérogène tolérée. Dans le cas où une pénalité est appliquée, nous avons choisi de multiplier la valeur du critère de distance, par un coefficient paramétrable. La valeur de ce paramètre est ajoutée à une partie fixe égale à un, de façon à former un taux de pénalisation, t. La valeur choisie, un, revient à doubler le critère de distance. Ceci signifie que les combinaisons de fusion pénalisées seront retardées tant qu'il existera des combinaisons non pénalisées. Un taux de pénalité plus petit permettrait à des fusions pénalisées de rester compétitives dans certains contextes.

Le tableau 3.15 rassemble les résultats pour la longueur des pétales des iris. Nous pouvons constater que la position des centres varie en fonction de ce paramètre, pour des valeurs de p comprises entre 0.70 et 0.95. Nous avons vérifié que la position des centres est relativement stable pour des valeurs de p voisines. L'effet de la pénalité est particulièrement sensible sur la position du troisième centre : il est ramené de 6.90 à environ 5.90 ce qui diminue notablement la largeur de la bande d'incertitude.

L'évolution de l'hétérogénéité peut aussi être un élément d'appréciation de la qualité de la partition. Le tableau 3.16 contient les résultats des derniers pas de la procédure lorsqu'aucune pénalité n'est appliquée et lorsque la valeur du paramètre p est 0.90.

Les deux configurations donnent des résultats comparables. Le minimum pour le critère de l'homogénéité des densités est obtenu pour une partition de deux sous-ensembles flous. Il correspond également à une nette augmentation de l'hétérogénéité comme l'illustre la figure 3.22, pour la situation $p = 0.90$. Ces indications sont en parfaite conformité avec l'observation. Le premier critère indique que la distribution est formée de deux modes, tandis que le second signale que le second mode regroupe deux classes. Dans ce cas une partition de trois sous-ensembles flous représente un bon compromis.

Iris : longueur des pétales					
p	3 valeurs			2 valeurs	
0.55	1.34	4.49	6.90	1.34	5.08
0.60	1.34	4.49	6.90	1.34	5.08
0.65	1.34	4.49	6.90	1.34	5.08
0.70	1.48	4.41	5.76	1.48	4.95
0.75	1.48	4.41	5.76	1.48	4.95
0.80	1.45	4.30	5.76	2.97	5.76
0.85	1.45	4.27	5.49	1.45	4.84
0.90	1.45	4.43	5.95	1.45	4.98
0.95	1.45	4.43	5.95	1.45	4.98
1.00	1.34	4.49	6.90	1.34	5.08
k-means	1.46	4.23	5.56	1.49	4.93

TAB. 3.15 – Évolution des centres des SEF en fonction des paramètres de pénalité pour la longueur des pétales des iris

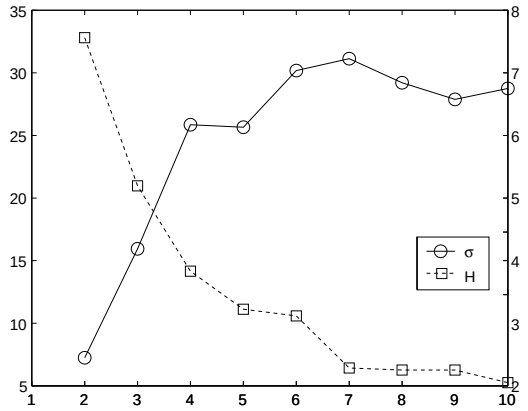


FIG. 3.22 – Évolution des indicateurs de qualité des partitions pour la longueur des pétales des iris.

σ_d	H	Centres des SEF					
		Sans pénalité					
5.90	8.395	1.34	5.08				
16.14	6.833	1.34	4.49	6.90			
21.92	5.265	1.34	2.98	5.04	6.90		
21.68	3.720	1.34	1.60	4.03	5.04	6.90	
24.85	2.816	1.34	1.60	4.03	4.68	5.39	6.90
		p = 0.90					
7.24	7.561	1.45	4.98				
15.95	5.195	1.45	4.43	5.95			
25.86	3.832	1.45	3.35	4.91	5.95		
25.66	3.224	1.45	1.70	4.15	4.91	5.95	
30.18	3.118	1.34	1.60	1.70	4.15	4.91	5.95

TAB. 3.16 – Évolution des centres des SEF et des indicateurs pour la longueur des pétales des iris

L'étude de la largeur des pétales, tableau 3.17, conduit à des différences entre les deux configurations. En l'absence de pénalité, les deux indicateurs montrent une préférence pour une partition de trois sous-ensembles flous. Ceci correspond à la fois au nombre d'espèces, donc de classes, et au nombre de modes que l'on peut discerner dans l'histogramme. Lorsqu'une pénalité est appliquée, le premier sous-ensemble flou est davantage centré sur le premier mode conduisant ainsi à une baisse de σ_d.

σ_d	H	Centres des SEF					
		Sans pénalité					
2.43	9.144	0.10	1.69				
2.16	6.457	0.10	1.40	2.24			
6.23	5.196	0.10	0.90	1.79	2.24		
5.34	3.340	0.10	0.32	1.36	1.79	2.24	
6.22	2.248	0.10	0.32	1.22	1.52	1.79	2.24
		p = 0.90					
1.01	7.746	0.22	1.67				
3.50	5.153	0.22	1.32	2.24			
6.87	3.779	0.22	0.95	1.48	2.24		
8.68	3.025	0.22	0.95	1.27	1.67	2.24	
8.07	2.771	0.10	0.32	0.95	1.27	1.67	2.24

TAB. 3.17 – Évolution des centres des SEF et des indicateurs pour la largeur des pétales des iris

L'interprétation du critère d'homogénéité des densités ne doit pas être trop stricte. Son objectif n'est pas de désigner la meilleure partition, mais d'attirer l'attention sur un ensemble de partitions possibles. Le choix final reviendra à l'utilisateur. Celui-ci pourra également s'appuyer sur l'évolution de l'hétérogénéité.

Il convient toutefois de prendre garde à ne pas surestimer l'indicateur basé sur cette évolution. Il est construit dans un espace monodimensionnel alors que nombre de problèmes sont multidimensionnels. Il peut être préférable de travailler, dans une dimension donnée, avec un sous-ensemble

flou fortement hétérogène en sachant que c'est la combinaison dans l'espace multidimensionnel qui est la clé de la solution.

3.6 Conclusion

La procédure de construction d'une hiérarchie de partitions validée dans ce chapitre présente l'avantage de la lisibilité. Les contraintes imposées, résumées dans le concept de partition floue forte, garantissent une interprétation linguistique des ensembles flous.

A chaque étape deux sous-ensembles flous voisins sont fusionnés suivant un critère : la minimisation de la variation de la somme des distances entre les points de l'ensemble d'apprentissage.

La distance, il faudrait écrire semi-distance pour être rigoureux, proposée, pour tenir compte du partitionnement, se compose de deux fonctions : la distance interne, entre points appartenant à un même ensemble flou, et la distance externe pour des points qui appartiennent à des ensembles différents. Elle peut être numérique, basée sur la distance euclidienne entre les noyaux des sous-ensembles flous, ou plus symbolique, chacun des sous-ensembles flous est alors identifié par son numéro d'ordre.

La méthode a été développée en utilisant de nombreux paramètres. Leur influence a été étudiée, et nous pouvons proposer des conditions d'usage standard, pour lesquelles les résultats sont stables. La référence est la distance numérique, la somme des distances étant calculée sur l'ensemble de la partition.

Afin de garantir une égalité de traitement pour tous les sous-ensembles flous de la partition, dans le cas de la distance dite symbolique, le nombre de sous-ensembles flous pris en compte pour la somme des distances doit être limité aux seuls quatre modifiés par la fusion. Le biais dû à l'abandon de la propriété de partition floue forte est plus important, pour la distance numérique, lorsque la somme des distances est calculée sur ces quatre ensembles seulement.

La distance numérique est préférable à la distance symbolique pour les données de nature numérique. La distance symbolique devra être testée sur des données adaptées, telles que des évaluations sensorielles.

La partition initiale résulte d'un regroupement en valeurs uniques. L'effet de cette tolérance sur la position des centres est limité si la somme des distances est calculée pour l'ensemble des paires de points. Assimiler les points à ces valeurs uniques permet d'accélérer notablement l'algorithme mais, en contrepartie, constitue une source d'instabilité. Dans le cas de données continues, la tolérance peut être compatible avec l'erreur de mesure ou bien correspondre à un niveau de résolution choisi. Lorsque le cardinal de l'ensemble d'apprentissage est trop important pour effectuer les calculs en considérant chacun des points, un regroupement pourra être réalisé par des méthodes statistiques, telles que les $k - means$ (Saporta, 1990).

Dans le cadre de l'apprentissage supervisé, il est possible de contraindre la procédure de fusion en pénalisant les combinaisons qui détruisent l'homogénéité de la partition du point de vue de la sortie. Ces éventuelles pénalités sont à appliquer sur les derniers pas de la fusion seulement.

L'écart type des densités des sous-ensembles flous de la partition peut être utilisé comme un indice de qualité. L'évolution de cette courbe au cours du processus de fusion passe par un minimum. Le nombre de sous-ensembles flous souhaitable est à rechercher autour de cette valeur. Nous pouvons également utiliser l'évolution de la courbe de l'hétérogénéité de la partition du point de vue de la

sortie pour qualifier les partitions. Il est préférable d'user de cette dernière information comme un complément de la précédente pour tenir compte du fait que la procédure a été appliquée sur une entrée isolée alors que le problème est, dans le cas général, multidimensionnel.

Chapitre 4

Induction de règles floues par composition multidimensionnelle

Trois points non alignés définissent non seulement un triangle, ce qui est est évident, mais un cercle, ce qui ne l'est pas.

Denis Guedj

(Guedj, 1998)

Sommaire

Rappelons que nous nous sommes donnés trois conditions à satisfaire pour qu'une base de règles induites soit interprétable :

1. Les sous-ensembles flous doivent s'interpréter comme des labels linguistiques ;
2. Le nombre de règles doit être petit ;
3. Les règles, notamment dans les systèmes de grande dimension, doivent être incomplètes, c'est-à-dire définies par les seules variables nécessaires. En effet, la présence systématique de l'ensemble des variables dans la définition des prémisses de chacune des règles, loin d'être une caractéristique du problème étudié, doit être considérée comme un inconvénient des méthodes d'induction.

La méthode d'induction de règles que nous proposons se réalise en trois étapes :

- La première étape définit les meilleures partitions dans chaque dimension. Elle a été présentée au chapitre précédent. Pour chacune des entrées le partitionnement ascendant hiérarchique, ou PAH, produit une famille de partitions de taille croissante, dont le nombre de sous-ensembles flous varie de deux à un maximum fixé, voisin de dix. Cette étape est une succession de phases monodimensionnelles.
- La deuxième étape, dite étape de sélection, consiste à générer des systèmes de règles multidimensionnelles en s'appuyant sur les familles de partitions, et à sélectionner l'un des plus performants parmi les plus compacts.
- La troisième étape, dite étape de simplification, vise à éliminer les règles et les variables les moins importantes et à obtenir des règles incomplètes.

La première étape de notre méthode d'induction garantit l'interprétabilité des sous-ensembles flous sur chacune des entrées et satisfait donc la première des conditions d'interprétabilité de la base de règles.

Les deux dernières étapes font l'objet du présent chapitre. L'induction de règles multidimensionnelles doit aussi satisfaire les deuxième et troisième conditions de lisibilité de la base de règles. L'objectif du nombre de règles petit n'est pas complètement atteint à la fin de la deuxième étape. La phase de simplification peut conduire à diminuer ce nombre.

L'objectif de la simplification de la base de règles est de réaliser un compromis entre la précision du système et son interprétabilité. Pour conduire le processus de simplification nous sommes amenés à accepter une dégradation contrôlée de l'indice de performance, l'erreur moyenne, ErM, présentée dans le glossaire de l'annexe A, et à proposer des indices qui dépassent la seule performance numérique. La lisibilité de la base de règles est évaluée à partir de critères complémentaires comme le nombre de règles du système ou le nombre de variables éliminées dans la prémisse des règles incomplètes.

Nous avons souligné l'importance de la sélection des variables pour la classe d'applications que nous souhaitons traiter. Pour la réaliser, au cours de l'étape de simplification, nous définissons un niveau intermédiaire entre le niveau global, la base de règles, et le niveau local, une règle isolée. Dans le premier cas, la variable éliminée n'est plus disponible pour aucune des règles, dans la second, la décision d'élimination est prise au seul niveau de la règle. Nous introduisons la notion de contexte pour définir l'impact d'une variable lorsque les autres sont fixées.

La section suivante présente l'étape de sélection, puis la section 4.2 introduit les différents niveaux de simplification de la base de règles. La section 4.3 détaille les différentes procédures de simplification et introduit les critères utilisés. Enfin, des exemples d'application sont traités en section 4.4.

4.1 Construire le système le plus performant parmi les plus compacts

Nous allons d'abord présenter la méthode de construction d'un système d'inférence floue que nous allons utiliser. Nous considérons un système de p entrées, chaque entrée comportant un nombre différent de sous-ensembles flous : $s_1, \ldots, s_p$, et d'une sortie qui peut être nette ou floue. Nous détaillerons ensuite l'étape de sélection du meilleur système.

4.1.1 Construction des systèmes d'inférence floue à l'aide du PAH

La construction du système d'inférence floue se réalise en deux temps : il s'agit d'abord de générer les prémisses de l'ensemble des règles possibles pour les partitions choisies sur les différentes entrées, soit $r_p = \prod_{i=1}^{p} s_i$ leur nombre ; puis de sélectionner les règles les plus influentes tout en initialisant leur conséquence, encore appelée conclusion.

Cette initialisation peut se réaliser de manière plus ou moins précise. Si l'objectif est la sélection des systèmes, une optimisation grossière des conclusions des règles permet de rejeter rapidement les moins adaptés. C'est la solution que nous adoptons. Dans une perspective d'utilisation, ou de caractérisation plus fine, l'initialisation pourrait être plus soignée.

La sortie observée dans la base d'apprentissage peut être soit une variable numérique continue, soit une classe. La conclusion des règles peut être soit nette, une valeur ou bien une classe, soit un ensemble flou. Considérons le cas des règles dont la conclusion est nette.

L'initialisation des conclusions se fait à partir des valeurs observées pour un ensemble d'exemples choisis, dont le choix est expliqué plus loin, E_r pour la règle r. Si l'on travaille en classification, la conclusion de la règle est simplement la classe majoritaire dans E_r.

Dans le cas de sortie continue, l'initialisation la plus simple consiste à calculer la conclusion comme la somme, pondérée par les degrés de vérité, des sorties observées dans E_r. Soit, en notant $\mu_r(x_i)$ le degré de vérité de l'exemple i pour la règle r, et y_i la sortie observée de l'exemple i :

$$C_r = \frac{\sum_{i \in E_r} \mu_r(x_i) * y_i}{\sum_{i \in E_r} \mu_r(x_i)} \tag{4.1}$$

L'initialisation d'une conclusion floue se fait en deux temps. Tout d'abord une sortie nette est calculée suivant l'équation 4.1, comme pour une sortie continue. Puis, la conclusion de la règle est l'ensemble flou pour lequel le degré d'appartenance de la sortie nette est maximum. Cette méthode suppose que le partitionnement de la sortie soit défini au préalable, soit par un expert, soit de manière automatique comme une grille régulière, ou bien par des méthodes telles que le partitionnement ascendant hiérarchique.

Revenons sur le choix de l'ensemble E_r. Il est formé d'exemples choisis en fonction de leur degré de vérité pour la règle, suivant deux stratégies possibles.

La première stratégie, dite *dcrmentale*, consiste à ne retenir que les exemples qui activent le plus la règle. Leur nombre minimum est fixé par le paramètre $EffectifMin$. Le degré de vérité seuil est d'abord fixé à son maximum, soit 1. Si le nombre d'exemples, dont le degré de vérité pour la règle est supérieur ou égal au seuil, est inférieur à $EffectifMin$, la valeur seuil du degré de

vérité est décrémentée par pas, 0.1 ou 0.05 par exemple. La procédure décrémentale s'arrête dès que le nombre d'exemples est suffisant, ou bien si le degré seuil atteint une valeur limite paramétrée, $MuMin$.

Cette stratégie privilégie les prototypes de la règle, dont la définition est donnée dans le glossaire de l'annexe A. Elle considère que la gestion des exemples qui ont un degré de vérité moindre sera assurée par interpolation au moment de l'inférence.

La seconde, dite *minimum*, consiste à considérer l'ensemble des exemples dont le degré de vérité pour la règle est supérieur à un seuil donné, $MuMin$. Cette stratégie, par l'utilisation d'un ensemble plus large, autorise l'emploi de procédures plus fines, telles qu'une descente de gradient (Glorennec, 1999). Dans ce cas, la valeur initiale peut être également déterminée suivant l'équation 4.1.

Quelle que soit la stratégie appliquée, si le cardinal de E_r est inférieur à $EffectifMin$, la règle correspondante n'est pas sélectionnée car son influence dans l'ensemble d'apprentissage est jugée insuffisante.

Le rôle des paramètres $EffectifMin$ et $MuMin$ est différent dans chacune des stratégies. La stratégie décrémentale est pilotée par le paramètre d'effectif, le degré de vérité minimum constituant une limite, tandis que la stratégie minimum est pilotée par le paramètre de degré de vérité, l'effectif minimum n'étant qu'une vérification.

Les effets de ces paramètres sont importants sur le nombre de règles retenues. Ils sont étudiés dans la section 4.4.

Il possible, qu'en fonction des paramètres choisis, il existe des exemples qui n'activent que faiblement les règles. Un exemple sera déclaré inactif si son degré de vérité, cumulé sur l'ensemble des règles, NbR, est inférieur au paramètre $MuMin$. Soit :

$$Exemple\ k\ inactif \iff \sum_{j=1}^{NbR} \mu_j(x_k) < MuMin \tag{4.2}$$

Le nombre d'exemples inactifs nous renseigne sur la part de l'ensemble d'apprentissage qui n'est pas gérée par le système. Il sera utilisé pour définir un indice de couverture comme nous le verrons dans la section 4.3.5. L'utilisateur peut décider de relâcher les contraintes sur les paramètres $EffectifMin$ et $MuMin$, pour diminuer le nombre des exemples inactifs.

4.1.2 Sélection du meilleur système

Rappelons que la performance numérique d'un système pour un ensemble d'apprentissage est mesurée par l'erreur moyenne, définie dans le glossaire de l'annexe A, lorsque la sortie est continue, par le nombre d'exemples mal classés en classification.

Les familles de partitions résultant du partitionnement ascendant hiérarchique, PAH, permettent de construire un grand nombre de systèmes d'inférence floue en faisant varier le nombre de sous-ensembles flous sur chacune des entrées.

Le principe de la composition multidimensionnelle réalisée à partir du PAH consiste à initialiser un système simple, comprenant deux sous-ensembles flous par entrée, et à le caractériser par son indice de performance. Puis à chaque étape, un sous-ensemble flou est rajouté sur l'une des entrées. L'entrée choisie est celle qui minimise l'erreur. Les coordonnées des centres des sous-ensembles flous sont celles résultant du partitionnement ascendant hiérarchique. L'arrêt est paramétrable. Il peut

être décidé après un nombre d'itérations fixé, ou bien lorsque l'indicateur d'erreur remonte. On obtient dans ce cas le système le plus performant parmi les plus compacts.

Pour améliorer l'interprétabilité, nous acceptons de dégrader la performance numérique en simplifiant la base de règles. Faut-il effectuer cette opération de simplification à chaque étape de la sélection, ou bien est-il possible de sélectionner le meilleur système puis de le simplifier ? Les étapes de sélection et de simplification sont-elles réellement dissociables ? Pour répondre, il est préférable de transformer l'interrogation. Si ces étapes sont imbriquées, la simplification s'effectue à chaque pas de l'étape de sélection, alors la question déterminante est la suivante : quelle doit être l'influence de la simplification sur le pas de sélection suivant ?

Soit nous considérons que cette influence est nulle, et dans ce cas, il est inutile de réaliser la simplification au cours de l'étape de sélection, soit nous nous appuyons, dans des conditions qui restent à déterminer, sur les résultats de la simplification pour engager le pas suivant de l'étape de sélection. Ces répercussions peuvent être de deux ordres : soit conserver une simplification pour la suite de la procédure, soit l'écarter définitivement. Il paraît très difficile de définir, compte tenu de l'état d'avancement actuel, des critères permettant de figer les résultats d'un pas. C'est la raison pour laquelle nous n'effectuerons la simplification que sur le système issu de l'étape de sélection.

4.2 Simplifier la base de règles

L'étude bibliographique, présentée au chapitre 2, montre que la sélection de variables s'effectue classiquement de manière globale, plus rarement au niveau d'une règle.

L'élimination globale, qui suppose qu'une variable est d'égale importance pour toutes les régions de l'espace d'entrée, n'est pas du tout adaptée aux systèmes de grande dimension. Au contraire, les variables influentes dépendent de l'état du système. La sélection règle par règle, à laquelle il est plus difficile d'attacher une sémantique, ne permet pas d'appréhender l'ensemble d'un contexte. C'est pourquoi nous introduisons un niveau de sélection intermédiaire en formalisant la notion de contexte.

Un contexte est défini par l'ensemble des règles, que nous appelons groupe, dont les prémisses sont identiques à l'exception du label d'un sous-ensemble flou d'une même variable, et d'un seul. Un groupe permet d'évaluer l'influence d'une variable particulière dans le contexte défini par les autres variables. Ce niveau d'intervention nous paraît pertinent. Il constitue un intermédiaire entre une sélection globale et une sélection règle par règle.

L'objectif de cette étape est d'éliminer les variables inutiles dans la définition des règles. Cette simplification est réalisée en trois phases, correspondant à des niveaux d'intervention de plus en plus précis.

4.2.1 Trois niveaux de simplification

Ces niveaux sont traités en séquence : ce n'est que lorsqu'il n'y a plus aucune simplification possible à un niveau que le niveau suivant est examiné.

Le premier niveau est celui du groupe de règles, si la variable correspondante n'a qu'une influence limitée dans le contexte du groupe, elle est éliminée dans chacune des règles du groupe. Les règles du groupe deviennent alors identiques et peuvent être fusionnées en une seule. L'identification des groupes est basée sur la notion de distance entre règles qui sera présentée dans la section 4.3.1. Le

critère de fusion des règles d'un groupe, qui vérifie l'homogénéité de la sortie des exemples attirés par le groupe, sera détaillé dans la section 4.3.3.

Le deuxième niveau vise à éliminer la redondance. Pour ce faire la présence de chacune des règles dans la base doit être justifiée. Une règle sera éliminée, si malgré son absence, la base de règles reste performante, et si le nombre d'exemples qui n'activent que faiblement les règles reste en deçà d'un seuil fixé. Les critères correspondants de performance et de couverture sont présentés respectivement dans les sections 4.3.4 et 4.3.5.

Enfin, le troisième niveau s'intéresse à chacune des variables des règles restantes. Chacune des variables est examinée indépendamment règle par règle. Une variable est éliminée de la prémisse d'une règle, si la sortie des exemples attirés par la règle demeure homogène malgré l'élargissement de l'espace d'entrée couvert. Le critère d'élimination d'une variable est celui utilisé pour la fusion des règles des groupes.

4.2.2 Maintenir certaines propriétés de la base de règles

Le système d'inférence floue résultant de l'étape précédente est un système formé de règles complètes, définies par le même nombre de variables. Au cours des différentes phases de la simplification, nous devons veiller à maintenir les propriétés de la base de règles.

Outre l'indice de performance qui constitue souvent le critère principal, voire exclusif, de l'optimisation, nous surveillerons l'évolution de deux autres propriétés requises pour une base de règles : la cohérence et la complétude. Ces propriétés sont définies dans l'introduction du chapitre 2. La propriété de continuité, bien qu'importante, n'est pas étudiée dans ce mémoire.

Pour tenir compte du contexte de l'apprentissage, la complétude est mesurée par le nombre d'exemples inactifs défini dans l'équation 4.2. Elle est d'autant meilleure que ce nombre est petit. La valeur de référence, celle du système issu de l'étape de sélection, est sous le contrôle de l'utilisateur.

Lors des phases de fusion des règles d'un groupe et d'élimination des variables, la complétude ne peut que s'améliorer car la partie de l'espace d'entrée couverte par les règles augmente au cours du processus. En revanche, la phase d'élimination des règles peut conduire à une détérioration de la complétude.

La cohérence est vérifiée au niveau de chacune des règles. Plusieurs auteurs proposent, notamment dans le cadre de l'affinement de partition, comme indiqué dans la section 2.1.2, des indicateurs qui caractérisent l'homogénéité de la sortie des exemples attirés par la règle. Nous utilisons un indice, décrit dans la section 4.3.3, qui appartient à cette famille et qui est proche de l'indice de controverse de (Rojas et al., 2000).

Une règle reflète des incohérences si la sortie observée pour les exemples activés par la règle est très hétérogène. La démarche de simplification proposée vise à élargir l'espace couvert par la règle. La détection d'incohérences se fait donc, lors des phases de fusion des règles au sein des groupes et d'élimination des variables, sur la base de la situation issue de l'étape précédente. Il s'agit de l'étape de sélection pour la phase de fusion des règles des groupes, et de la situation après l'élimination des règles pour la phase d'élimination des variables.

Les sections suivantes présentent les trois procédures de simplification de la base de règles : la fusion au sein des groupes, l'élimination des règles et l'élimination des variables dans les règles restantes.

4.3 Méthodes et outils pour la simplification

La fusion des règles d'un groupe constitue la première étape de la simplification de la base de règles d'un système d'inférence floue. Le principe consiste à supprimer la variable étudiée de la prémisse des règles du groupe si son influence dans le contexte du groupe est limitée. Ceci correspond simplement à fusionner les règles du groupe en une seule.

Pour préciser la notion de groupe de règles, nous devons définir une fonction de distance entre règles.

4.3.1 Distance entre règles

La distance entre deux règles peut être définie simplement comme le nombre de variables d'entrée dont le label est différent dans la prémisse des deux règles. Considérons, à titre d'exemple, ces deux règles d'un système de quatre variables d'entrée :

Règle 1 : SI Var_1 EST A_3 ET Var_2 EST B_2 ET Var_3 EST C_2 ET Var_4 EST D_3 ALORS ...
Règle 2 : SI Var_1 EST A_1 ET Var_2 EST B_4 ET Var_3 EST C_2 ET Var_4 EST D_2 ALORS ...

La *Règle 1* et la *Règle 2* peuvent être décrites par les séquences 3223 et 1422. Elles différent par trois chiffres : le premier, le second et le quatrième. Leur distance vaut donc trois.

En généralisant à un système à p entrées, la distance est donnée par la formule :

$$d_r = \sum_{i=1}^{p} (1)\{m_i \neq n_i\}$$

dans laquelle m_i et n_i représentent les labels de la variable i dans les règles à comparer et p le nombre de variables d'entrées.

Chacun des éléments de cette somme est appelé distance partielle.

Cette fonction de distance peut sembler sommaire car elle ne tient aucun compte de la valeur des labels des sous-ensembles flous. Si les sous-ensembles sont ordonnés, et correspondent par exemple à des labels *Petit*, *Moyen* et *Grand*, alors il serait plus juste de considérer que la distance entre les labels *Petit* et *Grand* est le double de celle entre deux labels adjacents. Nous pensons néanmoins que son caractère élémentaire est bien adapté à la simplification. Si la variable doit être éliminée, cela signifie que le label du sous-ensemble flou de cette variable est sans importance, et donc que ce label ne doit pas intervenir dans la distance partielle.

La distance partielle pour une variable présente dans une règle et absente dans l'autre est nulle par convention. Toutefois, le nombre de distances partielles constitue un élément de la distance qui sera utilisé pour l'identification des groupes. Notons également qu'une variable absente est un élément neutre pour le calcul du degré de vérité de la règle pour un exemple.

4.3.2 Fusion des règles d'un groupe

Formellement, un groupe de règles est formé de l'ensemble des règles dont la distance, entre toutes les paires possibles du groupe, est inférieure ou égale à l'unité.

Un groupe de règles est caractérisé par son effectif, le nombre de règles qui le composent, et par sa règle générique. C'est la règle qui remplacera l'ensemble des règles du groupe si la fusion est

décidée. Sa prémisse est la partie commune des prémisses des règles du groupe. Sa conclusion est optimisée comme celle de n'importe qu'elle autre règle, vote majoritaire en classification ou bien moyenne pondérée si la sortie est continue.

Pour un système complet de p variables, $s_1, \ldots, s_p$ étant le nombre de sous-ensembles flous sur chacune des entrées, le nombre de groupes est :

$$NbGr = \sum_{i=1}^{p} \left(\prod_{\substack{j=1 \\ j \neq i}}^{p} s_j \right)$$

Ce nombre peut être supérieur au nombre de règles. Considérons par exemple un système de 3 entrées, tel que $s_1 = 3$, $s_2 = 4$ et $s_3 = 2$. Le nombre de règles possibles est dans ce cas égal à 24, $3 \times 4 \times 2$, tandis que le nombre de groupes maximal est 26, $3 \times 4 + 3 \times 2 + 4 \times 2$.

Une règle peut appartenir à plusieurs groupes, soit p groupes pour un système complet de p entrées.

La procédure de fusion au sein des groupes est itérative. Le système initial est celui issu de l'étape de sélection. Ce système n'est pas *a priori* complet du fait de la sélection des règles les plus influentes, même si les règles peuvent être qualifiées de complètes car définies par les mêmes variables.

A chaque pas, les opérations suivantes sont réalisées :

- identification des groupes candidats à la fusion
- calcul de l'indice de fusion de chacun des groupes indépendamment
- fusion au sein des groupes qui satisfont le critère
- actualisation de la base de règles
- calcul de l'indice de performance du nouveau système

Lorsque un groupe satisfait le critère de fusion, la règle générique du groupe est ajoutée à la base de règles, et toutes les règles initiales qui forment le groupe sont supprimées.

Lorsqu'un groupe ne satisfait pas au critère, les règles initiales sont conservées dans la base.

Comme les règles peuvent appartenir à plusieurs groupes, la priorité est donnée à la suppression : si une règle appartient à au moins un groupe dont la fusion est acceptée, alors, quelle que soit la valeur du critère pour les autres groupes auxquels elle appartient, elle sera supprimée de la base car son espace d'entrée est inclus dans celui de la règle générique.

La nouvelle base de règles contient donc les règles génériques des groupes fusionnés, les règles candidates pour former un groupe qui n'a pas été fusionné et qui n'ont pas été éliminées par la constitution d'un autre groupe, et les règles isolées qui n'appartenaient à aucun groupe.

Une fois la base de règles actualisée, l'indice de performance permet de vérifier que l'ensemble des modifications, effectuées sur un critère local à un groupe de règles, est compatible avec les exigences de précision.

La même procédure peut alors être appliquée aux nouvelles règles. A partir de la deuxième itération, il faut tenir compte des variables absentes. Pour cette raison, l'identification des groupes se

fait en deux étapes. Les groupes sont d'abord formés des règles dont la distance est rigoureusement égale à l'unité, et ne contient aucune distance partielle correspondant à une variable présente dans une règle et absente dans l'autre. Si aucun groupe n'est formé, alors la contrainte sur la distance est relâchée. Les règles candidates sont celles pour lesquelles la distance est nulle, c'est-à-dire qui contiennent des distances partielles correspondant à des variables présentes dans une règle et absentes dans l'autre. Par exemple, si la variable absente est notée, 0, la distance entre les règles 1021 et 0321 est nulle.

Nous pouvons remarquer qu'une règle isolée à une étape, peut faire partie d'un groupe à une étape ultérieure.

Cette procédure est appliquée tant que l'indice de performance est satisfaisant et tant qu'il y a des groupes candidats.

Les trois indices employés lors de l'étape de simplification sont maintenant présentés.

4.3.3 Indice d'hétérogénéité

Cet indice est utilisé comme indice de fusion des règles d'un groupe, et comme indice d'élimination des variables dans une règle. Il se propose de caractériser l'hétérogénéité de la sortie des exemples attirés par une règle. Si l'hétérogénéité est faible cela signifie que l'élargissement de l'espace couvert par la règle correspondant est pertinent.

Deux cas sont à distinguer, soit la sortie est une variable numérique continue, soit il s'agit d'une classe.

1. En classification, les valeurs de sortie sont simplement les numéros des classes. Appelons $c_1, \ldots, c_g$ les différentes conclusions possibles au sein du groupe de règles, et considérons la cardinalité floue des ensembles d'exemples ayant la même conclusion. Soient $n_1, \ldots, n_g$ les cardinalités telles que $n_1 \geq n_2 \geq \ldots \geq n_g$. n_i est donc la somme des degrés de vérité des exemples qui activent une règle dont la conclusion est c_i.

   L'indice décrivant l'hétérogénéité des conclusions, h, est calculé comme :

$$h = \frac{\left(\sum_{i=1}^{g} n_i\right) - n_1}{\sum_{i=1}^{g} n_i} \tag{4.3}$$

   Dans cette expression, n_1 représente la cardinalité de la conclusion majoritaire. Cette expression varie de 0, lorsque les conclusions de tous les exemples attirés par la règle générique du groupe sont identiques, à un nombre légèrement supérieur à $\frac{g-1}{g}$, lorsque les g conclusions différentes ont des cardinalités voisines.

   La valeur h est donc proportionnelle à l'hétérogénéité des cardinalités des différentes conclusions du groupe. Si h est suffisamment petit, nous pouvons considérer les exemples qui n'activent pas la sortie majoritaire comme marginaux par rapport à la règle. Dans le cas contraire, si h est important, l'utilisation de la règle conduirait à des réductions abusives.

2. En cas de sortie continue, l'indice est une fonction de l'écart type. Notons σ l'écart type de la distribution des sorties observées dans l'ensemble d'apprentissage, et σ_r l'écart type des sorties observées dans l'ensemble E_r. L'expression de l'indice h est :

$$h = \frac{\sigma_r}{\sigma} \tag{4.4}$$

Il varie entre zéro, si l'écart type des sorties des exemples activés par la règle est nul, et un, s'il est égal à celui de la distribution toute entière.

Cet indice d'hétérogénéité pourrait également être utilisé comme un degré de confiance dans la conclusion de la règle correspondante.

4.3.4 Indice de performance

Contrairement à l'indice précédent, local à une règle, l'indice de performance caractérise la base de règles dans son ensemble. Il est utilisé à chacune des étapes de la simplification : fusion des règles d'un groupe, élimination des règles et élimination des variables.

Il est défini comme la variation relative de l'erreur.

Soient e_{ref} l'erreur du système de référence et e_{cour} celle du système courant, e s'exprime comme :

$$e = \frac{e_{cour} - e_{ref}}{e_{ref}} \tag{4.5}$$

La variation relative de l'erreur peut être positive ou négative. Positive si la performance est détériorée, négative si elle est améliorée. La borne inférieure de e est -1, ce qui correspond à une erreur nulle, $e_{cour} = 0$. Il n'y a pas de borne supérieure. Une valeur de $e = 1$ signifie que l'erreur a été doublée. Le système courant sera retenu si la variation relative de l'erreur associée est inférieure à un seuil donné.

4.3.5 Indice de couverture

Cet indice est un complément indispensable du précédent. En effet, il est toujours possible d'obtenir une erreur nulle. Il suffit pour cela de ne traiter aucun exemple ou simplement des prototypes des différentes règles.

Comme le précédent, il caractérise la base de règles dans son ensemble.

Parmi les expressions possibles, nous avons choisi comme indice de couverture le nombre d'exemples inactifs. Un exemple est déclaré inactif si son degré de vérité cumulé sur l'ensemble des règles est inférieur à un seuil donné, comme indiqué dans l'équation 4.2.

4.3.6 Élimination des règles et des variables inutiles

L'élimination des règles est une des composantes importantes de l'étape de simplification. La redondance peut produire des effets inattendus car il n'y a aucun contrôle réalisé au moment de l'inférence. Toutes les règles sont activables. Dans le cas ou plusieurs règles couvrent la même partie de l'espace d'entrée, leurs contributions seront ajoutées pour calculer la sortie d'un point appartenant à cette portion d'espace. Si ces conclusions sont contradictoires, elles peuvent se compenser, dans le cas contraire, leur influence sera plus importante que souhaité.

Pratiquement, une règle sera éliminée si, lorsqu'on la retire, la base de règles satisfait encore les indices de performance et de couverture.

La dernière phase consiste à simplifier chacune des règles restantes. Pour cela, toutes les variables sont étudiées au fur et à mesure. Une variable sera éliminée si la règle, dont l'espace d'entrée couvert est élargi, satisfait les trois indices présentés dans les sections précédentes. Il faut donc que la sortie des exemples attirés par la règle simplifiée reste homogène et que la base de règles demeure performante. L'indice de couverture est automatiquement satisfait car les règles simplifiées deviennent moins spécifiques.

Après simplification, il est possible que la règle soit identique à une autre de la base déjà simplifiée. Pour les raisons évoquées précédemment, elle sera éliminée.

Les résultats de la phase d'élimination dépendent de l'ordre d'examen des règles, et de l'ordre d'examen des variables dans la prémisse.

L'ensemble de la méthode, génération de la famille de partitions floues, sélection du meilleur système formé de règles définies par les mêmes variables et simplification de la base de règles, est maintenant appliqué à deux cas bien connus.

4.4 Validation

La procédure d'induction de règles, par composition multidimensionnelle à partir du PAH, est maintenant complètement définie. Nous proposons de la valider sur des exemples connus. Nous avons choisi deux applications largement étudiées dans la littérature qui représentent deux classes de problèmes. La première, les iris de Fisher, est un problème de classification avec des données de référence en statistiques. La seconde, est un problème de régression qui vise à expliquer la qualité globale d'un riz à partir d'appréciations sensorielles recodées.

Le tableau 4.1 présente les indicateurs utilisés pour caractériser les différents systèmes.

Indicateur	Signification
$Nb\ Sef$	nombre de sous-ensembles flous par variable
$Max\ R$	nombre de règles correspondant à l'ensemble des combinaisons possibles
$\#R$	nombre de règles dans le système
$M\ cl$	nombre d'individus mal classés
ErM	erreur moyenne (voir glossaire A)
$ErMax$	écart maximum entre les sorties inférée et observée sur l'ensemble des exemples
$\#I$	nombre d'exemples inactifs
$\#E$	nombre total de variables éliminées dans les règles
$Max\ E$	nombre maximal de variables éliminées dans une règle

TAB. 4.1 – Description des indicateurs qui qualifient les systèmes

Le nombre de sous-ensembles flous par variable est l'élément de base de la description du système. Le nombre maximal de règles en est directement déduit. Les systèmes sélectionnés et simplifiés sont caractérisés par leur performance et le nombre de règles. La performance est mesurée en nombre d'individus mal classés pour les problèmes de classification, par l'erreur moyenne et l'erreur maximale pour les problèmes de régression. Les systèmes simplifiés sont également caractérisés par le nombre de variables éliminées dans l'ensemble des règles et au maximum dans une règle. Le nombre de

variables éliminées est à rapprocher du nombre total de variables dans les règles, produit du nombre de règles par le nombre d'entrées. Le nombre maximal de variables éliminées est à comparer au nombre de variables présentes dans le système.

4.4.1 Les iris de Fisher

L'objectif de cette étude est de mettre en évidence des règles de classification de trois sortes d'iris, Setosa, Versicolor et Virginica, en fonction de quatre variables descriptives : la largeur et la longueur des pétales et des sépales. Les données ont été présentées dans la section 3.5.1.

Configurations étudiées avec toutes les variables

Les paramètres du PAH pour l'étude des iris sont les suivants : toutes les valeurs distinctes sont conservées pour former la partition initiale ; la distance est numérique car les variables sont elles mêmes numériques, et calculée sur la totalité des sous-ensembles flous ; il n'y a aucune fusion pénalisée.

La composition multidimensionnelle est également paramétrée. L'effectif pris en compte pour calculer la conclusion des règles tout au long de la procédure est choisi suivant la stratégie décrémentale. L'effectif utilisé pour le calcul de l'indice d'homogénéité des conclusions est choisi suivant la stratégie minimum. La proportion d'exemples inactifs maximum autorisée est fixée à dix pour cent.

La sensibilité à deux paramètres, $EffectifMin$ et $MuMin$, est évaluée dans ce cadre. Pour chacun, trois valeurs sont étudiées, 1, 3 et 5 pour l'effectif minimum, 0.3, 0.5 et 0.7 pour le degré d'appartenance minimum. Ce qui représente les neuf configurations décrites dans le tableau 4.2.

Nom	EffectifMin	MuMin	Proportion inactifs tolérée
C1	1	0.3	0.1
C2	1	0.5	1.0
C3	1	0.7	1.0
C4	3	0.3	0.1
C5	3	0.5	1.0
C6	3	0.7	1.0
C7	5	0.3	1.0
C8	5	0.5	1.0
C9	5	0.7	1.0

TAB. 4.2 – Les neuf configurations étudiées pour les iris

Pour les configurations dont le degré d'appartenance minimum est de 0.5 ou 0.7, la sélection s'arrête rapidement, du fait de la contrainte sur la proportion d'exemples inactifs, et conduit à une solution non satisfaisante. Nous avons décrit celle correspondant à la deuxième configuration, elle est notée $C2$ (ns) dans le tableau récapitulatif. La contrainte est relâchée pour obtenir un système satisfaisant du point de vue de l'indice de performance. Pour la configuration sept, cette même contrainte est également relâchée, sinon la procédure s'arrête après six itérations.

Les différentes étapes de la sélection, pour la première configuration, sont indiquées dans le tableau 4.3. Comme le minimum de l'erreur de la quatrième itération est supérieur à l'erreur de référence acquise à l'itération précédente, la procédure s'arrête.

	Nombre de sef par entrée				Mal classés	Inactifs	Minimum
Départ	2	2	2	2	46	0	
Itération 1	3	2	2	2	34	0	
	2	3	2	2	41	0	
	2	2	3	2	30	0	
	2	2	2	3	16	0	Min
Itération 2	3	2	2	3	37	0	
	2	3	2	3	16	0	
	2	2	3	3	10	0	Min
	2	2	2	4	21	0	
Itération 3	3	2	3	3	5	0	Min
	2	3	3	3	13	0	
	2	2	4	3	6	0	
	2	2	3	4	16	0	
Itération 4	4	2	3	3	12	3	
	3	3	3	3	11	1	Min
	3	2	4	3	26	0	
	3	2	3	4	15	0	

TAB. 4.3 – Les étapes de la sélection pour la première configuration

Afin de s'assurer que le premier minimum atteint n'est pas un minimum local, la procédure de sélection est effectuée sur un nombre fixe de neuf itérations. Les résultats sont rapportés dans le tableau 4.4.

		Départ	Itérations successives								
C1	Mal classés	46	16	10	5	11	11	15	11	10	11
	Blancs	0	0	0	0	1	2	5	4	4	6
C2	Mal classés	46	16	19	10	10	8	8	8	8	8
	Blancs	13	38	42	44	16	20	35	40	36	51
C3	Mal classés	46	16	20	14	14	8	18	37	33	43
	Blancs	51	68	63	82	87	100	106	105	110	117
C4	Mal classés	46	17	10	9	11	12	9	9	10	13
	Blancs	0	1	0	10	15	8	7	13	33	42
C5-C6-C8-C9	Mal classés	46	16	23	24	37	47	51	61	66	68
	Blancs	33	38	43	47	69	73	85	99	100	103
C7	Mal classés	46	17	13	14	15	17	23	19	25	30
	Blancs	0	1	6	10	10	11	29	20	34	32

TAB. 4.4 – Évolution de l'erreur et des exemples inactifs pour les neuf configurations

Les systèmes correspondants aux configurations cinq, six, huit et neuf sont identiques. Seuls le nombre d'exemples inactifs, car lié au paramètre $MuMin$, et le nombre de mal classés, car le calcul de la conclusion des règles dépend de l'effectif, diffèrent. Ceux indiqués dans le tableau 4.4 sont relatifs à la configuration cinq. Pour les autres ces nombres sont 68, 38, et 85 pour les inactifs ; 16, 16, et 23 pour les mal classés.

Parmi les neuf configurations testées, deux courbes d'évolution, celles de la deuxième et de la

troisième configurations, présentent un minimum local dès la première itération. Ceci nous conduit à préférer exécuter un nombre fixé d'itérations et à rechercher celle correspondant au minimum global, plutôt que d'arrêter la procédure au premier minimum rencontré.

L'étape de simplification est conduite avec des paramètres identiques pour toutes les configurations. La valeur seuil de l'hétérogénéité des conclusions est 0.25, celle de la dégradation des performances de 1.0, ce qui correspond à autoriser un nombre de mal classés double de celui de la situation initiale.

Pour chacune des configurations, les systèmes issus des étapes de sélection et de simplification sont décrits par les paramètres définis dans le tableau 4.1.

Les résultats sont résumés dans le tableau 4.5. Lorsque après la simplification le système le plus simple n'est pas le plus performant, les deux sont décrits avec les extensions *si* pour le plus simple, celui qui compte le plus de variables absentes, *sp* pour le plus performant.

	Système formé de règles complètes					Système après simplification				
	Nb Sef	Max R	#R	M cl	#I	#R	M cl	#I	#E	Max E
C1	3 2 3 3	54	11	5	0	3	16	0	9	3
C2 (ns)	2 2 3 2	24	6	25	13					
C2 (sp)	2 2 6 3	72	12	8	20	4	6	0	10	3
C2 (si)	2 2 6 3	72	12	8	20	4	7	0	12	3
C3	2 5 3 3	90	12	8	100	3	9	31	6	3
C4 (sp)	3 2 3 3	54	8	9	10	4	7	2	8	3
C4 (si)	3 2 3 3	54	8	9	10	4	9	1	9	3
C5-C6-C8-C9	2 2 2 3	24	5	16	38	3	8	1	6	3
C7	2 2 3 3	36	6	13	6	3	13	25	3	1

TAB. 4.5 – Résultats des neuf configurations testées sur les iris

Le nombre de règles du système sélectionné dépend étroitement du nombre d'exemples minimum pour initialiser la conclusion de la règle, $Effectif\ min.$ Les trois configurations pour lesquelles l'effectif minimum est seulement d'un exemple, les trois premières, présentent un nombre de règles significativement plus élevé que les autres.

Quelle que soit la configuration issue de la sélection, le système simplifié compte trois ou quatre règles. Le nombre d'exemples mal classés est voisin pour tout un ensemble de configurations. Remarquons tout de même que les configurations un et sept présentent des performances moins satisfaisantes. Pour les configurations deux, quatre et cinq, la performance en classification du système simplifié est meilleure que celle de système formé des règles complètes. Le nombre d'exemples inactifs pour les configurations trois et sept, est élevé. L'explication est double. D'une part, le mode de calcul, le seuil de référence étant de 0.7 pour ces configurations, tend à conduire à un nombre d'inactifs plus élevé. D'autre part, le fait que les règles de ces systèmes figurent parmi les moins simplifiées, six variables éliminées de la définition des prémisses seulement pour l'une et trois pour l'autre, restreint la partie de l'espace d'entrée couverte par les règles. Les systèmes les plus simples sont ceux issus des configurations un et deux.

L'analyse de ces résultats montre que ce sont les paramètres de la configuration numéro deux qui sont les plus adaptés à ce jeu de données. Bien que le système initial ne soit pas le plus performant, huit individus mal classés, ni très complet, vingt individus inactifs, la simplification conduit à des

résultats honorables : sept individus mal classés avec quatre règles définies chacune par une variable seulement et aucun exemple inactif.

Les règles correspondant à chacune des configurations sont données, sous forme de matrices symboliques, dans le tableau 4.6. La matrice de la première colonne contient les règles du système issu de la sélection, la matrice de la deuxième celles du système le plus performant issu de l'étape de simplification, et éventuellement, lorsque le système le plus simple est différent du plus performant, les règles correspondantes sont indiquées dans la dernière matrice. Les variables absentes sont mises à zéro.

Configurations étudiées avec les seules variables des pétales

Ces essais montrent que toutes les variables ne sont pas nécessaires à la définition du problème. La lisibilité des bases de règles simplifiées est notablement améliorée. Mais même si le nombre de variables éliminées, est, comme le nombre de règles dans la système simplifié, relativement homogène, cela ne signifie par pour autant que les règles résultantes dans les différentes configurations soient très proches. En effet, ce ne sont pas toujours les mêmes variables qui ont été sélectionnées. La redondance dans l'information autorise plusieurs choix dans la sélection.

La base de règles la plus simple est celle issue de la première configuration. Une seule variable, la largeur des pétales, est nécessaire pour la définir. Elle contient seulement une règle par classe. Sa performance est limitée : seize individus mal classés.

La plus simple de la deuxième configuration est intéressante. Elle contient une règle supplémentaire, définie également par une seule variable. La base ne contient donc aucune règle d'interaction, dont la prémisse serait définie par au moins deux variables. Nous pouvons également noter que cette dernière règle est définie par le sous-ensemble flou numéro cinq. Pour cette variable, la longueur des pétales, il s'agit du seul label utilisé. Est-il vraiment nécessaire de définir cinq sous-ensembles flous pour n'en utiliser qu'un seul ? Quelle sémantique attacher à cette règle ?

La différence entre les deux systèmes simplifiés issus de la deuxième configuration réside dans la définition de la prémisse de la dernière règle. Le prix à payer pour obtenir un individu supplémentaire bien classé est excessif : au lieu de définir la règle par une variable, trois sont dans ce cas nécessaires. Dans la perspective de réutilisation du système il est souvent préférable de le concevoir le plus simple possible car le plus apte à travailler en généralisation. Cette optimisation, pour un gain négligeable, consiste en l'apprentissage d'une donnée très particulière. Elle serait évitée par l'emploi d'une technique de validation au cours de l'apprentissage.

Le nombre de variables maximum éliminées dans une règle est à mettre en relation avec le nombre moyen de variables éliminées par règle, soit le rapport du nombre de variables éliminées au nombre de règles. Ces deux chiffres donnent une idée de l'hétérogénéité du nombre de variables nécessaires à la définition d'une règles. Ils peuvent permettre d'attirer l'attention sur des règles qui seraient définies avec un nombre de variables nettement supérieur à la moyenne. De telles règles peuvent correspondre à des exceptions, mais aussi résulter d'une procédure trop contrainte.

Le tableau 4.7 indique les occurrences des différentes variables dans les règles simplifiées. Il montre que les variables les plus présentes dans les règles sont la largeur et la longueur des pétales. Nous allons poursuivre l'étude avec ces seules variables.

Les deux premières variables sont désactivées, et nous nous plaçons dans la deuxième configuration. Puis, dans la même configuration, nous ne conservons que la largeur des pétales. Les résultats sont rapportés dans les tableaux 4.8 et 4.9.

Base de règles des différents systèmes			
Règles complètes		Simplifié performant	Le plus simple
C1			
1, 1, 1, 1,	1	0, 0, 0, 1, 1	
1, 1, 2, 2,	2	0, 0, 0, 2, 2	
1, 2, 1, 1,	1	0, 0, 0, 3, 3	
2, 1, 2, 2,	2		
2, 2, 1, 1,	1		
2, 2, 2, 2,	2		
2, 1, 3, 2,	3		
2, 1, 2, 3,	3		
2, 2, 2, 3,	3		
3, 1, 3, 3,	3		
3, 2, 3, 3,	3		
C2			
1, 1, 1, 1,	1	0, 0, 0, 2, 2	0, 0, 0, 2, 2
1, 1, 2, 1,	1	0, 0, 0, 1, 1	0, 0, 0, 1, 1
1, 2, 1, 1,	1	0, 0, 0, 3, 3	0, 0, 0, 3, 3
1, 2, 2, 1,	1	2, 1, 5, 0, 3	0, 0, 5, 0, 3
1, 1, 3, 2,	2		
2, 1, 3, 2,	2		
2, 1, 4, 2,	2		
2, 1, 5, 2,	3		
2, 1, 5, 3,	3		
2, 2, 5, 3,	3		
2, 1, 6, 3,	3		
2, 2, 6, 3,	3		
C3			
1, 4, 1, 1,	1	2, 0, 0, 0, 3	
1, 5, 1, 1,	1	1, 0, 1, 1, 1	
1, 3, 1, 1,	1	2, 0, 0, 2, 2	
2, 3, 2, 2,	2		
2, 3, 2, 3,	3		
2, 3, 3, 3,	3		
2, 1, 2, 2,	2		
2, 2, 3, 3,	3		
2, 5, 3, 3,	3		
C4			
1, 1, 1, 1,	1	0, 0, 1, 1, 1	0, 0, 1, 1, 1
1, 1, 2, 2,	2	0, 1, 2, 0, 2	0, 1, 2, 0, 2
1, 2, 1, 1,	1	0, 0, 0, 3, 3	0, 0, 0, 3, 3
2, 1, 2, 2,	2	0, 1, 2, 3, 3	0, 0, 2, 3, 3
2, 2, 1, 1,	1		
2, 1, 2, 3,	3		
3, 1, 3, 3,	3		
3, 2, 3, 3,	3		
C5 - C6 - C8 - C9			
1, 1, 1, 1,	1	1, 0, 1, 1, 1	
1, 2, 1, 1,	1	2, 0, 2, 0, 3	
2, 1, 2, 2,	2	0, 0, 0, 2, 2	
2, 1, 2, 3,	3		
2, 2, 2, 3,	3		
C7			
1, 1, 1, 1,	1	1, 0, 1, 1, 1	
1, 1, 2, 2,	2	0, 1, 2, 2, 2	
1, 2, 1, 1,	1	2, 1, 0, 3, 3	
2, 1, 2, 2,	2		
2, 1, 2, 3,	3		
2, 1, 3, 3,	3		

TAB. 4.6 – Règles des systèmes sélectionnés et simplifiés des différentes configurations

Variable	C1	C2	C3	C4	C5	C7	Total
Longueur des sépales	0	0	3	0	2	2	7
Largeur des sépales	0	0	0	1	2	0	3
Longueur des pétales	0	1	1	3	2	2	9
Largeur des pétales	3	3	2	3	3	2	16

TAB. 4.7 – Occurrences des variables dans les règles simplifiées

	Sélection					Simplification				
	Nb Sef	Max R	#R	M cl	#I	#R	M cl	#I	#E	Max E
longueur et largeur	0 0 3 3	9	4	6	0	3	6	0	2	1
largeur	0 0 0 6	6	6	6	0	5	6	0	0	0

TAB. 4.8 – Résultats de la configuration C2 sur les pétales des iris

La réduction du système à la seule largeur des pétales conduit à un nombre de sous-ensembles flous important. La configuration obtenue après simplification du système comprenant les deux caractéristiques des pétales nous semble être à la fois la plus lisible et la plus performante. Chacune des deux entrées est découpée en trois sous-ensembles flous, ce nombre est compatible avec le critère de qualité d'une partition défini au chapitre précédent. Les conclusions, notées dans la section 3.5.6, indiquent en effet que l'optimum suivant ce critère se situe autour de trois sous-ensembles flous pour chacune de ces variables.

Interprétation des résultats

Trois règles sont retenues, soit une par variété. Deux sont définies par une seule variable, la dernière en nécessite deux. Elles s'énoncent de la façon suivante :

- Si la largeur des pétales est petite alors l'iris est de type Setosa
- Si la largeur des pétales est grande alors l'iris est de type Virginica
- Si la largeur des pétales est moyenne et si la longueur des pétales est moyenne alors l'iris est de type Versicolor

La performance, six iris mal classés, figure parmi les meilleures.

Les résultats peuvent être présentés sous la forme d'une matrice de confusion, tableau 4.10. Les colonnes représentent les classes réellement observées, les lignes les classes prédites. Chacune des cases représente un effectif. Le chiffre 1 sur la troisième ligne et la deuxième colonne indique qu'un individu de la classe 2 a été affecté par le système à la classe 3. Le résultat idéal consiste en une matrice diagonale.

Ces résultats peuvent être comparés à ceux obtenus par d'autres méthodes de classification. Les tableaux 4.11 et 4.12 contiennent les matrices de confusion obtenues par l'analyse factorielle discriminante en calibration et en validation croisée (Martens & Naes, 1989).

La figure 4.1 montre la répartition des individus dans le plan factoriel. La classe 1, *Setosa*, est nettement séparée des deux autres.

Ainsi, la méthode proposée s'est avérée particulièrement efficace sur cette application. Elle est aussi performante que nombre de techniques aveugles, de type boîte noire, tout en étant caractérisée par des règles qu'un botaniste peut aisément vérifier.

```
Longueur et largeur des pétales
            0, 0, 1, 1,    1          0, 0, 0, 3,    3
            0, 0, 2, 2,    2          0, 0, 0, 1,    1
            0, 0, 3, 2,    3          0, 0, 2, 2,    2
            0, 0, 2, 3,    3


Largeur des pétales
            0, 0, 0, 1,    1          0, 0, 0, 1,    1
            0, 0, 0, 2,    1          0, 0, 0, 2,    1
            0, 0, 0, 3,    2          0, 0, 0, 3,    2
            0, 0, 0, 4,    2          0, 0, 0, 4,    2
            0, 0, 0, 5,    3          0, 0, 0, 6,    3
            0, 0, 0, 6,    3
```

TAB. 4.9 – Les règles des systèmes formés à partir des caractéristiques des pétales

Prédit \ Réel	1	2	3
1	50	0	0
2	0	49	5
3	0	1	45

TAB. 4.10 – Matrice de confusion obtenue par le système d'inférence floue

Prédit \ Réel	1	2	3
1	50	0	0
2	0	49	3
3	0	1	47

TAB. 4.11 – Matrice de confusion obtenue par l'analyse factorielle discriminante en calibration

Prédit \ Réel	1	2	3
1	50	0	0
2	0	48	4
3	0	2	46

TAB. 4.12 – Matrice de confusion obtenue par l'analyse factorielle discriminante en validation

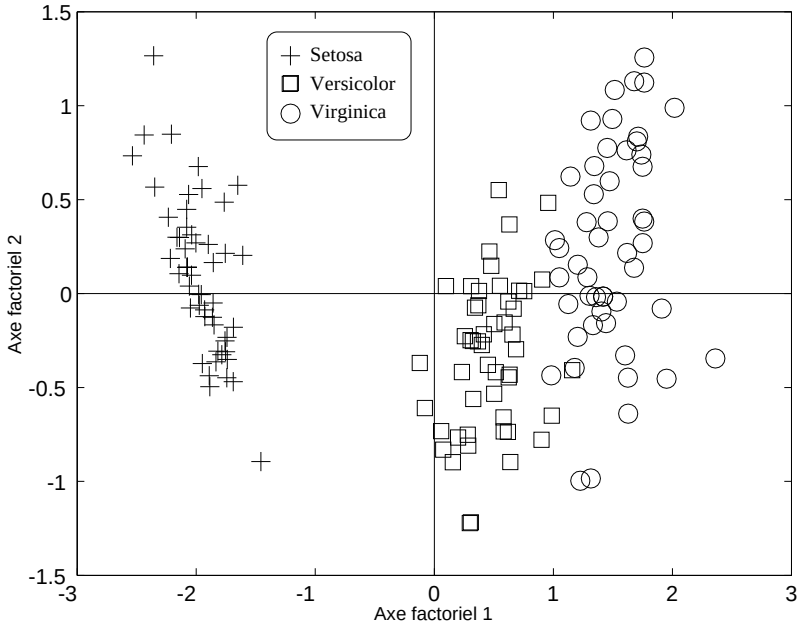


FIG. 4.1 – Plan factoriel construit par l'analyse factorielle discriminante pour les iris

4.4.2 La qualité du riz

Contrairement au jeu de données précédent, la sortie de celui-ci est une variable continue, la qualité globale, qui varie entre zéro et un. L'objectif est définir les règles sous-jacentes dans le raisonnement des experts pour apprécier la qualité globale du riz en fonction de différentes caractéristiques : l'arôme, l'apparence, le goût, le collant et la dureté.

Bien qu'il s'agisse initialement d'appréciations sensorielles, les données consistent en des valeurs numériques, normalisées dans l'espace unité. Elles représentent le codage de l'agrégation d'évaluations d'un panel d'experts sur différentes qualités de riz. La perte d'information induite par cette agrégation et par ce codage, n'est pas maîtrisée. Nous n'avons pas pu, malheureusement, nous procurer les évaluations sensorielles avant leur mise en forme numérique.

L'application a été présentée par (Ishibuchi et al., 1994), puis revisitée par (Nozaki et al., 1997) et plus récemment par (Cordon & Herrera, 2000). Le jeu de données est formé de 105 individus. Les distributions des variables d'entrée et de sortie sont rapportées sur la figure 4.2.

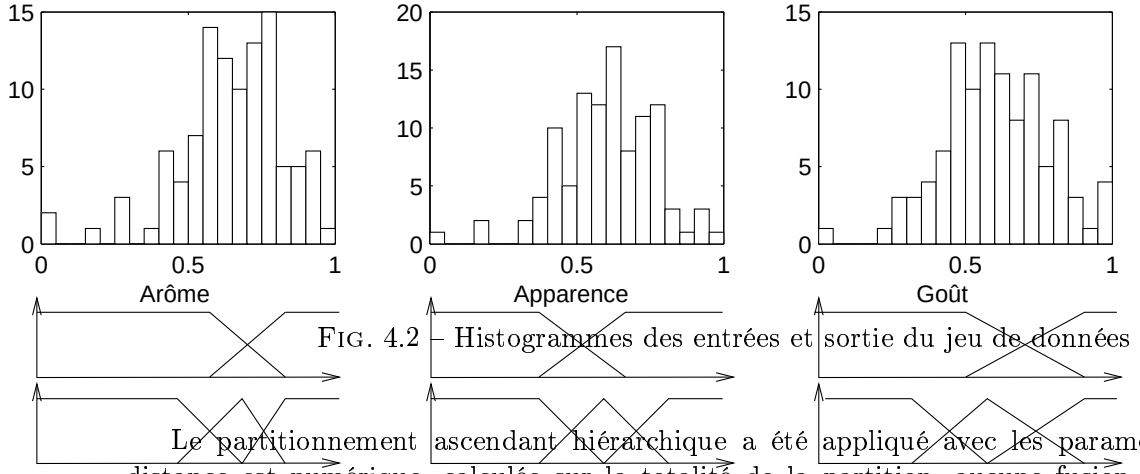
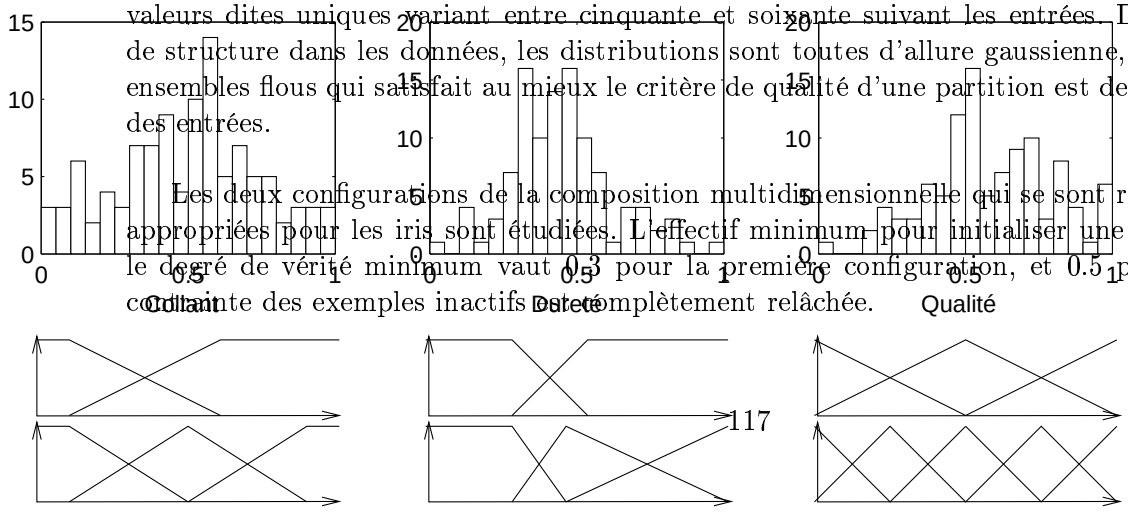


FIG. 4.2 – Histogrammes des entrées et sortie du jeu de données Riz

Le partitionnement ascendant hiérarchique a été appliqué avec les paramètres suivants : la distance est numérique, calculée sur la totalité de la partition, aucune fusion n'est pénalisée, la tolérance d'égalité sur les valeurs uniques est fixée à un pour cent. Ce qui conduit à un nombre de valeurs dites uniques variant entre cinquante et soixante suivant les entrées. Du fait de l'absence de structure dans les données, les distributions sont toutes d'allure gaussienne, le nombre de sous-ensembles flous qui satisfait au mieux le critère de qualité d'une partition est de deux pour chacune des entrées.

Les deux configurations de la composition multidimensionnelle qui se sont révélées être les plus appropriées pour les iris sont étudiées. L'effectif minimum pour initialiser une règle est fixé à un, le degré de vérité minimum vaut 0.3 pour la première configuration, et 0.5 pour la seconde. La cohérence des exemples inactifs est complètement relâchée.



La performance du système n'est plus mesurée par le nombre de mal classés mais par trois indicateurs : l'erreur moyenne, ErM, calculée suivant l'équation 4.6 ; l'indice utilisé dans (Cordon & Herrera, 2000), appelé ici *Err Riz*, calculé suivant l'équation 4.7 ; et enfin, l'écart maximum entre la valeur inférée et la sortie observée sur l'ensemble des exemples, noté $ErMax$. Dans ces équations, y_i et $\widehat{y_i}$ représentent les sorties observée et inférée par le système pour l'exemple i.

$$ErM = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \tag{4.6}$$

$$Err\ Riz = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{4.7}$$

Résultats publiés par Cordon et Herrera

Dans (Cordon & Herrera, 2000), les auteurs visent aussi à la construction d'un système interprétable. Leur méthode de génération est basée sur celle de (Wang & Mendel, 1992b), présentée dans la section 2.1.3. La conclusion des règles est un ensemble flou. Les règles sont ensuite sélectionnées par un algorithme génétique. Deux partitionnements des entrées sont étudiés. Chacun d'eux consiste en une grille triangulaire régulière formant une partition floue forte d'un nombre fixe de sous-ensembles flous pour chacune des entrées. Dans le premier cas, le nombre de sous-ensembles flous est égal à deux, dans le second cas, il vaut trois. Du jeu de données initial sont extraits, par tirage aléatoire, dix échantillons d'apprentissage formés de 75 individus, et dix échantillons de test comprenant 30 exemples.

Leurs résultats sont résumés dans le tableau 4.13. *Err App* et *Err Test* représentent les valeurs moyennes de *Err Riz* pour les ensembles d'apprentissage et de test, $\#R$ est le nombre de règles moyen. Comme nous n'avons pas appliqué de procédure de validation, nos résultats seront comparés avec ceux qu'ils ont obtenus, en moyenne, sur l'exemple d'apprentissage.

	Après génération			Après sélection		
Nb Sef	#R	Err App	Err Test	#R	Err App	Err Test
2 2 2 2 2	19.8	0.02192	0.02412	5	0.00341	0.00398
3 3 3 3 3	25.7	0.00595	0.00736	12.2	0.00185	0.00290

TAB. 4.13 – Résultats obtenus par Cordon et Herrera sur le riz

La conclusion des règles est une constante

L'application de notre méthode d'induction conduit aux résultats rassemblés dans le tableau 4.14. Rappelons que la signification des différents indicateurs a été définie dans le tableau 4.1. La conclusion des règles induites est une constante initialisée suivant la formule de l'équation 4.1.

Le nombre d'exemples inactifs atteint une valeur importante, 43 sur 105 individus, avec la deuxième configuration. Cela provient du paramétrage : le seuil d'inactivité est de 0.5, alors qu'il vaut 0.3 pour la première configuration.

Notre meilleure performance, obtenue par la première configuration, est comparable à ces résultats. Le nombre de règles induites par notre méthode est sensiblement supérieur au leur. La valeur de *Err Riz* est dans ce cas de 0.00285 pour le système simplifié le plus performant, et de 0.00250

pour le système formé des règles complètes correspondant. L'erreur moyenne est de sept pour mille. Ces indices, qui travaillent en moyenne, ne fournissent qu'une vague indication de la performance du système.

	Indicateurs des différents systèmes								
	Nb Sef	Max R	#R	ErM	Err Riz	ErMax	#I	#E	Max E
C1	3 2 2 4 2	96	24	0.00690	0.00250	0.164	4		
C1 (sp)			14	0.00737	0.00285	0.198	0	6	2
C1 (si)			6	0.01094	0.00629	0.331	0	8	4
C2	2 3 2 2 2	48	11	0.01013	0.00539	0.186	43		
C2 (si)			9	0.00970	0.00494	0.290	0	8	4

TAB. 4.14 – Résultats des deux configurations testées sur le riz

L'écart maximal entre la sortie inférée et la sortie observée complète ces indices. Cette valeur augmente nettement avec la simplification. C'est un résultat attendu. La présence de certaines règles dans la base, ou de certaines variables dans la définition des règles n'est justifiée que par un petit nombre d'exemples. La simplification, sans trop dégrader l'erreur moyenne, produit des résultats significatifs sur ces exemples.

Les figures 4.3 et 4.4 proposent une représentation graphique des erreurs de prédiction. La première colonne est relative au système issu de la sélection, la seconde au système simplifié le plus performant, et, éventuellement, la troisième au système le plus simple, s'il est différent du plus performant. Sur la première ligne sont tracés les graphes indiquant la valeur inférée en fonction de la valeur observée. Sur la deuxième, l'écart est représenté en fonction de la valeur observée. Enfin, la dernière ligne contient l'histogramme des écarts.

La distribution de la sortie montre qu'il y a une seule valeur nulle détachée du reste de l'histogramme, la suivante vaut 0.18. Cette valeur est plus mal gérée par la configuration un que par la seconde, et dans les deux cas, les performances, pour cette valeur, se dégradent avec la simplification. Les remarques que nous avons faites sur l'écart maximum sont illustrées graphiquement par le changement d'échelle sur l'axe des ordonnées des représentations des écarts en fonction des sorties observées pour les systèmes les plus simples. La valeur de l'écart ne semble pas être corrélée avec la valeur de la sortie observée pour les systèmes issus de la sélection. En revanche, on peut observer la présence d'auto-corrélation dans la distribution des écarts des systèmes les plus simples. Ceci indique clairement que le modèle simplifié est inadapté.

Cette conclusion est confirmée par l'examen des règles, présentées dans le tableau 4.15. Pour le système simplifié le plus performant de la première configuration, six variables sont éliminées avec au maximum, deux dans une règle. Cet équilibre est complètement détruit dans les systèmes les plus simples. Deux règles sont définies avec une seule variable, tandis que toutes les autres sont des règles complètes.

La procédure de simplification elle-même, ainsi que la lisibilité des règles produites, sont gênées par le fait que les conclusions des règles sont des valeurs numériques. Celles-ci sont parfois très proches les unes des autres, ce qui rend l'interprétation des règles difficile. Pour améliorer la lisibilité de la base de règles, la conclusion doit être un ensemble flou, interprétable comme un label linguistique.

Base de règles des deux configurations pour le riz

Règles complètes		Simplifié performant		Le plus simple	
C1:					
3, 2, 2, 3, 1,	0.827	0, 2, 0, 2, 2,	0.528	0, 0, 0, 0, 2,	0.528
3, 2, 2, 3, 2,	0.708	3, 2, 2, 4, 0,	0.971	0, 0, 0, 4, 0,	0.971
3, 2, 2, 4, 1,	0.984	0, 1, 1, 2, 2,	0.391	3, 2, 1, 0, 1,	0.566
3, 2, 2, 4, 2,	1.000	2, 0, 1, 2, 2,	0.468	1, 1, 1, 1, 2,	0.134
1, 1, 1, 1, 2,	0.134	3, 2, 1, 0, 1,	0.566	3, 1, 1, 1, 2,	0.298
2, 1, 1, 1, 2,	0.246	1, 1, 1, 1, 2,	0.134	1, 2, 1, 3, 1,	0.448
1, 1, 1, 1, 1,	0.285	2, 2, 1, 3, 1,	0.702		
1, 1, 1, 2, 2,	0.357	2, 2, 2, 3, 1,	0.722		
1, 2, 1, 2, 1,	0.416	3, 2, 1, 3, 2,	0.708		
1, 2, 1, 2, 2,	0.463	3, 1, 1, 1, 2,	0.298		
2, 1, 1, 2, 2,	0.433	1, 1, 1, 3, 2,	0.241		
2, 2, 1, 1, 2,	0.394	1, 2, 1, 3, 1,	0.448		
2, 2, 1, 2, 2,	0.497	1, 2, 1, 3, 2,	0.431		
2, 2, 1, 3, 1,	0.702	1, 1, 2, 4, 1,	0.734		
2, 2, 2, 2, 2,	0.597				
2, 2, 2, 3, 1,	0.722				
3, 2, 1, 3, 1,	0.531				
3, 2, 1, 3, 2,	0.708				
3, 2, 1, 4, 1,	0.741				
3, 1, 1, 1, 2,	0.298				
1, 1, 1, 3, 2,	0.241				
1, 2, 1, 3, 1,	0.448				
1, 2, 1, 3, 2,	0.431				
1, 1, 2, 4, 1,	0.734				
C2 :					
2, 2, 2, 2, 1,	0.861	1, 0, 0, 0, 0,	0.489		
2, 3, 2, 2, 1,	0.968	0, 0, 0, 0, 1,	0.861		
1, 1, 1, 1, 2,	0.090	1, 1, 1, 1, 2,	0.090		
1, 1, 1, 1, 1,	0.285	1, 1, 1, 1, 1,	0.285		
1, 1, 1, 2, 2,	0.241	1, 1, 1, 2, 2,	0.241		
1, 2, 1, 1, 2,	0.474	2, 1, 1, 1, 2,	0.298		
1, 2, 1, 2, 2,	0.485	1, 3, 2, 2, 1,	0.819		
2, 1, 1, 1, 2,	0.298	2, 3, 1, 2, 1,	0.531		
1, 3, 2, 2, 1,	0.819	1, 1, 2, 2, 1,	0.734		
2, 3, 1, 2, 1,	0.531				
1, 1, 2, 2, 1,	0.734				

TAB. 4.15 – Règles des systèmes sélectionnés et simplifiés pour le jeu de données Riz

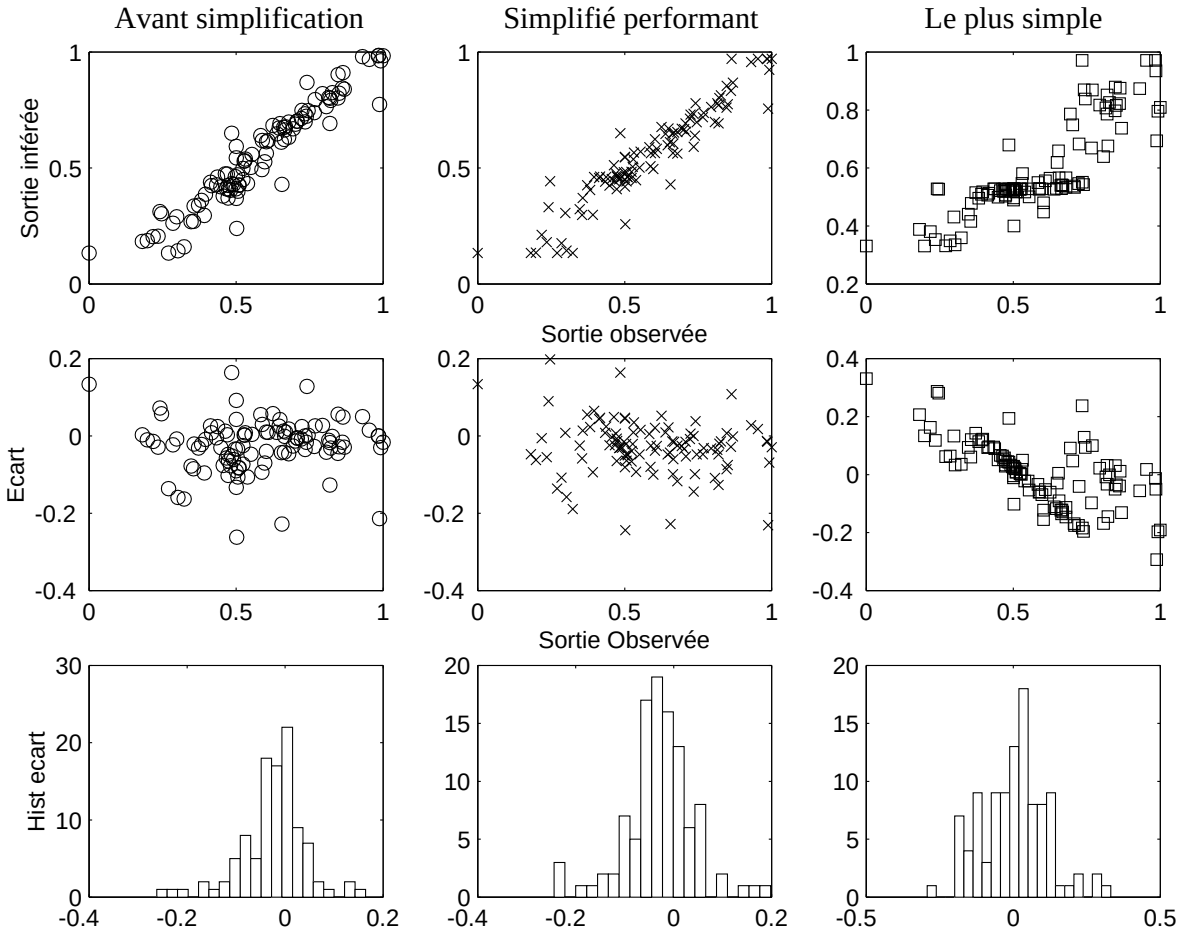


FIG. 4.3 – Représentation des erreurs de prédiction pour la configuration un

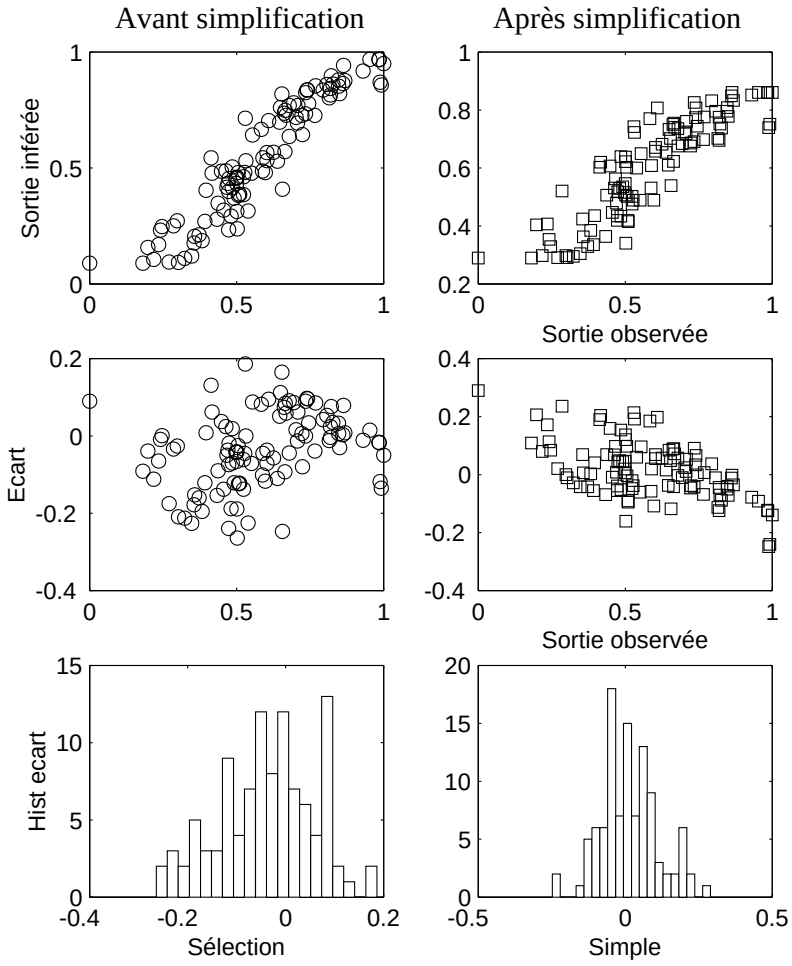


FIG. 4.4 – Représentation des erreurs de prédiction pour la configuration deux

La conclusion des règles est un ensemble flou

La sortie est découpée en sous-ensembles flous triangulaires répartis régulièrement.

Les conclusions floues des règles sont initialisées en deux temps : tout d'abord une conclusion nette est calculée comme précédemment, puis celle-ci est fuzzifiée. La conclusion floue est l'étiquette de l'ensemble pour lequel le degré d'appartenance de la conclusion nette est le plus fort.

Nous utilisons la méthode des aires pour la défuzzification. Pour chacune des règles activées par l'exemple, la conclusion est caractérisée par l'alpha-coupe correspondant au degré de vérité de la règle et par l'abscisse du centre de gravité de cette alpha-coupe. L'agrégation est une pondération de ces abscisses par les aires, comme indiqué dans la formule :

$$\widehat{y_i} = \frac{\sum_{r=1}^{R} \alpha^{C^r} \; area(C_\alpha^r)}{\sum_{r=1}^{R} area(C_\alpha^r)}$$

Dans cette équation C_α^r est l'alpha-coupe de l'ensemble C^r, avec $\alpha = w^r(x_i)$, et α^{C^r} est l'abscisse du centre de gravité de C_α^r.

La défuzzification est illustrée sur la figure 4.5.

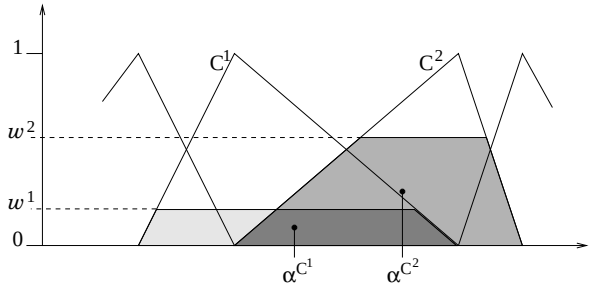


FIG. 4.5 – Défuzzification : méthode des aires

Les performances du système dépendent du nombre de sous-ensembles flous. S'il est trop faible, le système ne pourra pas rendre compte de certains comportements, s'il est trop important, certains d'entre eux risquent d'être spécifiques à quelques données seulement. Dans ce cas, les performances en apprentissage peuvent se dégrader sur des données inconnues. Nous avons choisi de tester deux découpages : trois et cinq sous-ensembles flous. Les tests seront effectués pour chacune des configurations déjà étudiées avec des conclusions de règles nettes, effectif minimum d'initialisation des règles égal à un et degré de vérité minimum minimum égal à 0.3 ou 0.5.

Les résultats sont rassemblés dans la tableau 4.16. Ils sont à rapprocher de leurs homologues, obtenus avec des conclusions nettes, synthétisés dans le tableau 4.14.

Le découpage en trois sous-ensembles flous conduit à des solutions peu performantes. Notre effort d'analyse portera sur les systèmes dont la sortie est découpée en cinq sous-ensembles flous.

Pour la première configuration, le minimum de la sélection, noté *C1-1*, est obtenu lors de la neuvième itération. Il n'améliore que faiblement le premier minimum local qui correspond à un système, noté *C1-2*, qui peut être comparé à son homologue. Le nombre de règles, 24, est identique. Les erreurs moyennes sont semblables : 0.00750 pour l'un, 0.00690 pour l'autre. Les structures

	Indicateurs des différents systèmes									
	Nb Sef	Nb Sef Sortie	Max R	#R	ErM	Err Riz	ErMax	#I	#E	Max E
C1	3 2 4 2 3	3	144	25	0.01125	0.00664	-0.349	4		
C1 (si)				18	0.01134	0.00676	-0.349	7	4	1
C2	2 3 2 2 2	3	48	11	0.01060	0.00589	-0.316	43		
C2 (si)				10	0.01346	0.00952	-0.406	0	17	4
C1-1	5 2 4 4 4	5	640	45	0.00673	0.00238	0.259	10		
C1-1 (sp)				32	0.00631	0.00209	-0.206	1	22	2
C1-1 (si)				15	0.00947	0.00471	0.290	0	8	2
C1-2	3 2 2 4 2	5	96	24	0.00750	0.00295	0.250	4		
C1-2 (sp)				21	0.00711	0.00266	0.250	1	7	1
C1-2 (si)				15	0.00909	0.00434	-0.319	0	2	2
C2	2 3 2 2 3	5	72	12	0.00834	0.00365	-0.319	39		
C2 (si)				8	0.00935	0.00459	-0.319	3	1	1

TAB. 4.16 – Résultats des configurations testées, avec conclusion floue, sur le riz

des systèmes simplifiés, *C1 (sp)* pour les conclusions nettes, *C1-2 (si)* pour les conclusions floues, sont également voisines : 14 règles dans un cas, 15 dans l'autre. Le nombre maximum de variables éliminées dans une règle est 2 dans les deux cas. Comme pour les systèmes issus de la sélection, l'erreur moyenne est plus importante lorsqu'on utilise les sorties floues. Nous pouvons penser qu'il s'agit du prix à payer pour obtenir des règles interprétables, comme le montre le tableau 4.17.

L'utilisation de règles dont la conclusion est floue permet d'améliorer notablement leur lisibilité. En revanche, il semble que ces systèmes soient moins faciles à simplifier que leurs homologues dont la conclusion des règles est nette. Cette observation est à relier à la remarque précédente relative à la précision. La dégradation des performances due aux conclusions floues limite la marge de manœuvre pour la procédure de simplification. La méthode d'initialisation des conclusions floues, qui se limite à fuzzifier une conclusion nette, n'est sans doute pas optimale. Les indicateurs relatifs aux variables éliminées montrent que le système, indépendamment de la nature de la conclusion des règles, n'est pas facile à simplifier, chacune des variables semble être importante.

Les règles restent difficiles à interpréter. Si l'on considère le système issu de la configuration $C2$ simplifiée, nous pouvons observer que certaines caractéristiques semblent assez bien reliées à la qualité globale, comme le goût et le collant, respectivement variables 3 et 4. L'arôme, variable 1, présente un comportement semblable excepté pour la règle numéro sept, pour laquelle une valeur faible pour l'arôme conduit, dans un contexte favorable pour le goût et le collant, à une bonne qualité. L'apparence, variable 2, et la dureté, variable 5, montrent un comportement plus chaotique.

Des compléments sur cette application, consécutifs à une modification de l'algorithme qui sera décrite dans le chapitre suivant, sont présentés dans l'annexe F. Ils contiennent des éléments d'interprétation des règles.

Base de règles des deux configurations pour le riz : conclusions floues

Règles complètes		Simplifié performant		Le plus simple	
C1-2:					
3, 2, 2, 3, 1,	4	1, 2, 1, 0, 0,	3		
3, 2, 2, 3, 2,	4	3, 2, 2, 3, 1,	4		
3, 2, 2, 4, 1,	5	3, 2, 2, 4, 1,	5		
3, 2, 2, 4, 2,	5	3, 2, 2, 4, 2,	5		
1, 1, 1, 1, 2,	2	1, 1, 1, 1, 2,	2		
2, 1, 1, 1, 2,	2	2, 1, 1, 1, 2,	2		
1, 1, 1, 1, 1,	2	1, 1, 1, 1, 1,	2		
1, 1, 1, 2, 2,	2	1, 1, 1, 2, 2,	2		
1, 2, 1, 2, 1,	3	2, 1, 1, 2, 2,	3		
1, 2, 1, 2, 2,	3	2, 2, 1, 1, 2,	3		
2, 1, 1, 2, 2,	3	3, 2, 1, 3, 1,	3		
2, 2, 1, 1, 2,	3	3, 2, 1, 4, 1,	4		
2, 2, 1, 2, 2,	3	3, 1, 1, 1, 2,	2		
2, 2, 1, 3, 1,	4	1, 1, 1, 3, 2,	2		
2, 2, 2, 2, 2,	3	1, 1, 2, 4, 1,	4		
2, 2, 2, 3, 1,	4				
3, 2, 1, 3, 1,	3				
3, 2, 1, 3, 2,	4				
3, 2, 1, 4, 1,	4				
3, 1, 1, 1, 2,	2				
1, 1, 1, 3, 2,	2				
1, 2, 1, 3, 1,	3				
1, 2, 1, 3, 2,	3				
1, 1, 2, 4, 1,	4				
C2 :					
2, 2, 2, 2, 1,	4	1, 2, 1, 0, 2,	3		
2, 3, 2, 2, 1,	5	2, 2, 2, 2, 1,	4		
2, 3, 2, 2, 2,	5	2, 3, 2, 2, 1,	5		
1, 1, 1, 1, 2,	2	1, 1, 1, 1, 3,	1		
1, 1, 1, 1, 3,	1	2, 2, 1, 2, 2,	4		
1, 2, 1, 1, 2,	3	1, 1, 1, 1, 1,	2		
1, 2, 1, 2, 2,	3	1, 1, 2, 2, 1,	4		
2, 1, 1, 1, 3,	2	2, 3, 1, 2, 1,	3		
2, 2, 1, 2, 2,	4				
1, 1, 1, 1, 1,	2				
1, 1, 2, 2, 1,	4				
2, 3, 1, 2, 1,	3				

TAB. 4.17 – Règles des systèmes sélectionnés et simplifiés pour le jeu de données Riz, la sortie est découpée en cinq ensembles flous

4.5 Conclusion

La composition multidimensionnelle s'appuie sur les familles de partitions monodimensionnelles générées par la méthode présentée au chapitre précédent pour constituer une méthode originale d'induction de règles floues interprétables.

Elle est réalisée en deux étapes. La première consiste à générer un ensemble de systèmes formés de règles complètes, puis à sélectionner, parmi ceux-ci, l'un des plus performants et des plus compacts. La seconde étape vise à simplifier la base de règles.

Nous avons introduit trois niveaux de simplification. Le premier est celui d'un groupe de règles. Un groupe définit un contexte au sein duquel est vérifiée l'influence d'une variable. Si cette influence est jugée insuffisante, les règles du groupe sont fusionnées en une seule. Le deuxième niveau est celui d'une règle : pour éliminer la redondance, les règles inutiles sont supprimées de la base. Enfin, le dernier niveau est celui des variables qui définissent la prémisse de la règle : celles qui sont inutiles sont éliminées.

L'ensemble de cette procédure est guidée par trois indices. L'indice de performance garantit la précision de la sortie inférée. L'indice de couverture nous assure qu'une bonne partie de l'ensemble d'apprentissage est gérée par le système. L'indice d'homogénéité des sorties des exemples attirés par la règle permet de valider l'élargissement de l'espace d'entrée couvert par la règle lors des opérations de fusion des groupes de règles ou d'élimination des variables.

Deux exemples, déjà étudiés dans la littérature, sont traités : l'un de classification, les iris, l'autre de régression, la qualité du riz.

Les résultats sont satisfaisants pour le problème de classification. Les trois classes sont gérées par trois règles, une par classe, définies par une ou deux variables. Ces trois règles sont facilement interprétables par un botaniste. Sur les 150 individus de l'ensemble d'apprentissage, 6, seulement, sont mal classés, ce qui représente une performance particulièrement honorable.

L'étude du problème de régression a été réalisée avec les mêmes paramètres, pour la composition multidimensionnelle, que ceux utilisés pour la classification. Elle soulève d'autres questions. Pour les sorties continues, les bords de distribution de faible effectif participent peu à l'initialisation des règles. Si les règles correspondantes disparaissent dès les premières étapes de la simplification, aucune interpolation possible vers les valeurs extrêmes de sortie n'est possible. Faut-il imposer, par la contrainte, le maintien de règles capables d'inférer les valeurs extrêmes ? Les trois indices utilisés garantissent cela pour la classification. Pour l'obtenir dans le cas de la régression, est-il possible d' ajuster les paramètres existants, ou bien est-il nécessaire de définir un nouvel indice ? Nous avons bien conscience que l'étude de la sensibilité de la méthode aux paramètres, ébauchée dans ce chapitre, reste à approfondir.

Lorsque la conclusion des règles induites est une valeur numérique, le fait que certaines sont très proches les unes des autres ne facilite pas l'interprétation des règles, ni, sans doute, la procédure de simplification. Les résultats obtenus ne sont pas bons. Les règles simplifiées ne sont pas homogènes, certaines sont complètes, d'autres définies par une seule variable ; l'écart est corrélé à la valeur de sortie désirée. L'usage de conclusions floues améliore la lisibilité des règles, la qualité globale semble être reliée plus particulièrement à l'arôme au goût et au collant.

L'indice de performance, utilisé dans l'ensemble des étapes de sélection et de simplification, a une grande influence sur les résultats. La discussion a montré que la performance ne pouvait pas être réduite à un indice du type erreur moyenne. Elle doit aussi intégrer des caractéristiques de la

distribution des écarts et être reliée au niveau de couverture de l'ensemble d'apprentissage. Est-il possible d'envisager une métrique qui prenne en compte l'ensemble de ces éléments ? Comment définir un critère en fonction d'objectifs qui ne se résument pas à l'amélioration de l'erreur moyenne ?

En dépit de toutes ces difficultés, nous allons maintenant quitter les sentiers balisés d'applications déjà étudiées pour nous intéresser aux mécanismes de formation de la couleur au cours de la vinification du vin rouge.

Sans vouloir anticiper sur le prochain chapitre, nous indiquons au lecteur que des résultats complémentaires sur l'application du riz se trouvent dans l'annexe F.

Chapitre 5

Application à un procédé de vinification

On le sait maintenant, l'ordre c'est l'exception. Dans le cœur, dans le cerveau, les régularités absolues sont les signes d'un dysfonctionnement grave. L'être sain a besoin pour vivre de désordre.

Gérard Clergue

(Clergue, 1997)

Sommaire

L'objectif de l'étude est la conception d'un simulateur de l'impact des choix de conduite de vinification sur la couleur et les tanins du vin rouge[a].

Il a été décidé d'étudier prioritairement la couleur car les tanins sont plus difficiles à mesurer. La couleur est caractérisée par une variable numérique appelée intensité colorante. Il s'agit de la somme de deux densités optiques mesurées aux longueurs d'onde de 420 et 520 nanomètres, ce qui correspond aux couleurs bleu pourpre et vert.

L'application décrite dans ce chapitre est contractuelle et conduite en partenariat avec le centre de transfert de technologie TRIAL et la cave coopérative La Malepère, située à Arzens, au sud ouest de Carcassonne. Le problème traité présente un caractère générique, les résultats de cette étude seront à la disposition des acteurs de l'ensemble de la filière régionale.

La cave d'Arzens est l'une des plus importantes de France : elle rassemble 345 adhérents qui cultivent environ 2600 hectares. La production correspondante varie, suivant les années, entre 150000 et 200000 hectolitres. Ce vin est produit avec un procédé traditionnel et des matériels de haute technologie. La production se décompose en trois types de vins, auxquels correspondent trois types de vinification. Pour les vins de pays la vinification est économique, seul le remontage, ou le pigeage suivant l'équipement de la cuve, est réalisé. Pour la vinification des vins de cépage, il est de plus procédé à un levurage et à un enzymage à dose variable. Enfin, les vins de garde ont droit à d'autres égards : des tanins exogènes peuvent leur être apportés, une micro-oxygénation permet de stabiliser la couleur. En outre la cave dispose d'un groupe de froid permettant éventuellement d'augmenter la phase pré-fermentaire en refroidissant la cuve au voisinage de dix degrés.

Aujourd'hui la demande des consommateurs évolue vers des vins plus tanniques et plus colorés. La cave ne peut répondre à la totalité de cette demande, elle estime la proportion de ses vins de cépage susceptible d'être vendue dans ces conditions à vingt pour cent, ce qui représente un volume de douze mille hectolitres.

La vinification est un procédé complexe, et les mécanismes de formation de la couleur et des tanins sont mal connus. Les anthocyanes, responsables de la coloration du vin, ont été longuement étudiées. Nous savons que leur stabilité est variable du fait des interactions avec les tanins et l'oxygène, que l'intensité de leur couleur dépend également de facteurs du milieu comme la teneur en SO_2 ou le pH. Mais, bien que les phénomènes chimiques liés à la fermentation et à l'extraction soient maintenant connus, car étudiés en laboratoire, les règles opérationnelles pour la gestion des vinifications font défaut. L'environnement de la cuve en situation ne peut pas être maîtrisé comme celui du laboratoire. Tenant compte de la diversité de la matière qu'il a à vinifier, l'œnologue cherche d'abord à faire du bon vin en intégrant la variabilité biologique de la matière première, les interactions entre les différents paramètres de la vinification et des contraintes sanitaires ou externes de type climatique. C'est dans ce contexte, qu'il souhaite obtenir des vins plus colorés.

Les approches visant à évaluer l'effet particulier d'une variable sur la couleur n'ont pas donné de résultats satisfaisants du fait de nombreuses interactions non maîtrisées. La simulation globale, à partir d'un ensemble d'exemples, est susceptible d'aider à la production de règles de vinification permettant d'améliorer la couleur du vin. La logique floue devrait permettre de gérer l'effet millésime, la variabilité de la matière première et les variations des conditions extérieures notamment. Elle peut aussi rendre compte du caractère incertain de la connaissance.

Le projet, qui se déroule sur une période de deux ans, a démarré en septembre 2000. Ce chapitre relate l'état d'avancement des travaux à mi-contrat. Au cours de cette première année nous nous

[a]Les termes techniques de vinification sont décrits dans l'annexe B.

sommes attachés à la réalisation d'outils facilitant la collecte des données, numériques et qualitatives, ainsi qu'à l'induction d'un premier jeu de règles.

Le simulateur est un outil d'aide à la décision, principalement destiné à l'œnologue, qui doit permettre de choisir certains paramètres de vinification en fonction des conditions initiales de la vendange, notamment le cépage, et de l'objectif à atteindre pour la couleur, ou plus généralement pour la qualité. Il inclura, à terme, des règles expertes, formulées par l'œnologue, et des règles induites, issues de l'apprentissage à partir de données.

La partie présentée dans ce mémoire concerne la mise en forme des données et l'induction de règles qui décrivent les relations entre les variables d'état de la vendange, les opérations de conduite de la vinification et la couleur du vin.

La première étape de la conception du simulateur, présentée dans la section 5.1, consiste en la mise en forme des données. Deux types de règles seront induites à partir des données. La section 5.2 introduit le principe de la construction d'un arbre de décision binaire, une méthode de référence pour l'induction de règles, et analyse les règles induites par cette méthode à partir de nos données. L'application de la méthode présentée dans les deux chapitres précédents, qui conduit à des règles floues interprétables, est traitée dans la section 5.3. La section 5.4 compare les résultats obtenus par ces deux méthodes et propose des perspectives pour la gestion de l'application.

5.1 Mise en forme des données

La mise en forme des données est rendue indispensable par plusieurs facteurs. Elles sont hétérogènes, de nature très différente, aussi bien qualitatives que numériques. Certaines, comme les relevés de température ou de densité, doivent être condensées. Du fait du mode de collecte, elles sont susceptibles de contenir des incohérences. Enfin, toutes ne sont pas pertinentes pour essayer d'améliorer la couleur du vin.

5.1.1 Les données brutes

Trois types de données sont disponibles :
- Les données initiales sur la vendange : il s'agit principalement du cépage. L'indice réfractométrique, mesuré lors de l'apport, indique le potentiel en alcool. Nous ne l'avons pas utilisé pour la couleur.
- Les courbes d'évolution de la densité et de la température : un traitement est nécessaire pour en extraire des paramètres caractéristiques.
- Les actions externes effectuées sur la cuve durant la fermentation.
- Le résultat des analyses, dont, notamment, l'intensité colorante qui quantifie la couleur.

Le nombre de combinaisons de cépages utilisées durant cette campagne de mesure, 21, est très important, surtout lorsqu'il est mis en relation avec le nombre de vinifications dont nous disposons, 30. Nous avons décidé de regrouper les cépages suivant leur potentiel de couleur, qui est disponible dans la littérature. L'ordre décroissant du tableau 5.1 est proposé par l'œnologue de la cave d'Arzens, Bruno Mailliard.

Les actions externes effectuées sur la cuve sont saisies dans une base de données à l'aide d'un formulaire développé dans le cadre d'une application client serveur utilisable via un navigateur internet standard. Nous présentons les principales opérations. L'ajout d'anti-oxydant est indispensable pour garantir l'état sanitaire du moût. L'apport de moût concentré rectifié permet d'augmenter le degré d'alcool final et favorise l'extraction de tanins. L'insertion de levures facilite le démarrage de

Numéro d'ordre	Cépages
1	Alicante
2	Sélections Cabardès et Malepère
3	COT, Syrah, CMS (Cot, Merlot, Syrah)
4	Cabernet Sauvignon et Cabernet franc
5	Caladoc, Egiodola et Métis
6	Merlots et Merlot, Chenanson, Egiodola
7	Chenanson, Portan, Portan Terra Vitis, Grenache
8	Muscat rosé

TAB. 5.1 – Les cépages groupés et ordonnés par potentiel de couleur décroissant

la fermentation. Ces actions ne sont pas prises en compte pour la conception du simulateur car leur effet sur la couleur est supposé faible.

Les opérations qui facilitent l'extraction sont essentielles pour notre simulateur. Elles représentent des variables de conduite importantes. La première d'entre elles est l'enzymage : l'action est caractérisée par un équivalent dose d'un enzyme de qualité standard. Les autres sont celles qui facilitent les échanges entre le jus et le chapeau. Une variable binaire indique si la cuve est équipée d'un système de remontage ou de pigeage. Le nombre de délestages effectués au cours de la vinification est comptabilisé.

Les enregistrements de température sont dépouillés par phases, les limites des phases étant déterminées par l'étude de la courbe d'évolution de la densité.

5.1.2 Traitement de la courbe de densité et variables induites

Pour piloter la fermentation, l'œnologue utilise principalement l'évolution des courbes de densité et de température. La cave d'Arzens est bien instrumentée, la température est enregistrée au rythme d'une mesure toutes les quatre heures, sur chacune des cuves. La densité est relevée manuellement quotidiennement ou presque.

La courbe d'évolution de la densité est le principal outil de suivi de la fermentation utilisé par l'œnologue. Elle est découpée comme indiqué sur la figure 5.1. Deux repères temporels sont utilisables. Dans le premier, la fermentation est découpée en trois phases : la phase de fermentation proprement dite est précédée d'une phase pré-fermentaire et suivie d'une phase post-fermentaire. Les limites entre ces phases sont déterminées par la valeur de la densité. La phase de fermentation commence le jour où la densité diminue notablement, la valeur seuil est fixée à 10 points. Elle se termine lorsque la densité atteint une valeur limite fixée à 997. La phase de fermentation est elle-même partagée par la limite, fixée à 1050, qui représente, par convention, la transition entre les phases aqueuse et alcoolique. Ces durées sont exprimées en jours et sont respectivement notées : $d1$, $d2$, $d3$, dal et daq. La durée $d2$ est partagée en deux : $dal - d1$ et $daq - d3$. Nous calculons également la pente moyenne au cours de la phase de fermentation, notée simplement *pente*.

Un travail préliminaire a permis de déterminer un ensemble de variables pertinentes, à extraire des relevés de température, qui sont caractéristiques des trois phases de fermentation (voir la figure 5.1) :

- $Moy1$: moyenne de la phase pré-fermentaire
- $Moy2\_1$: moyenne de la première partie de la phase de fermentation
- $Max2\_1$: maximum de la première partie de la phase de fermentation

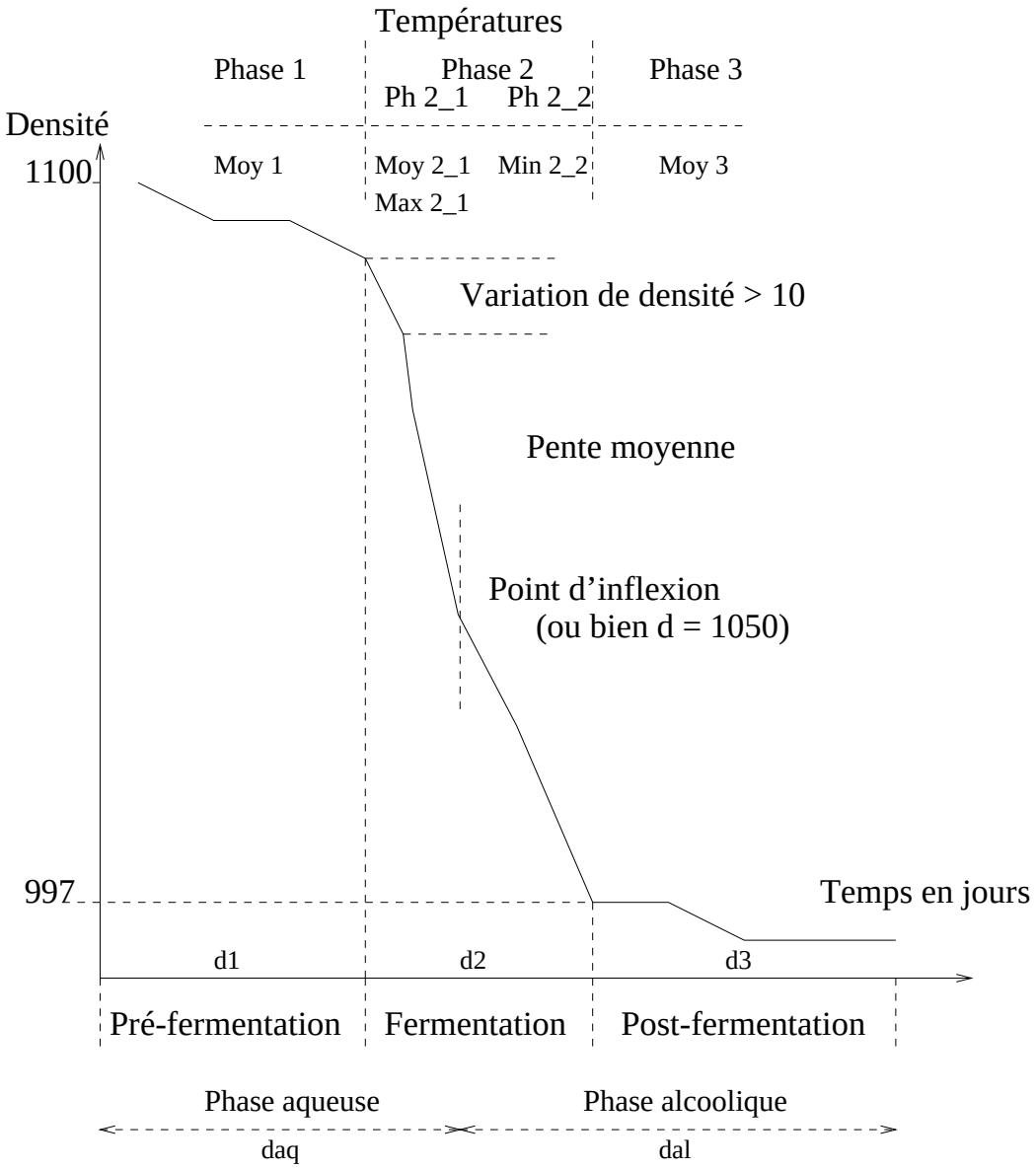


FIG. 5.1 – Découpage en zones de la courbe d'évolution de la densité

- $Min2\_2$: température minimale atteinte au cours de la deuxième partie de la phase de fermentation
- $Moy3$: moyenne de la phase post-fermentaire

5.1.3 Notre échantillon

Les analyses relatives à la couleur et aux tanins sont effectuées sur un produit stable, après un séjour en cuve de vieillissement. Les cuves de vieillissement, d'un volume plus important que les cuves de fermentation, peuvent donc contenir des mélanges. Pour la conception du simulateur, nous avons décidé de travailler avec les cuves pures, les seules qui permettent de relier avec certitude les paramètres de vinification à l'intensité colorante. Les mouvements de cuve seront saisis grâce au logiciel en cours de développement, ce qui permettra d'assurer la traçabilité d'un lot de vendanges, depuis l'apport sur les quais de réception, jusqu'à la livraison chez le client.

Nous disposons, après la première campagne de mesures effectuée pendant les vendanges 2000, d'un ensemble de trente vinifications. Nous devrions rapidement, après les vendanges 2001, disposer d'une base de données plus importante. Ces trente vinifications sont décrites, et classées suivant le potentiel de couleur de leur cépage, dans le tableau 5.2. Les variables relatives aux opérations sont Ez pour la dose d'enzymage, Rm pour le remontage, valeur 1, ou le pigeage, valeur 2, Dl pour le nombre de délestages et Tn pour l'apport de tanins exogènes.

La figure 5.2 montre les histogrammes des principales variables d'entrée.

L'histogramme de la variable de sortie, l'intensité colorante, est tracé sur la figure 5.3, ainsi que le découpage en cinq sous-ensembles flous effectué à l'aide d'une grille régulière.

	d1	d2	d3	daq	dal	daq-d1	dal-d3	pente	Moy1	Moy2_1	Max2_1	Min2_2	Moy3	cépage	Ez	Rm	Dl	Tn	Ic
1	2	7	3	5	7	3	4	-12.9	16.4	23.6	28.7	20.4	23.4	2	0	1	0	0	7.33
2	2	5	1	5	3	3	2	-18.8	19.6	26.2	30.9	24.7	20.9	2	2	1	0	0	8.28
3	1	6	0	4	3	3	3	-13.7	19.2	23.9	28.9	25.2	16	2	0	1	0	0	7.41
4	2	5	1	4	4	2	3	-14.6	20.3	27.3	28.8	20.9	23.7	2	2	1	0	0	9.82
5	2	14	4	5	15	3	11	-7.1	22.4	31.3	33.9	20	23.6	2	0	2	0	0	7.6
6	2	9	3	4	10	2	7	-10.7	18.5	25	28.8	20.5	23.6	2	2	2	0	0	8.61
7	2	9	0	6	5	4	5	-10.2	18.1	27.3	29.9	25.1	13	2	3	2	4	0	7.7
8	1	5	0	3	3	2	3	-17	17.5	25.3	29	13.9	13.9	3	2	1	0	0	7.12
9	2	4	0	3	3	1	3	-17.8	20.5	28.2	28.8	10.6	10.6	3	2	1	0	0	6.73
10	4	7	0	7	4	3	4	-13.1	16.8	22.8	28	13.1	13.1	3	3.5	2	1	1	7.22
11	2	7	0	5	4	3	4	-11.9	14.2	23.9	27.8	26	11.5	4	2	1	0	1	6.92
12	1	10	0	3	8	2	8	-7.9	18.5	26.3	29	18.9	17.5	4	0	1	0	0	8.38
13	1	5	1	3	4	2	3	-16.6	18.5	27.4	29	24.8	21.3	4	0	1	0	0	9.71
14	1	10	0	5	6	4	6	-9.1	13.7	15.3	22.7	23.7	10.1	4	0.8	2	0	0	6.25
15	2	7	6	7	8	5	2	-12.7	19.9	24.9	28.8	26.2	23.8	4	2	2	0	0	6.46
16	1	7	0	5	3	4	3	-14	18.8	26.8	28.9	27.2	9.9	4	2	2	0	0	5.58
17	2	4	0	4	2	2	2	-20.8	18.7	26.6	28.5	10.4	10.4	5	0	1	0	0	11.23
18	1	4	1	3	3	2	2	-23.8	23.8	28.3	30.2	26.7	23	6	2	1	0	0	7.95
19	1	9	0	4	6	3	6	-8.9	18.9	24.2	27.7	22.8	18.2	6	0	1	0	0	4.3
20	1	6	0	4	3	3	3	-13.5	19.6	29.3	33.1	22.7	19.6	6	0.8	1	0	0	6.35
21	2	6	0	5	3	3	3	-14	23.5	28.6	32.3	25	14.8	6	0	1	0	0	8.21
22	1	5	0	3	3	2	3	-19.8	16.9	24	27	16.8	16.8	6	0	1	0	1	7.79
23	2	5	0	3	4	1	4	-15.4	21.5	28.6	29	25.1	16.1	6	0	1	0	0	6.99
24	1	7	0	4	4	3	4	-12.1	19	27.4	29	23	16.3	6	0	1	0	0	6.78
25	1	13	5	2	17	1	12	-4.5	18.6	22.1	23.8	19.4	23	6	0	2	0	0	5.6
26	3	6	4	8	5	5	1	-15.8	20.2	26.1	29	26.2	23.5	6	2	2	0	0	5.29
27	1	7	8	4	12	3	4	-13.6	25	28.1	29	24.3	20.9	6	0	2	0	0	5.72
28	3	8	0	5	6	2	6	-12.2	19	24.4	28.4	24.1	21.4	6	2	2	1	0	9.32
29	3	8	0	4	7	1	7	-10.9	21.7	23.8	24	19.1	19.1	6	2	2	0	1	7.18
30	1	4	0	2	3	1	3	-21.8	23	25.4	26.6	23.6	20.8	7	3.5	1	0	1	9.25

TAB. 5.2 – Les données disponibles pour la conception du simulateur

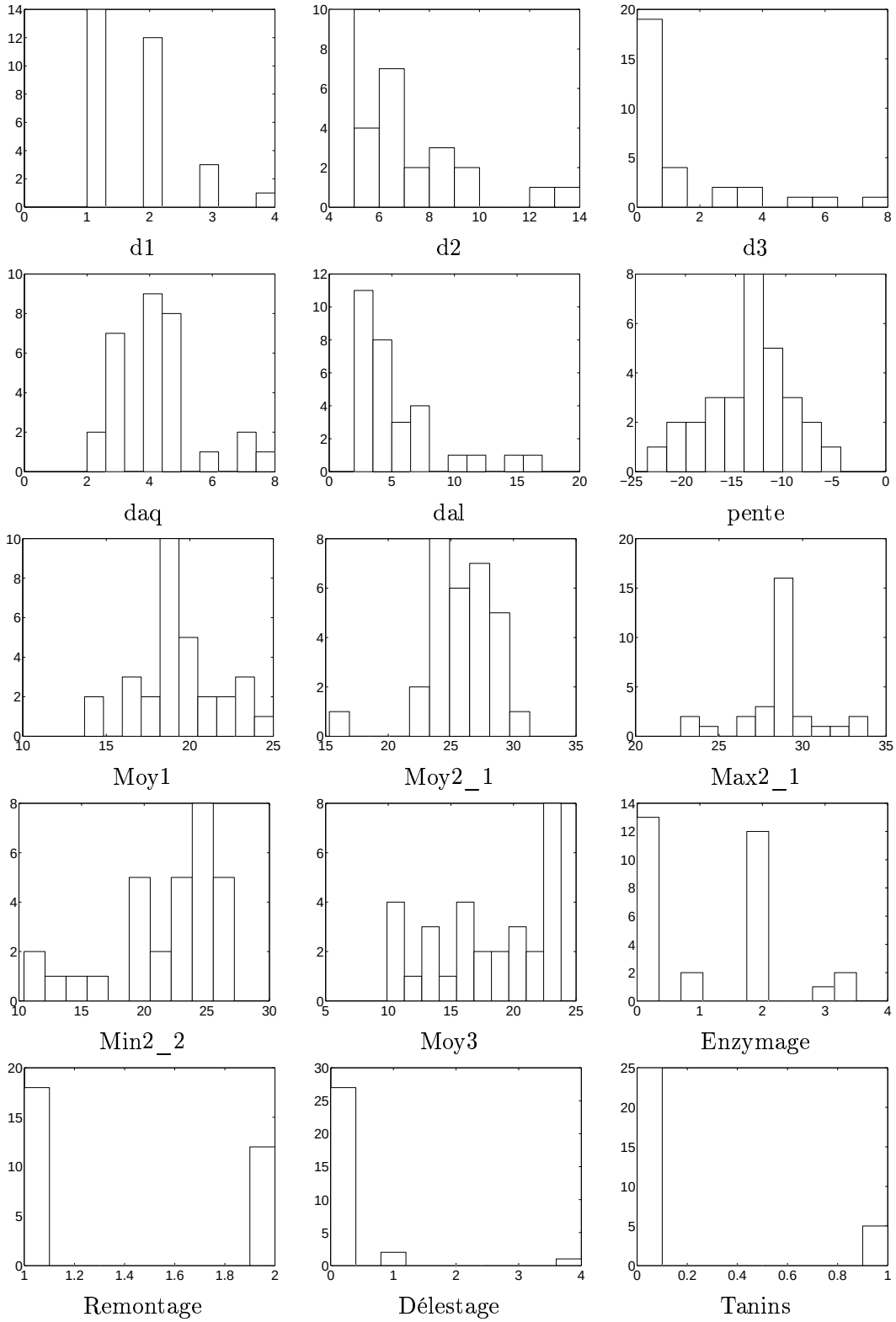


FIG. 5.2 – Histogrammes des principales variables d'entrée

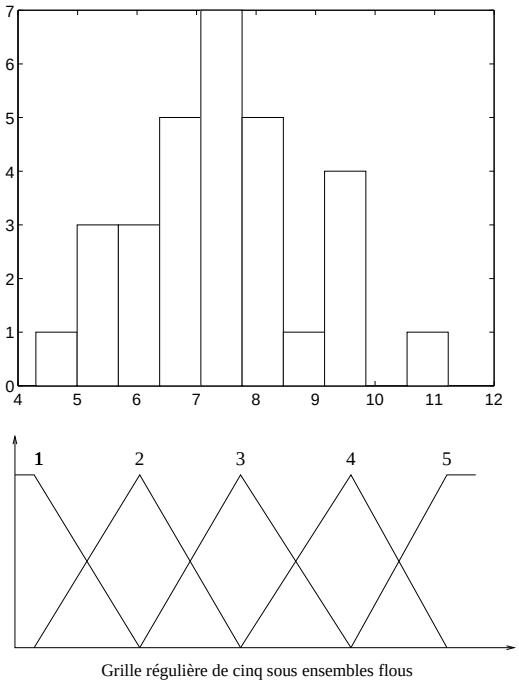


FIG. 5.3 – Histogramme et partition de l'intensité colorante

5.1.4 Analyse des relations linéaires

Les coefficients des corrélations linéaires entre les différentes variables d'entrée et de sortie figurent dans le tableau 5.3.

Le cépage étant une variable qualitative, et bien qu'il ait une influence sur la couleur, sa corrélation avec les autres variables n'a aucun sens et n'est pas indiquée dans ce tableau.

Des corrélations significatives, indiquées en gras dans le tableau 5.3, sont observables entre certaines variables d'entrée, notamment les durées, et la pente moyenne de fermentation. Si l'on ne tient pas compte des durées complémentaires $daq - d1$ et $dal - d3$, trois variables restent liées : $d2$, dal et $pente$. D'autre part les deux températures relatives à la première partie de la phase de fermentation sont très liées : $Moy2\_1$ et $Max2\_1$. L'intensité colorante n'est corrélée à aucune variable d'entrée.

Cette recherche de relations linéaires est complétée par une analyse en composantes principales. Nous aurions pu espérer qu'elle conduise à un espace factoriel au sein duquel les vinifications dont les intensités colorantes sont proches, soient regroupées, même en ensembles non convexes. Malheureusement, nous n'avons rien observé d'utile.

5.1.5 Les ensembles d'apprentissage

Les données sont exploitées de diverses façons. Deux types de règles sont induites. Les règles générées par la méthode présentée dans ce mémoire pourront être comparées à celles produites par un arbre de décision classique. Cette technique, dont les caractéristiques et les limites sont bien connues, présente l'avantage d'être disponible dans plusieurs bibliothèques statistiques.

Nous avons signalé que deux repères temporels étaient utilisables. Le premier est celui des trois phases de fermentation, le second est celui des deux phases d'extraction. Les variables de durée

	d1	d2	d3	daq	dal	daq -d1	dal -d3	pente	Moy1	Moy2_1	Max2_1	Min2_2	Moy3	Ez	Rm	Dl	Tn
d1	1.00	0.01	0.00	0.63	-0.02	0.09	-0.03	0.06	-0.03	-0.03	0.02	-0.21	0.04	0.46	0.40	0.27	0.29
d2	0.01	1.00	0.34	0.13	**0.83**	0.16	**0.91**	**0.90**	-0.18	-0.23	-0.14	0.06	0.17	-0.23	0.59	0.17	-0.14
d3	0.00	0.34	1.00	0.21	0.73	0.27	0.21	0.25	0.32	0.09	0.05	0.17	0.60	-0.16	0.49	-0.16	-0.26
dac	0.63	0.13	0.21	1.00	-0.04	**0.83**	-0.23	0.19	-0.19	-0.11	0.20	0.24	-0.01	0.32	0.51	0.33	-0.03
dal	-0.02	**0.83**	0.73	-0.04	1.00	-0.03	**0.82**	0.70	0.13	-0.07	-0.14	0.01	0.48	-0.28	0.58	-0.04	-0.17
dac-d1	0.09	0.16	0.27	**0.83**	-0.03	1.00	-0.28	0.20	-0.23	-0.11	0.24	0.46	-0.05	0.08	0.36	0.22	-0.25
dal-d3	-0.03	**0.91**	0.21	-0.23	**0.82**	-0.28	1.00	**0.79**	-0.07	-0.17	-0.24	-0.14	0.19	-0.26	0.42	0.07	-0.03
pente	0.06	**0.90**	0.25	0.19	0.70	0.20	**0.79**	1.00	-0.28	-0.27	-0.18	0.09	0.07	-0.25	0.50	0.17	-0.17
Moy1	-0.03	-0.18	0.32	-0.19	0.13	-0.23	-0.07	-0.28	1.00	0.67	0.39	0.19	0.41	-0.03	-0.01	-0.14	-0.16
Moy2_1	-0.03	-0.23	0.09	-0.11	-0.07	-0.11	-0.17	-0.27	0.67	1.00	**0.82**	0.06	0.21	-0.10	-0.26	0.03	-0.28
Max2_1	0.02	-0.14	0.05	0.20	-0.14	0.24	-0.24	-0.18	0.39	**0.82**	1.00	0.14	0.17	-0.10	-0.25	0.08	-0.38
Min2_2	-0.21	0.06	0.17	0.24	0.01	0.46	-0.14	0.09	0.19	0.06	0.14	1.00	0.30	0.01	0.13	0.07	-0.19
Moy3	0.04	0.17	0.60	-0.01	0.48	-0.05	0.19	0.07	0.41	0.21	0.17	0.30	1.00	-0.06	0.13	-0.20	-0.17
Ez	0.46	-0.23	-0.16	0.32	-0.28	0.08	-0.26	-0.25	-0.03	-0.10	-0.10	0.01	-0.06	1.00	0.30	0.40	0.39
Rm	0.40	0.59	0.49	0.51	0.58	0.36	0.42	0.50	-0.01	-0.26	-0.25	0.13	0.13	0.30	1.00	0.33	0.00
Dl	0.27	0.17	-0.16	0.33	-0.04	0.22	0.07	0.17	-0.14	0.03	0.08	0.07	-0.20	0.40	0.33	1.00	0.00
Tn	0.29	-0.14	-0.26	-0.03	-0.17	-0.25	-0.03	-0.17	-0.16	-0.28	-0.38	-0.19	-0.17	0.39	0.00	0.00	1.00
Ic	0.14	-0.33	-0.29	-0.23	-0.28	-0.40	-0.15	-0.42	0.06	0.19	0.16	-0.25	0.07	0.09	-0.31	0.08	0.07

TAB. 5.3 – Matrice des corrélations des données disponibles pour la conception du simulateur

utilisées dans le premier repère sont : $d1$, $d2$ et $d3$. Celles utilisées dans le second sont daq et dal. Les variables secondaires, $dal - d1$ et $daq - d3$, qui conduisent à un découpage de la durée de fermentation incompatible avec notre volume de données, ne sont pas mobilisées pour l'instant.

Pour chacun de ces deux repères, nous travaillerons avec l'ensemble des autres variables disponibles.

Les variables binaires ne doivent pas être manipulées comme les autres avec la logique floue. En effet, elles interdisent toute interpolation comme nous le verrons dans la suite de ce chapitre. Les variables enzymage et délestage, dont les distributions sont quasiment discrètes dans notre base de données, ne sont pas intrinsèquement binaires. Leur distribution devrait être davantage continue pour une base de données plus importante.

Comme le potentiel de couleur dépend des cépages, les effets des pratiques de vinification ne sont réellement observables qu'au sein d'un même groupe de cépages. En effet, regrouper des vinifications de différents cépages suivant leur intensité colorante peut conduire à assimiler des pratiques qui ont permis d'élever sensiblement l'intensité colorante de cépages dont le potentiel de couleur est faible à des pratiques qui n'ont pas du tout valorisé le potentiel de couleur d'autres cépages. A terme, lorsque les données collectées le permettront, les règles seront induites à partir de vinifications réalisées avec des cépages homogènes.

Cette approche est proposée avec le cépage majoritaire dans notre base d'apprentissage, le Merlot. Comme seulement douze vinifications sont concernées, les règles seront induites par le seul arbre de décision, à titre purement informatif.

5.2 Induction de règles par un arbre de décision binaire

5.2.1 Principe de la méthode

Les arbres de décision, encore appelés arbres de segmentation, visent à produire une partition des exemples suivant leur proximité dans le domaine de la variable de sortie. Leur extension floue a été présentée dans la section 2.1.4. Comme leurs homologues flous ils sont utilisés pour induire des règles de décision à partir d'un ensemble d'exemples.

La structure de l'arbre est définie par les nœuds, les branches et les feuilles qui sont les nœuds terminaux. Le premier nœud s'appelle la racine. Les arbres de décision sont des arbres binaires, c'est-à-dire qu'il y deux branches, et seulement deux, issues d'un nœud. L'arbre correspond à un cheminement partant de la racine et conduisant à une feuille. Un nœud correspond à une question sur la valeur d'un descripteur, par exemple : est-ce que la température est inférieure à 25 ^oC ? Si la réponse, pour un individu donné, est oui alors l'individu empruntera la branche de gauche, dans le cas contraire il empruntera la branche de droite. L'apprentissage permet de définir la structure de l'arbre. A chacune des feuilles sera associée une étiquette : la classe majoritaire parmi les individus qui composent la feuille en classification, ou bien la moyenne des valeurs des sortie de ces individus lors qu'on travaille en régression. L'arbre peut être ensuite utilisé pour traiter des individus inconnus. L'individu entre par la racine, il subit les différents tests, ou questions, conduisant à une feuille. La sortie pour cet individu sera l'étiquette de la feuille correspondante. A chaque feuille correspond une règle de décision.

La construction de l'arbre se réalise en deux étapes : il faut choisir les questions associées aux nœuds, et répéter la procédure jusqu'à l'obtention de feuilles les plus homogènes possibles. En classification, une feuille sera dite pure si tous les individus de la base d'apprentissage qui

arrivent dans cette feuille appartiennent à la même classe. En régression, l'homogénéité de la feuille est mesurée par l'écart type de la sortie des exemples attirés par la feuille. Puis dans un deuxième temps, pour éviter de construire un arbre trop spécifique, trop lié aux seules données d'apprentissage et qui présentera de mauvaises performances sur d'autres données, on procède à un élagage. Cette procédure élimine des sous-arbres et conduit à des feuilles qui peuvent être non homogènes. En contrepartie, les règles de décision sont moins nombreuses et plus générales.

Il y a deux paramètres à définir pour une question : sur quel descripteur porte la question et, quel est la valeur de seuil ? Pour des descripteurs continus, comme une température par exemple, il y a une infinité de valeurs candidates possibles. Mais si nous considérons qu'un nœud, à un moment donné de l'apprentissage, contient un nombre fini d'éléments, n, le nombre de questions pertinentes possibles pour séparer l'effectif en deux classes est fini et égal à $n - 1$.

Le descripteur et la valeur seuil choisis sont ceux qui conduisent, à chaque itération, à l'ensemble des feuilles les plus homogènes.

La procédure d'élagage s'effectue habituellement avec un échantillon test, différent de l'ensemble utilisé pour l'apprentissage. Elle permet d'évaluer une famille d'arbres de plus en plus petits, on choisira parmi ceux-ci celui qui minimise un critère d'évaluation qui réalise un compromis entre la complexité et la performance.

La limite de cette technique est bien connue : les seuils sont des valeurs nettes et l'arbre résultant est assez sensible à de petites variations dans les données d'apprentissage. La version floue, en cours de développement dans notre environnement de travail, permet de s'affranchir de cet inconvénient.

Nous avons utilisé l'implémentation des arbres du logiciel libre R, qui fait partie du projet GNU (http ://www.R-project.org/).

5.2.2 Arbres calculés à partir de l'ensemble des cépages

Les trente vinifications disponibles sont étudiées dans les deux repères temporels envisagés, celui des trois phases de fermentation et celui des deux phases d'extraction.

Dans le repère des trois phases de fermentation

L'arbre élagué est représenté dans la figure 5.4.

Pour chacune des feuilles nous avons indiqué l'intensité colorante moyenne de la feuille, une étiquette correspondant à un label linguistique $faible$, *moyenne* ou *leve*, ainsi que le nombre d'individus dans la feuille et le nombre d'individus dont la valeur est hors des bornes correspondant à l'étiquette de la feuille.

Ainsi, sur la figure 5.4, la feuille dont la moyenne est 6 porte le label $faible$. La séquence H : 2 indique que deux individus présentent une intensité colorante supérieure à 6.5, valeur limite pour la classe $faible$. Ces deux intensités appartiennent à la classe moyenne. L'individu hors limite de la feuille 6.2 présente également une intensité moyenne. La feuille dont la moyenne vaut 8.5 correspond donc au label *leve*, elle contient un individu dont l'intensité colorante est moyenne.

Dans le repère des deux phases d'extraction

Dans le repère des deux phases d'extraction, figure 5.5, deux feuilles sont hétérogènes : la feuille d'étiquette $faible$ contient un individu d'intensité colorante *moyenne*, celle d'étiquette *moyenne* en contient deux de classe *leve*.

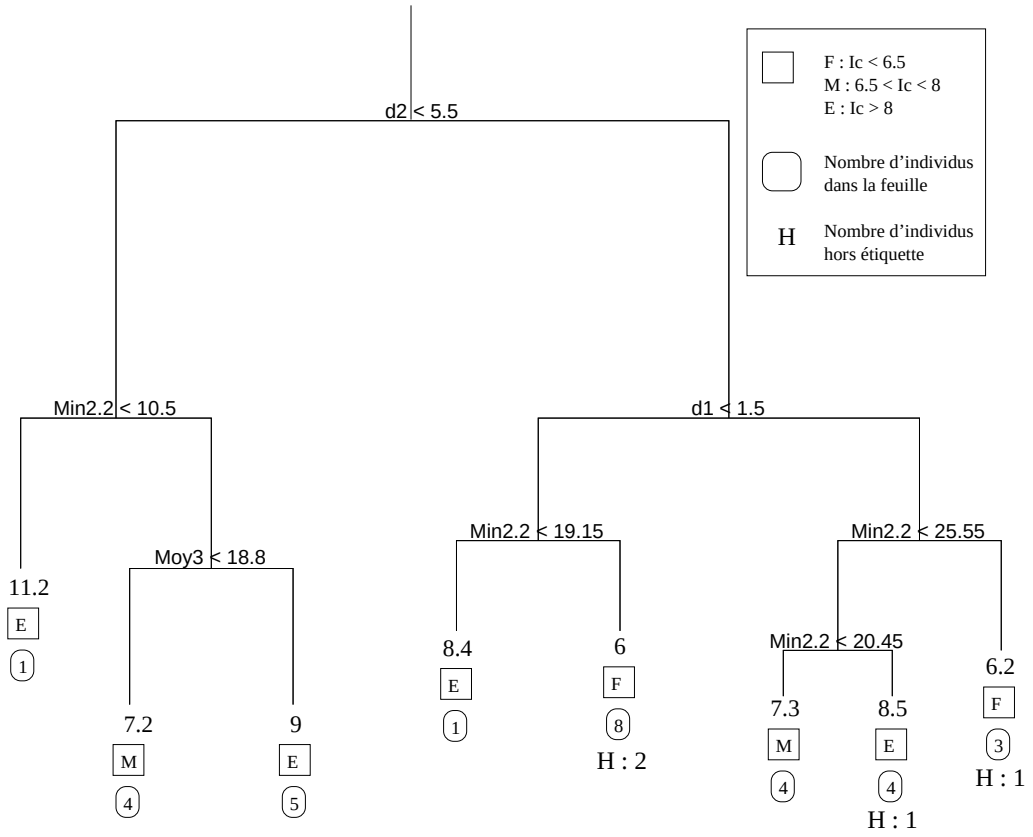


FIG. 5.4 – Arbre de décision dans le repère des trois phases de fermentation pour l'ensemble des cépages

Discussion sur ces deux représentations

Les limites des classes d'intensité colorante, faible si inférieure à 6.5 et élevée si supérieure à 8, peuvent prêter à discussion. Il semble que les vinifications dont nous disposons correspondent à des cépages sélectionnés. Les bornes, 5.5 voire 6 pour la limite faible, et, 7 pour la limite élevée, seraient plus représentatives de l'ensemble de la production. Les valeurs des bornes ont des conséquences évidentes sur l'homogénéité des feuilles.

Parmi les deux repères envisagés, celui des trois phases de fermentation est plus familier à l'œnologue, bien que la limite entre les phases aqueuse et alcoolique soit importante pour la conduite. C'est durant la phase aqueuse que se font les ajouts d'oxygène et d'azote. De même, pour éviter l'extraction de tanins trop grossiers, les délestages sont évités en phase alcoolique.

Interprétation des règles

Le faible nombre de données dont nous disposons ne nous permet pas de conclure sur la validité des règles induites. Nous nous sommes attachés à faciliter leur interprétation et leur analyse par l'œnologue.

La mise en évidence du rôle de la vitesse de fermentation, *pente* ou $d2$, a surpris l'œnologue. L'objectif de la phase de fermentation alcoolique est d'épuiser les sucres, le rapport entre la vitesse de fermentation et la couleur n'est pas expliqué. Il se trouve que ce sont ces variables qui sont le plus corrélées avec la sortie. Il est possible que ce fait résulte d'une spécificité de cette modeste base

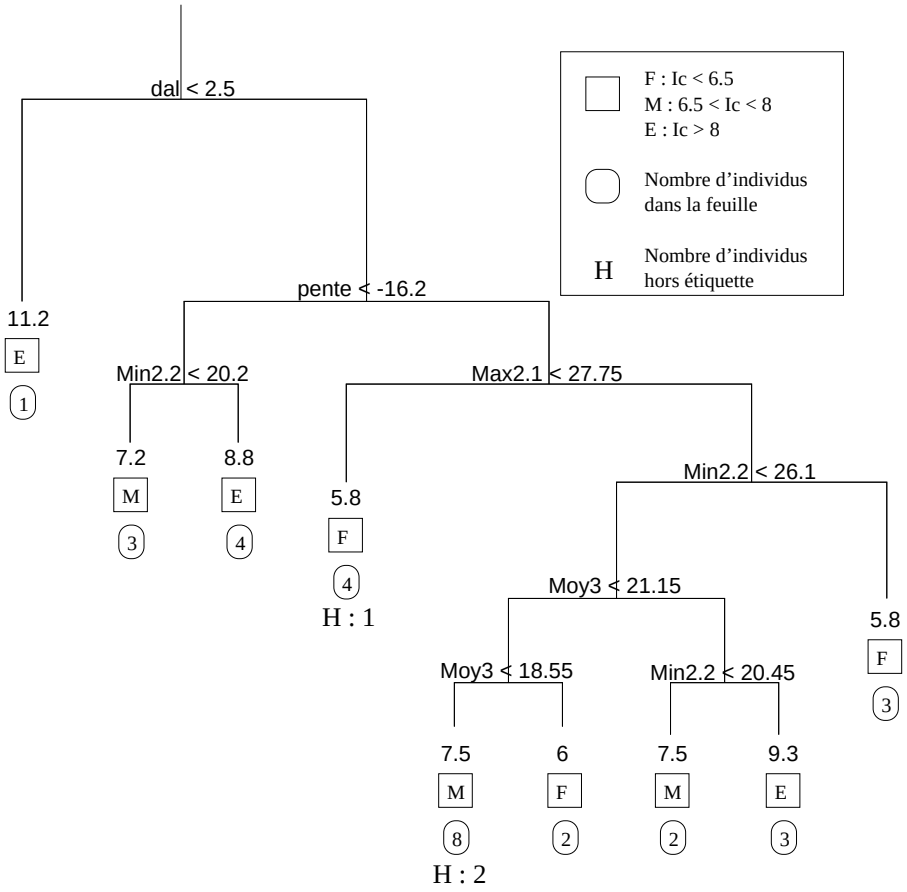


FIG. 5.5 – Arbre de décision dans le repère des deux phases d'extraction pour l'ensemble des cépages

de données. Les autres variables sélectionnées sont principalement des températures. Les limites affichées permettent de retrouver dans l'arbre le schéma d'une courbe de fermentation typique. La variable $Max2\_1$ représente la température maximale de fermentation, car au delà d'une densité voisine de 1050 la température commence à baisser. L'arbre du repère des deux phases d'extraction indique qu'il est préférable qu'elle soit supérieure à 27 ^oC. Une température maximale inférieure à 28 ^oC peut être la conséquence d'une action de refroidissement ou bien traduire une fermentation lente. $Min2\_2$ correspond à la température de fin de fermentation, d'après l'arbre du repère des deux phases d'extraction, elle doit être comprise entre 20 et 26 ^oC si l'on vise une intensité colorante élevée. Si elle descend au dessous de 20 ^oC cela signifie que les levures ont moins d'activité, que la fermentation alcoolique est en train de s'arrêter et que la fermentation malolactique aura de la difficulté à démarrer. Enfin, la température moyenne de la phase post fermentaire doit être supérieure à un seuil d'environ 21 ^oC, pour favoriser les processus d'extraction. Schématiquement, la courbe de température type, illustrée sur la figure 5.6, est constituée d'une montée en température jusqu'à 28 ou 30 ^oC pour atteindre une densité voisine de 1050, puis une phase de descente jusqu'à 24 ^oC pour atteindre la densité limite de 997, et enfin une stabilisation autour de cette même valeur.

L'arbre binaire, construit à partir de l'ensemble des cépages, a permis de retrouver les principaux phénomènes connus relatifs à la vinification. Nous pouvons également remarquer que, parmi les variables sélectionnées, aucune n'est relative à une action directe de l'œnologue sur la cuve.

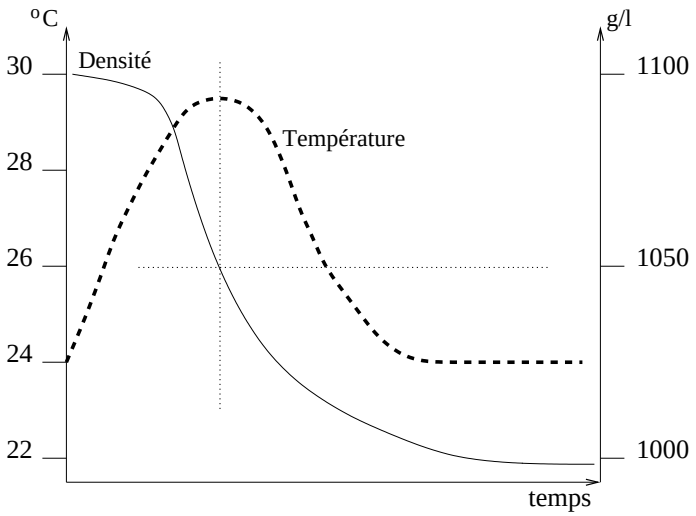


FIG. 5.6 – Une courbe de fermentation type

5.2.3 Arbres calculés avec le groupe des cépages Merlot

Les effets des pratiques de vinification sur la couleur ne peuvent être comparés qu'au sein d'un groupe de cépages homogènes. Cette approche est proposée avec le cépage majoritaire dans notre base d'apprentissage, le Merlot. Douze vinifications sont concernées.

Ce cépage présente un potentiel de couleur modeste, une intensité colorante de 7 peut être considérée comme élevée dans ce contexte. Les arbres sont donnés dans les figures 5.7 et 5.8. La représentation dans le repère des trois phases de fermentation est intéressante car l'arbre ne comporte que trois feuilles. Le seuil de pente sélectionné, -9.9, correspond, pour la feuille faible, à des fermentations très lentes. Pour un vin rouge la pente ne devrait pas être supérieure à -15, alors que des valeurs comprises entre -10 et -5 sont parfaitement acceptables pour un vin blanc. Si la variable *pente* n'est pas prise en compte, elle est, comme attendu, remplacée par $d2$. L'autre feuille faible provient d'une durée de macération excessive pour ce cépage. L'itération suivante, non représentée sur le graphe, provoque l'éclatement du nœud dont l'étiquette est *leve*. La variable sélectionnée est *dlestage* : l'usage de délestages conduit à élever l'intensité colorante.

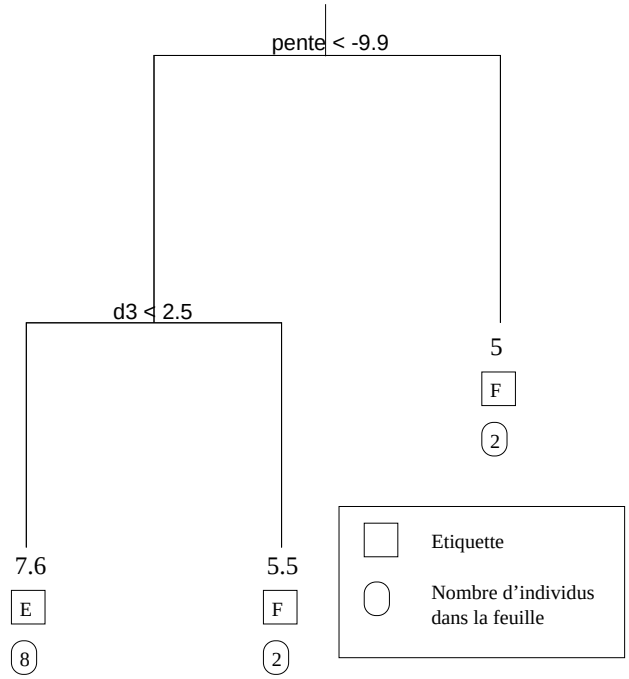


FIG. 5.7 – Arbre de décision dans le repère des trois phases de fermentation pour le groupe des cépages Merlot

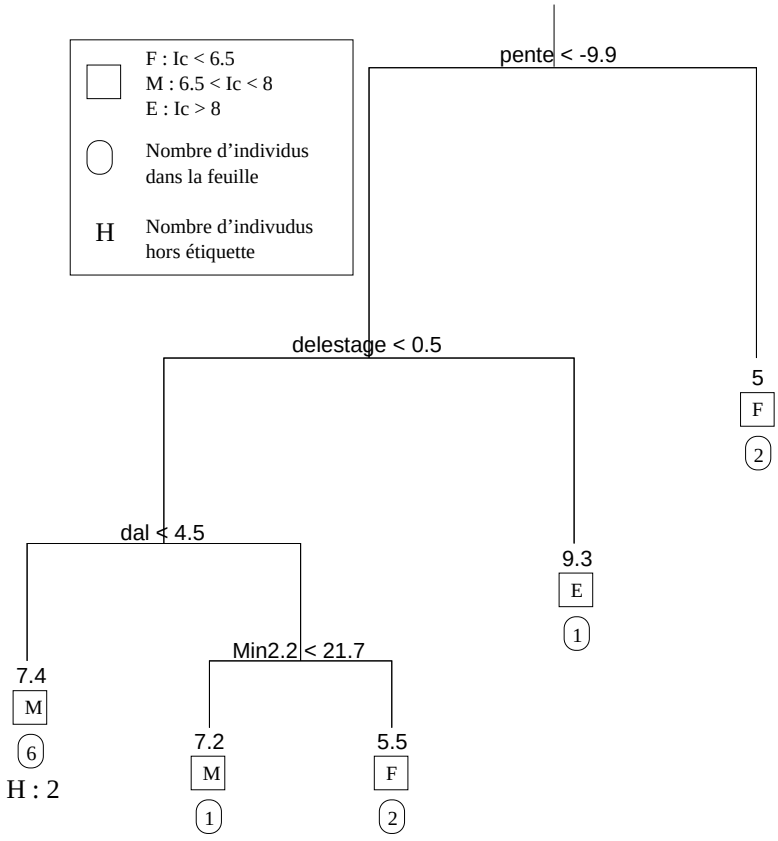


FIG. 5.8 – Arbre de décision dans le repère des deux phases d'extraction pour le groupe des cépages Merlot

5.3 Induction des règles floues

La méthode d'induction de règles floues présentée dans les deux chapitres précédents requiert plusieurs paramètres.

Pour la génération de la famille de partitions monodimensionnelles, la tolérance d'égalité sur les valeurs uniques est fixée à un pour mille, ainsi toutes les valeurs distinctes sont conservées. La distance est numérique, calculée sur la totalité de la partition. Aucune fusion n'est pénalisée.

Les paramètres de la sélection et de la simplification sont choisis en fonction de l'expérience acquise sur les données du riz et pour tenir compte de la faible cardinalité de l'ensemble d'apprentissage : l'effectif minimum pour initialiser une règle est fixé à un, le degré de vérité minimum requis est de 0.3. La contrainte des exemples inactifs est complètement relâchée.

Comme pour l'arbre de décision, les deux repères temporels envisagés, celui des trois phases de fermentation, et celui des deux phases d'extraction, sont étudiés de manière indépendante.

5.3.1 Avec l'algorithme de la composition multidimensionnelle

Nous nous plaçons dans le repère des trois phases de fermentation. Les variables utilisées sont présentées dans le tableau 5.4. Pour chacune d'entre elles nous avons indiqué les bornes du domaine de variation ainsi que le nombre de valeurs distinctes, *Nb uniques*, qui représente le nombre maximal de sous-ensembles flous dans la partition.

Num	Variable	Val Inf	Val Sup	Nb uniques
1	d1	1	4	4
2	d2	4	14	9
3	d3	0	8	7
4	pente	-23.8	-4.5	29
5	Moy1	13.7	25	26
6	Moy2_1	15.3	31.3	26
7	Max2_1	22.7	33.9	20
8	Min2_2	10.4	27.2	28
9	Moy3	9.9	23.8	27
10	enzymage	0	3.5	5
11	remontage	1	2	2
12	délestage	0	4	3
13	tanins	0	1	2
	Ic	4.3	11.3	30

TAB. 5.4 – Les variables utilisées dans le repère des trois phases de fermentation

Les quatre premières variables, les trois durées et la pente moyenne pendant la phase de fermentation, sont relatives à la cinétique de fermentation. Les cinq suivantes sont des températures correspondant à ces phases. Enfin, les quatre dernières quantifient les opérations effectuées au cours de la vinification dont l'effet sur la couleur est supposé important.

La sortie est découpée suivant une grille régulière formée de 5 sous-ensembles flous, comme indiqué sur la figure 5.3.

La combinaison de deux facteurs rend la procédure de sélection, décrite au chapitre précédent, inopérante. Ces deux facteurs sont le plus grand nombre de variables, le système en compte treize, conjugué avec la faible cardinalité, trente exemples, de l'ensemble d'apprentissage. Le nombre d'exemples est trop faible par rapport au nombre de combinaisons possibles dans l'espace des entrées, soit $2^{13} = 8192$. Rappelons que l'ensemble des règles possibles, correspondant à toutes les combinaisons des entrées, sont examinées. Les seules qui sont initialisées sont celles qui sont activées par un nombre minimum d'exemples, le paramètre $EffectifMin$, avec un degré de vérité pour chacun de ces exemples supérieur au seuil $MuMin$.

Le nombre de règles activées en fonction du degré de vérité minimum est donné dans le tableau 5.5. $\#R$ désigne le nombre de règles, $\#Ex$ le nombre d'exemples couverts par l'ensemble des règles sur les trente que compte l'ensemble d'apprentissage, PcM est le poids cumulé maximum d'une règle sur la base d'exemples. La valeur maximum du poids cumulé est égale au nombre d'exemples, soit 30 dans notre cas. Si nous considérons qu'une règle, pour être valide, doit être activée par une dizaine d'exemples avec un degré de vérité moyen de 0.5, le poids cumulé moyen attendu d'une règle est de l'ordre de 5. Les valeurs obtenues, inférieures à l'unité pour le poids cumulé maximum, montrent qu'il faut positionner les seuils à des niveaux non significatifs pour assurer une couverture, de fort mauvaise qualité, des deux tiers de l'ensemble d'apprentissage.

MuMin	#R	#Ex	PcM
0.1	20	21	0.63
0.2	10	19	0.63
0.3	6	17	0.53
0.4	2	7	0.53
0.5	0	0	0

TAB. 5.5 – Nombre de règles initialisées en fonction du degré de vérité minimum requis

Le tableau 5.6 synthétise la procédure de sélection. Lors de la première itération, le système est formé de deux sous-ensembles flous par variable d'entrée. Sa performance est mesurée par l'erreur moyenne, ErM. La valeur $\#I$ correspond au nombre d'exemples inactifs, dont le degré de vérité cumulé sur l'ensemble des règles est inférieur au seuil $MuMin$, comme indiqué dans l'équation 4.2. Puis, à chacune des itérations suivantes, un sous-ensemble flou est ajouté sur l'entrée dont le numéro figure sur la ligne $Variable$. Ce numéro est celui de la première colonne du tableau 5.4. Le système est alors caractérisé par les indicateurs de performance et de couverture.

Itération	1	2	3	4	5	6	7	8
Variable		12	1	8	3	3	4	2
ErM	1.05	1.05	1.05	1.31	1.17	1.17	1.18	1.18
#I	25	25	24	25	25	25	25	25
Itération	9	10	11	12	13	14	15	16
Variable	7	5	3	3	3	9	7	9
ErM	1.20	1.22	1.25	1.25	1.21	1.28	1.31	1.36
#I	24	25	25	25	24	26	26	26

TAB. 5.6 – Procédure de sélection lorsque le système initial est formé de deux sous-ensembles flous par entrée

Ce tableau montre que l'indicateur d'erreur n'évolue pas dans le sens souhaité, il augmente

au lieu de baisser, et que le nombre d'exemples inactifs est très grand dès le départ, 25 sur 30. Notre méthode d'initialisation, contrairement à celle qui consiste à générer une règle par exemple, ne garantit pas que les exemples soient des prototypes des règles. Nous sommes donc conduits à modifier notre algorithme.

5.3.2 Modification de la procédure de sélection

Le système initial de la procédure de sélection n'est plus formé de deux sous-ensembles flous par variable, mais d'un seul couvrant l'ensemble du domaine de variation. Ce système ne comporte donc qu'une seule règle dont la conclusion est le sous-ensemble flou de la sortie pour lequel la valeur d'appartenance de la moyenne des intensités colorantes est la plus grande. Comme la distribution est assez symétrique, il s'agit du sous-ensemble flou central, numéro trois. Cette démarche revient à introduire progressivement dans le modèle les variables qui améliorent le plus la performance.

5.3.3 Règles induites dans le repère des trois phases de fermentation

Les variables disponibles sont celles décrites dans le tableau 5.4. Les résultats de la procédure de sélection sont indiqués dans le tableau 5.7.

Itération	1	2	3	4	5	6	7	8	9	10	11	
Variable		4	12	12	8	4	11	10	10	4	4	
ErM	0.28	0.26	0.25	0.25	0.24	0.24	0.22	0.21	0.21	0.20	0.19	
#I	0	0	0	0	0	0	0	0	0	1	1	
Itération	12	13	14	15	16	17	18	19	20	21	22	23
Variable	4	13	10	10	8	3	3	3	3	3	3	4
ErM	0.19	0.18	0.18	0.16	0.16	0.17	0.28	0.46	0.58	0.46	0.15	0.42
#I	1	1	1	1	1	3	6	7	8	6	1	2

TAB. 5.7 – Procédure de sélection dans le repère des trois phases de fermentation

Le critère d'erreur évolue dans le sens attendu, et le nombre d'exemples inactifs reste petit.

Le système qui présente la meilleure performance comprend 32 règles, soit un nombre supérieur à celui du nombre d'exemples. Pour limiter le nombre de règles et pour essayer de conserver un caractère général aux règles induites nous considérons le système correspondant au premier minimum local. Bien que ce ne soit pas visible sur le tableau, du fait de l'impression de l'erreur avec deux chiffres significatif seulement, ce minimum local est atteint lors de l'itération numéro 11. Le système correspondant est formé de 19 règles impliquant 5 variables : *pente*, $Min2\_2$, *Enzymage*, *Remontage*, *Dlestage*.

Remarquons que cette procédure a tendance à privilégier les variables présentes dans le modèle. Lorsque deux variables sont en compétition pour être introduites à une itération donnée, l'itération suivante choisit, parce qu'il améliore mieux la performance, un découpage plus fin de la variable déjà sélectionnée plutôt qu'un découpage élémentaire de la variable rivale à l'étape précédente. Ainsi la variable douze, *Dlestage*, est complètement épuisée dès l'itération numéro quatre car elle ne compte que trois sous-ensembles flous au maximum. De même, la variable *pente*, numéro quatre, compte déjà cinq sous-ensembles flous, sur sept possibles, à l'itération numéro onze. Plus loin dans la procédure, les itérations dix sept à vingt deux ne concernent que la variable trois.

Le tableau 5.8 récapitule les principaux résultats.

	Indicateurs des différents systèmes							
	Nb Sef	MR	#R	ErM	ErMax	#I	#E	Max E
Sélectionné	0 0 0 5 0 0 0 2 0 3 2 3 0	180	19	0.190	-3.3	1		
Simplifié			11	0.244	-3.3	0	2	2

TAB. 5.8 – Résultats obtenus dans le repère des trois phases de fermentation

Performance numérique

L'erreur maximale est importante, -3.3, rappelons que l'intensité colorante varie entre 4.3 et 11.3. Elle ne concerne qu'un seul individu, celle qui lui est immédiatement inférieure vaut -2. L'erreur maximale n'est pas dégradée par la simplification. En revanche, la simplification affecte, comme attendu, la performance globale, mesurée par l'évolution de l'erreur quadratique moyenne, et les performances individuelles, la seconde erreur, après l'erreur maximale, est de $+2.5$.

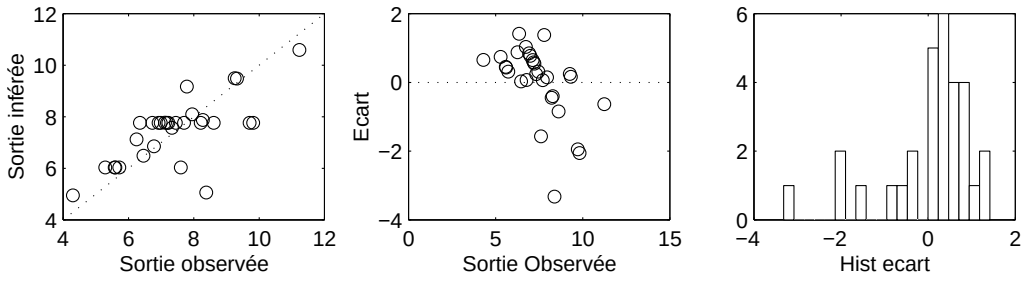


FIG. 5.9 – Représentation des erreurs de prédiction pour le système issu de la sélection

Les graphes de la figure 5.9 montrent qu'il n'y a pas d'interpolation : du fait du caractère discret de certaines variables, les opérations d'enzymage, de remontage, de délestage, il y a peu, voire pas du tout, de recouvrement des parties de l'espace d'entrée couvertes par chacune des règles. C'est la raison pour laquelle nous pouvons observer un ensemble d'individus pour lesquels la sortie inférée est la valeur centrale d'un sous-ensemble flou, voisine de 7.7. Cette absence de recouvrement est, pour certaines variables, imputable à la faible cardinalité de l'ensemble d'apprentissage. D'autres, le remontage ou l'apport de tanins exogènes, sont intrinsèquement binaires.

La simplification se traduit par une diminution importante du nombre de règles et l'obtention d'une seule règle incomplète par rapport au système issu de la sélection. Deux variables, *pente* et $Min2\_2$, sont supprimées de la définition de la prémisse. Dans ce système, seuls cinq exemples activent deux règles dont la sortie est différente : les exemples 7, 9 et 10 activent les règles 1 et 5 ; l'exemple 14 les règles 3 et 4 ; l'exemple 22 les règles 6 et 8.

Les performances du système simplifié sont illustrées graphiquement sur la figure 5.10.

Éléments d'interprétation des règles induites

Les règles sont rapportées dans le tableau 5.9. Pour mieux les analyser nous nous attardons sur les limites des sous-ensembles flous issues du PAH, puis nous associons, à ceux qui sont utilisés dans notre système, un terme linguistique. Enfin, nous présentons la notion de lien entre règles.

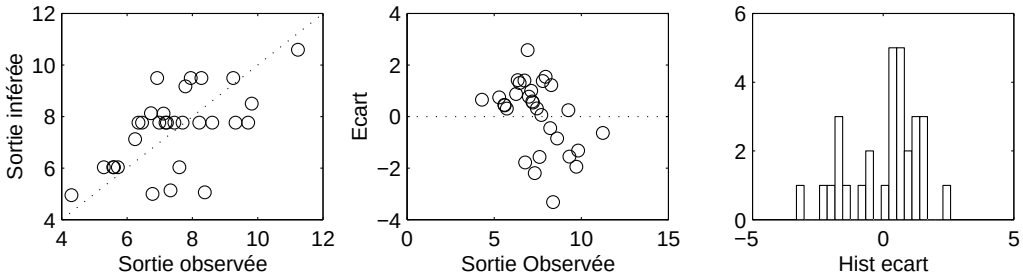


FIG. 5.10 – Représentation des erreurs de prédiction pour le système simplifié

Bornes des sous-ensembles flous : Les limites des sous-ensembles flous pour chacune des variables sélectionnées sont données dans les tableaux 5.10, 5.11, 5.12 et 5.13. La première colonne indique la valeur de l'écart type des densités des sous-ensembles flous dans la partition. Cette valeur est minimum pour les partitions de deux sous-ensembles flous pour les variables *pente* et $Min2\_2$, et pour celles formées de trois sous-ensembles flous pour les variables *enzymage* et *dlestage*. Le découpage retenu par la procédure de sélection, est proche de celui préconisé par l'indice. Il diffère par un découpage plus fin de la variable pente. Ce découpage, qui brouille partiellement la lisibilité des règles, s'explique par l'absence de recouvrement des sous-ensembles flous des autres variables.

Termes linguistiques associés aux sous-ensembles flous : Pour interpréter les règles, il est nécessaire d'étiqueter les sous-ensembles flous de chacune des variables, dont les limites viennent d'être présentées, avec des termes linguistiques qui sont porteurs de sens pour qui connaît le procédé.

La variable pente est découpée en 5 sous-ensembles flous auxquels on pourrait associer des étiquettes de $trs\ faible$ à $trs\ forte$. Comme les valeurs de cette variable sont négatives, l'étiquette $trs\ forte$ est associée au sous-ensemble flou numéro un, elle indique une grande valeur absolue et donc, une durée de fermentation très courte. Le numéro trois correspond à une pente *moyenne*, la pente $trs\ faible$ étant associée au sous-ensemble flou numéro 5, reflète une fermentation lente. $Min2\_2$ est découpée en deux sous-ensembles flous. Le numéro un sera étiqueté comme *basse*, le second comme *leve*. L'*Enzymage*, c'est-à-dire la dose d'enzymes, peut être soit $faible$, sous-ensemble flou numéro un, soit *moyen*, numéro deux, soit enfin *lev*. Pour la variable *Remontage*, la valeur un indique que la cuve est équipée d'un système de remontage, tandis que la valeur deux est associée à un système de pigeage. La variables *delestage* correspond au nombre d'opérations de cette nature effectuées pendant la vinification : il varie de $faible$, sous-ensemble flou un, à *lev*, numéro trois en passant par *moyen*. L'intensité colorante est découpée en 5 sous-ensembles flous auxquels sont associées les étiquettes de $trs\ faible$ à $trs\ leve$, comme illustré sur la figure 5.3.

Le tableau 5.14 fournit des éléments d'interprétation des règles du système simplifié. Chacune des règles est étiquetée par son numéro et, outre la description de la règle, nous indiquons son poids cumulé dans la base d'exemples, Pc, ainsi que la liste des exemples qui l'activent, $NumEx$.

Lien entre règles : Les numéros des exemples qui activent chacune des règles nous permettent d'évaluer le niveau de lien entre les règles. Deux règles seront dites liées si elles sont activées par des ensembles d'exemples voisins.

Cette notion de liens entre règles, introduite dans (Guillaume & Charnomordic, 2001c), est utile pour apprécier la cohérence d'une base de règles. Si deux règles sont fortement liées, et que leurs conclusions sont différentes, alors l'incohérence provient du fait que les domaines d'entrée couverts

Bases de règles obtenues dans le repère des trois phases de fermentation

Règles complètes														Système simplifié												
0,	0,	0,	1,	0,	0,	0,	2,	0,	2,	1,	1,	0,	3	0,	0,	0,	0,	0,	0,	0,	0,	0,	3,	1,	1,	0,	4
0,	0,	0,	1,	0,	0,	0,	2,	0,	3,	1,	1,	0,	4	0,	0,	0,	4,	0,	0,	0,	2,	0,	1,	1,	1,	0,	1
0,	0,	0,	3,	0,	0,	0,	2,	0,	1,	1,	1,	0,	3	0,	0,	0,	2,	0,	0,	0,	2,	0,	1,	1,	1,	0,	3
0,	0,	0,	4,	0,	0,	0,	2,	0,	1,	1,	1,	0,	1	0,	0,	0,	1,	0,	0,	0,	1,	0,	1,	1,	1,	0,	5
0,	0,	0,	2,	0,	0,	0,	2,	0,	2,	1,	1,	0,	3	0,	0,	0,	2,	0,	0,	0,	1,	0,	2,	1,	1,	0,	3
0,	0,	0,	2,	0,	0,	0,	2,	0,	1,	1,	1,	0,	3	0,	0,	0,	5,	0,	0,	0,	2,	0,	1,	2,	1,	0,	2
0,	0,	0,	1,	0,	0,	0,	1,	0,	1,	1,	1,	0,	5	0,	0,	0,	2,	0,	0,	0,	2,	0,	2,	2,	1,	0,	2
0,	0,	0,	2,	0,	0,	0,	1,	0,	2,	1,	1,	0,	3	0,	0,	0,	4,	0,	0,	0,	2,	0,	2,	2,	1,	0,	3
0,	0,	0,	4,	0,	0,	0,	2,	0,	2,	1,	1,	0,	3	0,	0,	0,	4,	0,	0,	0,	2,	0,	3,	2,	3,	0,	3
0,	0,	0,	5,	0,	0,	0,	1,	0,	1,	2,	1,	0,	2	0,	0,	0,	3,	0,	0,	0,	1,	0,	3,	2,	2,	0,	3
0,	0,	0,	5,	0,	0,	0,	2,	0,	1,	2,	1,	0,	2	0,	0,	0,	3,	0,	0,	0,	2,	0,	1,	2,	1,	0,	2
0,	0,	0,	2,	0,	0,	0,	2,	0,	2,	2,	1,	0,	2														
0,	0,	0,	4,	0,	0,	0,	1,	0,	2,	2,	1,	0,	3														
0,	0,	0,	4,	0,	0,	0,	2,	0,	2,	2,	1,	0,	3														
0,	0,	0,	4,	0,	0,	0,	2,	0,	3,	2,	3,	0,	3														
0,	0,	0,	3,	0,	0,	0,	1,	0,	3,	2,	2,	0,	3														
0,	0,	0,	3,	0,	0,	0,	2,	0,	2,	2,	1,	0,	2														
0,	0,	0,	3,	0,	0,	0,	2,	0,	1,	2,	1,	0,	2														
0,	0,	0,	3,	0,	0,	0,	2,	0,	2,	2,	2,	0,	4														

TAB. 5.9 – Règles des systèmes sélectionné et simplifié pour le repère des trois phases de fermentation

σ_d	Centres des SEF				
4.5	-15.8	-4.5			
22.1	-23.8	-13.7	-4.5		
26.2	-23.8	-15.6	-12.2	-4.5	
30.0	-23.8	-15.6	-13.1	-11.1	-4.5

TAB. 5.10 – Centres des SEF et indicateur de qualité pour la variable pente

σ_d	Centres des SEF				
19.7	13.3	24.5			
23.8	13.3	23.6	26.5		
39.7	13.3	22.0	25.1	26.5	
41.7	13.3	20.4	24.1	25.1	26.5

TAB. 5.11 – Centres des SEF et indicateur de qualité pour la variable $Min2\_2$

σ_d	Centres des SEF				
0.64	0	2.1			
0.57	0	1.8	3.2		
1.64	0	1.8	3	3.5	
1.48	0	0.8	2	3	3.5

TAB. 5.12 – Centres des SEF et indicateur de qualité pour la variable enzymage

σ_d	C. des SEF		
2.27	0	2	
1.47	0	1	4

TAB. 5.13 – Centres des SEF et indicateur de qualité pour la variable délestage

par les règles ne sont pas suffisamment spécifiques. Cette situation peut aussi correspondre à la présence d'une exception dans la base d'apprentissage. L'expression mathématique du niveau de lien proposée, pour les règles i et j, est :

$$L_{i,j} = \frac{N_{i,j}}{max(N_i,\ N_j)}$$

N_i représente le cardinal de l'ensemble E_i des exemples qui activent la règle i, $N_{i,j}$ est le cardinal de $E_i \cap E_j$.

Compte tenu de la taille de notre base de données, nous pouvons visuellement vérifier, à l'aide du tableau 5.14, qu'il n'existe aucune paire de règles liées.

Num	*pente*	*Min2_2*	*Enzymage*	*Remontage*	*Dlestage*	*Ic*	Pc	NumEx
1	0	0	3	1	1	4	1.74	2, 4, 8, 9, 11, 18, 30
2	4	2	1	1	1	1	1.32	1, 12, 19, 24
3	2	2	1	1	1	3	2.62	3, 13, 20, 21, 22, 23
4	1	1	1	1	1	5	0.99	17, 22
5	2	1	2	1	1	3	1.51	4, 8, 9, 20
6	5	2	1	2	1	2	1.05	5, 14, 25
7	2	2	2	2	1	2	1.17	16, 26
8	4	2	2	2	1	3	1.43	6, 14, 15, 29
9	4	2	3	2	3	3	0.78	7
10	3	1	3	2	2	3	0.99	10, 28
11	3	2	1	2	1	2	0.77	27

TAB. 5.14 – Éléments d'interprétation des règles simplifiées dans le domaine des trois phases de fermentation

Analyse des règles

Nous n'analyserons pas les règles neuf et onze car elles ne concernent qu'un seul exemple chacune.

Les règles trois et cinq conduisent toutes deux à une intensité colorante moyenne. Elles s'écrivent :
- Règle 3 : Si la *pente* est $forte$ ET si $Min2\_2$ est *leve* ET si l'*Enzymage* est $faible$ ET si la cuve est équipée d'un système de *Remontage* ET si *Delestage* est $faible$ Alors l'*Intensit colorante* est *moyenne*
- Règle 5 : Si la *pente* est $forte$ ET si $Min2\_2$ est $faible$ ET si l'*Enzymage* est *moyen* ET si la cuve est équipée d'un système de *Remontage* ET si *Delestage* est $faible$ Alors l'*Intensit colorante* est *moyenne*

Leur description diffère par deux variables : le minimum de la température dans la deuxième partie de la phase de fermentation et la dose d'enzymes. Quand cette température est plutôt élevée il y aurait besoin de moins d'enzymes. C'est une des règles de base de l'œnologue : les effets de trois facteurs, la durée de fermentation, la température et la dose d'enzyme, se compensent.

Les règles deux et trois diffèrent seulement par le label de la variable pente. Pour la première, il est de quatre, ce qui correspond à une pente $faible$ en valeur absolue et l'intensité colorante associée est $trs\ faible$, sous-ensemble flou un. Pour la seconde, il est de deux et l'intensité colorante correspondante est *moyenne*, sous-ensemble flou trois. Ces règles indiquent l'influence directe de la vitesse de fermentation sur l'intensité colorante. Cela semble être confirmé par la règle quatre : c'est en effet la seule pour laquelle la pente est $trs\ forte$, et cela correspond à la seule conclusion d'une intensité colorante $trs\ leve$.

Toutefois, pour la paire sept et huit, la relation de la pente à la sortie est inversée dans un contexte sensiblement différent : pigeage au lieu de remontage et dose d'enzymes moyenne.

Entre les règles deux et huit, dans un contexte de fermentation lente, l'augmentation de la dose d'enzymes et l'utilisation du pigeage au lieu du remontage permettent de faire évoluer l'intensité colorante d'une valeur $trs\ faible$ à une valeur *moyenne*. En comparant les règles trois et huit, il est possible de conclure que la même combinaison permet de compenser un déficit attendu du fait de la faible vitesse de fermentation. Pour l'œnologue, la différence entre le remontage et le pigeage n'est pas établie. La différence observée provient davantage du fait que les cépages sélectionnés sont, de préférence traités dans les cuves équipées d'un système de pigeage, qui présentent par ailleurs d'autres qualités liées à leur forme.

La seule règle incomplète, la première, indique l'impact de la dose d'enzymes sur l'intensité colorante. C'est encore un résultat attendu, les enzymes sont précisément incorporés pour faciliter l'extraction d'anthocyanes.

5.3.4 Règles induites dans le repère des deux phases d'extraction

Les variables $d1$, $d2$ et $d3$ sont remplacées par les durées aqueuse et alcoolique, daq et dal. Les caractéristiques et les numéros d'ordre des variables dans ce repère sont indiquées dans le tableau 5.15.

Le système dans ce repère est identique à celui trouvé dans le repère des trois phases de fermentation. Les deux évolutions, tableaux 5.7 et 5.16, diffèrent à partir de l'itération numéro dix sept : c'est à ce niveau seulement qu'est introduite la première variable qui n'est pas commune aux deux repères, $d3$.

Nous pouvons retenir que les deux repères temporels sont équivalents pour cette petite base de données. L'œnologue utilise plus spontanément celui des trois phases de fermentation.

Num	Variable	Val Inf	Val Sup	Nb uniques
1	daq	2	8	7
2	dal	2	17	11
3	pente	-23.8	-4.5	29
4	Moy1	13.7	25	26
5	Moy2_1	15.3	31.3	26
6	Max2_1	22.7	33.9	20
7	Min2_2	10.4	27.2	28
8	Moy3	9.9	23.8	27
9	enzymage	0	3.5	5
10	remontage	1	2	2
11	délestage	0	4	3
12	tanins	0	1	2
	Ic	4.3	11.3	30

TAB. 5.15 – Les variables utilisées dans la repère des deux phases d'extraction

Itération	1	2	3	4	5	6	7	8	9
Variable		3	11	11	7	3	10	9	9
ErM	0.28	0.26	0.25	0.25	0.24	0.24	0.22	0.21	0.21
#I	0	0	0	0	0	0	0	0	0
Itération	10	11	12	13	14	15	16	17	18
Variable	3	3	3	12	9	9	7	3	3
ErM	0.20	0.19	0.19	0.18	0.18	0.16	0.16	0.32	0.32
#I	1	1	1	1	1	1	1	1	1

TAB. 5.16 – Procédure de sélection dans le repère des deux phases d'extraction

5.4 Comparaison de méthodes, discussion et perspectives

Rappelons que l'objectif de l'application est la conception d'un simulateur de l'impact des choix de conduite de la vinification sur la couleur du vin rouge. Après une première campagne de mesures trente vinifications ont pu être exploitées. Comparons à présent les règles floues induites par notre méthode d'induction de règles et celles, exprimées en logique booléenne, induites à l'aide d'une méthode de référence, les arbres de décision.

L'arbre de décision n'a retenu aucune variable relative à une action externe. Les variables sélectionnées sont toutes des caractéristiques de l'évolution de la température ou de la densité. Elles représentent aussi une image des choix de conduite. Cette différence dans les variables sélectionnées par les deux méthodes s'explique par les critères de sélection. Le critère qui préside à l'élaboration de l'arbre de décision est la minimisation de la variance interne de l'ensemble des feuilles à une itération donnée. Notre critère principal de sélection est l'amélioration de l'indice de performance numérique. Ces deux critères conduiraient à des résultats équivalents si la séparation suivant les modes, ou les valeurs discrètes pour certaines variables, correspondait à la minimisation de la variance intra-groupe.

La base de règles floues, tout en intégrant une partie des règles de l'arbre de décision, indique l'influence des actions externes. Ces deux bases constituent des sources d'information complémentaires pour l'œnologue. Le caractère interprétable des règles a permis d'une part de retrouver les

grandes lois de la vinification, et d'autre part, d'apporter des éléments de réflexion sur la pratique de la vinification, par exemple sur le lien entre la vitesse de fermentation et la couleur.

Le faible volume de données disponibles après la première campagne de mesures nous interdit de tirer des enseignements valides des règles induites. Elles ne peuvent être que le support d'un dialogue avec l'expert.

La suite du projet, au delà de ce mémoire, devrait bénéficier de données supplémentaires et de méthodes, décrites dans le chapitre consacré à la bibliographie, telles que les arbres de décision flous, présentés dans la section 2.1.4, ou bien la sélection des règles par régression linéaire (OLS), section 2.6.2, qui sont en cours de développement dans notre environnement de travail.

La vinification est un exemple type de la classe d'applications que nous souhaitons traiter. Deux caractéristiques nous paraissent importantes à souligner.

Le nombre de données disponibles pour l'apprentissage est, et sera toujours, limité pour deux raisons principales. D'une part, le nombre d'expériences possibles est fini, il est déterminé, pour la vinification, par la saison des vendanges et par le nombre de cuves traitées. D'autre part, la collecte des données est coûteuse. En effet, seule une partie de celle-ci est automatisée, le relevé des températures. Pour le reste, l'intervention manuelle est indispensable. Comme le procédé est long, et physiquement réparti sur plusieurs sites, vinification plus vieillissement dans notre cas, plusieurs personnes sont impliquées, à des moments différents, dans cette collecte. La possibilité d'incohérences dans cette chaîne doit être gérée de façon spécifique. Des améliorations sont certes possibles, et en cours de développement, mais le caractère incertain et limité des données ne pourra pas être complètement dépassé. Pour prétendre pouvoir travailler dans ces conditions il faut accepter de procéder avec un ensemble d'apprentissage de taille réduite. Nous pouvons estimer que le nombre d'exemples raisonnablement nécessaires pour la validation des règles induites est de l'ordre d'une dizaine par règle.

La seconde caractéristique de cette classe d'applications est le nombre de variables disponibles. Contrairement à des systèmes de type boîte noire, les méthodes qui visent à produire de la connaissance interprétable ne peuvent gérer qu'un nombre limité de variables. La méthodologie proposée ne fait pas exception. Ce fait souligne l'importance du pré-traitement. Réalisé, à l'aide de techniques d'analyse de données multidimensionnelles, en relation étroite avec l'expert du procédé, il nous a permis de limiter le nombre de variables d'entrée à treize.

Le changement d'échelle inhérent à l'application a montré que si toutes les variables disponibles sont introduites dans le modèle, les parties d'espace couvertes par les règles sont si petites que très peu d'exemples activent des règles. Ceci nous a conduit à initialiser le système avec un seul sous-ensemble flou par variable au lieu de deux. Cette modification a été appliquée à la problématique de la qualité globale du riz présentée dans la section 4.4.2. Les résultats sont donnés dans l'annexe F.

Cette méthode n'est pas équivalente à la précédente. Elle inclut de fait une sélection globale des variables. Comme les règles sont définies par un plus petit nombre de variables, elles sont plus difficiles à simplifier. Les règles sont certes incomplètes, car définies par un sous-ensemble des variables seulement, mais elles ne permettent pas d'appréhender la notion de contexte, introduite dans la section 4.2, car elles sont toutes définies avec les mêmes variables. Une comparaison de ces deux démarches reste à établir.

Chapitre 6

Conclusion

Quelqu'un a perdu ses clés la nuit et les cherche près d'un réverbère. Un autre le rejoint et se propose de l'aider : "Vous êtes sûr de les avoir perdues là ?". "Non" répond l'autre. "Alors pourquoi les cherchez vous ici ?". "Parce que c'est le seul endroit où il y a de la lumière."

Bernard Werber

(Werber, 2000)

L'objectif du travail présenté dans ce mémoire est l'induction de règles floues interprétables. Il a été réalisé dans la perspective de la gestion des systèmes complexes. Nous sommes partis du constat que cette gestion nécessitait deux types de connaissance complémentaires, la connaissance experte et celle contenue dans les données. Ces deux contributions sont notamment indispensables pour les applications, telles que la simulation ou l'aide à la décision, dans lesquelles l'expert interagit avec le module informatique. L'autre postulat est que les systèmes d'inférence floue offrent un cadre de coopération entre ces deux types de connaissance sous la réserve de l'interprétabilité, par l'expert, des règles induites.

Nous avons établi trois conditions qui nous semblent nécessaires pour rendre le système de règles interprétable. En premier lieu le découpage des entrées en ensembles flous, le partitionnement, doit être lisible pour les experts. Les ensembles flous doivent correspondre à des termes linguistiques. Ensuite, le nombre de règles doit être minimal. Enfin, pour les systèmes impliquant un grand nombre de variables, les règles générées doivent être incomplètes, c'est-à-dire définies seulement par les variables les plus importantes pour le contexte de chaque règle. Or, historiquement, le but principal de l'induction de règles est l'approximation de relations entre entrées et sortie. L'étude des principales méthodes d'induction disponibles montre que cette interprétabilité n'est pas garantie. Elle ne peut être atteinte que sous contraintes.

L'interprétabilité est au cœur des préoccupations actuelles

Nombre de travaux récents qui visent à améliorer l'interprétabilité des systèmes induits sont parus. La plupart d'entre eux ont été publiés alors que notre propre démarche était bien engagée. Dans (Babuska & Verbruggen, 1995), les auteurs proposent de dériver un modèle linguistique, dont la conclusion des des règles est un ensemble flou, d'un modèle de Takagi-Sugeno, dont la sortie est une combinaison linéaire des entrées. Leur argumentation s'appuie sur les qualités complémentaires de ces deux types de modèles. Les modèles linéaires sont précis tout en utilisant un faible nombre de règles, et il est possible de les initialiser avec un faible nombre d'exemples. A l'inverse, les modèles linguistiques sont interprétables et plus aptes à travailler en généralisation. (Yen et al., 1998; Abonyi & Babuska, 2000) analysent l'impact des stratégies d'apprentissage, globale et locale par l'usage de matrices de pondération, sur la qualité, en terme d'interprétabilité, des modèles résultants. (Fiordaliso, 2001) propose de contraindre la conception des systèmes de Takagi-Sugeno pour en améliorer l'interprétation. Les contraintes portent sur les coefficients de la combinaison linéaire : positivité, somme égale à l'unité et absence d'ordonnée à l'origine, pour maintenir la sortie de chacune des règles dans un espace convexe.

(Cordon & Herrera, 2000), qui ont traité l'application de la qualité du riz que nous avons présentée, proposent plusieurs méthodes pour initialiser la conclusion floue des règles dans le but d'améliorer les performances numériques des modèles linguistiques.

Ayant remarqué que l'une des toutes premières conditions à satisfaire pour que le système soit interprétable est la lisibilité du découpage des entrées en sous-ensembles flous, (Glorennec, 1999; de Oliveira, 1999; Espinosa & Vandewalle, 2000) proposent de la garantir par la contrainte en imposant une partition standardisée, que nous avons appelée partition floue forte.

Les préoccupations, ainsi que l'énoncé des conditions d'interprétabilité, de (Jin, 2000) sont voisines des nôtres. Il nous semble toutefois qu'il ne conduit pas sa démarche à son terme, et obtient, par conséquent, des résultats décevants en terme d'interprétabilité.

Malheureusement le terme interprétabilité n'est pas défini de manière rigoureuse, ce qui conduit à des interprétations très diverses. Pour (Bikdash, 1999) l'interprétabilité visée est celle des coefficients

du modèle de Takagi-Sugeno à la manière des coefficients d'une série de Taylor. Ce type de résultats reste très éloigné de nos préoccupations d'interprétabilité des règles par l'expert.

Enfin, (Krone et al., 2000) proposent deux stratégies pour générer des règles incomplètes, définies par les seules variables utiles. La première consiste à ne retenir que les règles qui assurent la meilleure couverture de l'espace d'entrée tandis que la seconde vise à prévenir d'éventuels conflits. Les règles activées principalement par un exemple qui active encore plus une autre règle sont supprimées.

Ce foisonnement dans la littérature récente, auquel participe le présent travail, témoigne de la volonté d'une partie importante de la communauté floue de ramener les systèmes d'inférence floue vers les rives de la logique floue en privilégiant ce qui fait son originalité : l'interprétabilité des règles.

Notre contribution

Nous avons développé une méthode originale de conception d'un système d'inférence floue à partir d'une base de données. Elle est décomposée en trois étapes :

1. générer un partitionnement interprétable pour chacune des entrées ;
2. construire le système multidimensionnel le plus performant parmi les plus compacts en ne gardant que les règles ayant une influence significative dans l'ensemble d'apprentissage ;
3. simplifier le système en éliminant les variables inutiles.

Nous avons introduit les notions d'exemple inactif, de niveau de couverture et d'hétérogénéité de la sortie d'une règle pour caractériser, et maintenir au cours du processus, certaines propriétés de la base de règles, notamment la complétude et la cohérence. La complétude est caractérisée par un indice de couverture, égal au nombre d'exemples inactifs. Un exemple est déclaré inactif si son degré de vérité cumulé sur l'ensemble des règles est inférieur à un seuil donné. L'élimination des variables se traduit par un élargissement de l'espace d'entrée couvert par une règle. La détection d'incohérences se vérifie au moyen de la mesure de l'homogénéité de la sortie des exemples attirés par la règle.

La première étape conduit à un partitionnement interprétable. Elle est monodimensionnelle, appliquée à chacune des variables d'entrée du système. L'originalité de la méthode que nous proposons consiste à générer, non pas une partition, mais une famille de partitions floues de manière analogue à une classification ascendante hiérarchique. Nous l'avons appelée partitionnement ascendant hiérarchique. A chacune de ces partitions sera associé un indice de qualité. Cet indice, vise à caractériser la partition pour elle-même, indépendamment d'un éventuel usage ultérieur. L'utilisateur pourra alors, guidé par cet indice, choisir le nombre de sous-ensembles flous le plus adapté.

La partition initiale est formée d'autant de sous-ensembles flous qu'il y a de valeurs considérées comme distinctes dans la base d'apprentissage pour l'entrée considérée. Puis, à chaque étape, deux sous-ensembles flous adjacents sont fusionnés pour en former un seul, couvrant un domaine plus large, jusqu'à l'étape finale qui conduit à une partition de l'espace en un sous-ensemble flou unique couvrant l'ensemble du domaine de variation, encore appelé ensemble universel. La paire fusionnée est celle qui conserve au mieux la structure élaborée à l'étape précédente. Le critère de fusion s'appuie sur une fonction de distance entre observations qui prend en compte le partitionnement en cours de construction. La technique de fusion préserve l'intégrité sémantique de la partition à chaque étape du processus.

La deuxième étape a pour but la sélection du système d'inférence floue dit le plus performant parmi les plus compacts. La construction d'un système se réalise en deux temps : il s'agit d'abord de générer les prémisses de l'ensemble des règles possibles pour un partitionnement donné, puis de sélectionner les règles les plus influentes tout en initialisant leur conséquence.

L'étape de sélection est initialisée par la construction d'un système simple. Elle s'appuie sur les familles de partitions de chacune des entrées trouvées à l'étape précédente. Si le nombre de variables n'est pas trop grand, il est possible de commencer avec deux sous-ensembles flous par entrée. Dans le cas contraire, il faut se limiter à un seul sous-ensemble flou, couvrant l'ensemble du domaine de variation. Le système, formé de toutes les règles possibles, une seule dans le cas d'un seul sous-ensemble flou par variable, est caractérisé par son indice de performance. Puis à chaque pas, un sous-ensemble flou est ajouté sur l'une des entrées. Lorsqu'il n'y a qu'un seul sous-ensemble flou, le fait d'en ajouter un deuxième signifie simplement que la variable correspondante est introduite dans le modèle. L'entrée choisie est celle qui minimise l'erreur. Les coordonnées des centres des sous-ensembles flous sont celles résultant du partitionnement ascendant hiérarchique. L'algorithme s'arrête lorsque l'indicateur d'erreur remonte ou après un certain nombre d'itérations. Le résultat est donc le système le plus performant parmi les plus compacts.

Durant la procédure, des paramètres permettent de ne conserver que les règles, dont l'influence est significative.

La troisième étape vise à simplifier la base de règles. Elle est indispensable lorsque, pour former le système initial, les entrées sont découpées en deux sous-ensembles flous. En effet, les règles obtenues à l'étape précédente sont définies par le même nombre de variables.

Nous avons introduit la notion de contexte pour effectuer la sélection des variables. Un contexte est défini par l'ensemble des règles, que nous appelons groupe, dont les prémisses sont identiques à l'exception d'un sous-ensemble flou d'une même variable et d'un seul. Un groupe permet d'évaluer l'influence d'une variable particulière dans le contexte défini par les autres variables. Ce niveau d'intervention nous paraît pertinent. Il constitue un intermédiaire entre une sélection globale et une sélection règle par règle. Dans la littérature, la sélection de variables s'effectue classiquement de manière globale, plus rarement au niveau d'une règle. L'élimination globale, qui suppose qu'une variable donnée est d'égale importance quel que soit le contexte, n'est pas toujours adaptée aux systèmes de grande dimension. Souvent, les variables influentes dépendent de l'état du système. La sélection règle par règle, à laquelle il est plus difficile d'attacher une sémantique, ne permet pas d'appréhender l'ensemble d'un contexte.

Nous avons considéré différentes phases de simplification, correspondant à des niveaux d'intervention de plus en plus précis. Ces niveaux sont traités en séquence : ce n'est que lorsqu'il n'y a plus aucune simplification possible à un niveau que le niveau suivant est examiné.

a) Le premier niveau est celui du groupe de règles, si la variable correspondante n'a qu'une influence limitée dans le contexte du groupe, elle est éliminée dans chacune des règles du groupe. Les règles du groupe deviennent alors identiques et peuvent être fusionnées en une seule. L'identification des groupes est basée sur une notion de distance entre règles. Le critère de fusion des règles d'un groupe vérifie l'homogénéité de la sortie des exemples attirés par le groupe.
b) Le deuxième niveau vise à éliminer la redondance. Pour ce faire la présence de chacune des règles dans la base doit être justifiée. Une règle sera éliminée, si malgré son absence, la base de règles reste performante, et si le nombre d'exemples qui n'activent que faiblement les règles reste en deçà d'un seuil fixé. Les critères de performance et de couverture permettent de valider cette élimination.

c) Enfin, le troisième niveau s'intéresse à chacune des variables des règles restantes. Chacune des variables est examinée indépendamment. Une variable est éliminée de la prémisse si la sortie des exemples attirés par la règle demeure homogène malgré l'élargissement de l'espace d'entrée couvert. Le critère d'élimination d'une variable est celui utilisé pour la fusion des règles d'un groupe.

Ces simplifications, qui améliorent grandement la lisibilité du système, se paient parfois au prix d'une détérioration, dans des limites fixées, de la performance numérique du système.

Discussion et application à un procédé de vinification

La méthode proposée favorise la coopération avec la connaissance experte à tous les niveaux de la conception. Les familles de partitions monodimensionnelles sont lisibles : l'utilisateur peut valider les limites des sous-ensembles flous, ou même indiquer le nombre de sous-ensembles flous qui lui semble le plus adapté pour chacune des variables. Lors de l'étape de sélection, l'expert peut facilement désactiver des entrées. Toutes les étapes de la sélection et de la simplification sont enregistrées. L'analyse de l'historique et des règles des différents systèmes, la représentation graphique de la distribution des écarts sont des éléments indispensables pour choisir la configuration la plus adaptée. En effet, la performance est une notion multidimensionnelle, pour laquelle nous ne disposons pas de métrique, qui ne saurait être réduite à un indice tel que l'erreur quadratique moyenne. Cette relation entre la méthode et l'expert n'est pas courante. La plupart des auteurs voient l'intervention humaine dans les méthodes de conception automatique comme une contrainte.

La volonté de contribuer à la résolution de problèmes nous a amené à veiller au caractère utilisable des méthodes proposées, et nous a invité à développer des interfaces utilisateurs conviviales. Nous essayons de nous situer au cœur d'une dialectique entre l'application et la recherche. Les pistes de recherche abordées dans ce mémoire sont, au moins partiellement, formulées à partir de préoccupations appliquées. Nous voulons qu'elles produisent des outils à la disposition des professionnels et des laboratoires de recherche, et nous souhaitons que l'utilisation de ces outils nous renvoie des questions qui pourront être, à leur tour, la base de nouvelles problématiques de recherche, notamment autour de la fusion de bases de règles.

En appliquant la méthode développée à un procédé de vinification, avec pour objectif de mieux comprendre l'impact des actions de conduite sur la couleur au cours de la vinification du vin rouge, nous avons mis en évidence le potentiel de cette approche pour cette classe de problèmes. La production de vins plus tanniques et plus colorés vise à satisfaire une demande pressante des consommateurs et représente un important enjeu économique pour la filière viticole. Le système est principalement destiné à l'œnologue, dans un but d'aide à la conduite du procédé de vinification.

Une première sélection des variables pertinentes pour l'induction de règles a été réalisée, à partir d'une analyse multidimensionnelle des données, en relation étroite avec l'œnologue. Des règles ont été induites par un arbre de décision d'une part, et par la méthode que nous avons développée, d'autre part. Il est difficile, compte tenu du faible volume de données disponibles après la première campagne de mesures, d'accorder une validité, basée sur les seuls résultats de l'induction, à ces règles. Elles ont été principalement utilisées comme un support de dialogue avec l'œnologue, lequel y a retrouvé les grandes lois empiriques de la vinification, ainsi que de éléments de réflexion sur sa pratique. L'application se poursuit, avec notamment une nouvelle campagne d'acquisition de données lors des vendanges 2001.

Perspectives

Les chemins ouverts par ce travail n'ont pas tous été explorés. Pour le partitionnement ascendant hiérarchique, plusieurs types de distance ont été introduits. La distance symbolique, supposée mieux

adaptée à des données de nature symbolique, devra être testée sur ce type de données. De même les réductions contenues dans la distance dite simplifiée restent à valider.

Pour construire un système performant et compact, la gestion de l'application nous a amené à modifier notre stratégie. Nous avons tout d'abord envisagé d'introduire l'ensemble des variables dans le système. Malgré l'utilisation d'un partitionnement grossier, le découpage de l'espace correspondant est très fin lorsque le nombre de variables est important. Dans ce cas, les règles ne peuvent pas être initialisées. Le changement de stratégie a consisté à démarrer avec un système formé d'une seule règle : aucune variable n'est présente dans le modèle, seules celles qui améliorent la performance sont sélectionnées au cours de la procédure. Ces deux stratégies ne sont pas équivalentes. La seconde inclut une sélection globale des variables. Dans ce cas, le système de règles résultant est plus difficile à simplifier. Les règles sont certes incomplètes, leur prémisse ne contient qu'une partie des variables, mais elles sont toutes définies par les mêmes variables. L'utilisation de cette stratégie nous fait perdre le bénéfice de la notion de contexte. D'autres travaux sont nécessaires pour comparer ces deux approches.

La notion de performance est centrale dans les procédures d'apprentissage. Nous l'avons utilisée non seulement pour caractériser le système, mais également dans l'ensemble des étapes de sa conception. Nous avons montré que la performance est multidimensionnelle, et ne peut être réduite à un indice tel que l'erreur quadratique moyenne. La définition d'une métrique serait utile pour les algorithmes d'apprentissage.

Pour les besoins du développement, la méthode est complètement paramétrée. Un travail important est nécessaire pour figer un certain nombre de paramètres. Deux approches complémentaires sont possibles. La première consiste à inclure des procédures d'apprentissage pour certains paramètres, la seconde à proposer des valeurs par défaut pour quelques classes d'application. Ces classes peuvent être définies à partir du type de sortie, classification ou régression, mais aussi de l'allure de la distribution des valeurs de sortie, ou encore d'éventuelles contraintes, telle que la couverture du domaine de sortie. Comme les valeurs de sortie extrêmes sont en général rares, le processus d'induction, qui conduit à ne sélectionner que les règles les plus représentatives, élimine assez vite les règles capables d'inférer de telles valeurs. Il serait intéressant d'appliquer une stratégie spécifique pour ces queues de distribution, soit en effectuant une collecte de données biaisée, dans laquelle ces valeurs sont sur-représentées, soit en imposant par la contrainte la présence de règles capables d'assurer une couverture complète du domaine de la sortie.

Le processus d'induction, s'il représente une voie privilégiée pour tirer parti des données expérimentales, est intrinsèquement limité. Par essence, il ne peut pas traiter les exceptions. Il est aussi limité par la taille de la base de l'apprentissage qui ne peut prétendre couvrir l'ensemble des situations possibles. Il est également limité par le type de règles induites. Les règles induites sont des règles dites conjonctives à possibilité. La sémantique qui leur est attachée signifie *si l'entrée correspond à la prémisse de la règle alors il est possible que la sortie soit du type de la conclusion de la règle*. Activer une règle, c'est pour un exemple, ouvrir un champ de possibilité dans l'espace de la sortie. Leur agrégation est disjonctive : plus un exemple active de règles, plus sa sortie est incertaine, dans le sens où les valeurs possibles sont importantes. Les experts manipulent d'autres types de règles, notamment des règles implicatives, qui imposent une contrainte sur la sortie : *"Si je rencontre telle situation, alors je ne dois surtout pas agir de telle façon"*. Ces règles, dont la sémantique est *si la prémisse est vraie alors la conclusion est imposée*, sont plus lourdes à manipuler. Il est préférable de procéder à leur agrégation, qui est conjonctive, ce qui correspond bien à une restriction des valeurs possibles, avant l'inférence. Une typologie des règles floues, et de leur sémantique, a été établie par Dubois et Prade, (Dubois & Prade, 1996), et récemment prolongée par (Ughetto, 1998).

Ces raisons font que les règles induites ne peuvent pas être considérées comme suffisantes pour la gestion d'applications complexes. Leur coopération avec l'expertise est indispensable.

Les règles expertes sont sous-tendues par une longue expérience, elles sont relatives aux grandes tendances du procédé et traduisent l'influence des principales variables. Elles présentent un caractère universel. La problématique de la fusion de plusieurs bases de règles, notamment celle des règles expertes et des règles induites, n'est pas traitée dans ce mémoire. Quelques pistes de réflexion peuvent, dès à présent, être esquissées. La définition du partitionnement constitue un point d'attention particulier. Il nous semble qu'il ne faut pas demander aux experts de le définir complètement. Leur contribution peut s'arrêter à la définition des termes linguistiques dont ils ont besoin pour formuler leurs règles. En effet, leur demander d'établir les limites des ensembles flous correspondants, revient à les obliger à s'exprimer dans un langage qui n'est pas le leur. Il est plus efficace de leur demander de valider des limites induites à partir de données. L'exemple du partitionnement illustre assez bien le type de coopération possible entre l'expertise et les données. La fusion de la base des règles expertes et des règles induites sera facilitée si le partitionnement est commun aux deux bases. Le processus de fusion devra garantir, pour la nouvelle base, les propriétés fondamentales d'une base de règles : la cohérence, la complétude, la performance et l'interprétabilité.

Bibliographie

Abonyi, J. & Babuska, R. (2000) – Local and global identification and interpretation of parameters in takagi-sugeno fuzzy models. *Proc. of the Nineth IEEE International Conference on Fuzzy Systems*, San Antonio, Texas, USA.

Altug, S., Chow, M.-Y., & Trussel, H. J. (1999) – Heuristic constraints enforcement for training of and knowledge extraction from a fuzzy/neural architecture- part ii : Implementation and application – *IEEE Transactions on Fuzzy Systems, 7 (2)*, 151–159.

Aragon, L. (1973) – *Les communistes* (Oeuvres romanesques croisées d'Elsa Triolet et Aragon, Tome 26, Vol 4^{e} éd.) – Robert Laffont.

Babuska, R., Roubos, J. A., & Verbruggen, H. B. (1998) – Identification of mimo systems by input-output ts fuzzy models. *Fuzz-IEEE 98*, (pp. 657–662)., Anchorage, Alaska.

Babuska, R. & Verbruggen, H. B. (1995) – A new identification method for linguistic fuzzy models. *Proc. of the Fourth IEEE International Conference on Fuzzy Systems*, (pp. 905–912)., Yokohama, Japan.

Babuska, R. & Verbruggen, H. B. (1996) – An overview of fuzzy modeling for control – *Control Engineering Practice, 4 (11)*, 1593–1606.

Benoit, E. & Foulloy, L. (1993) – Exemple de capteur symbolique flou en reconnaissance des couleurs. *RGE*, tome 3, (pp. 22–27).

Bersini, H., Duchateau, A., & Bradshaw, N. (1997) – Using incremental learning algorithms in the search for minimal and effective fuzzy models. *Sixth international conference on fuzzy systems, Vol 3*, (pp. 1417–1422)., Barcelona, Spain.

Bezdek, J. C. (1981) – *Pattern Recognition with Fuzzy Objective Functions Algorithms* – Plenum Press, New York.

Bikdash, M. (1999) – A highly interpretable form of sugeno inference systems – *IEEE Transactions on Fuzzy Systems, 7 (6)*, 686–696.

Bonnet, J. (2000) – *La longue route de Jacob Ben Ezra* – Les Presses du Languedoc.

Bortolet, P. (1998) – *Modelisation et commande multivariable floues : application a la commande d'un moteur thermique* – Thèse de Doctorat, LAAS-CNRS, Institut National des Sciences Appliquées, Toulouse, France.

Bouchon-Meunier, B. (1993) – *La logique floue* – Presses Universitaires de France.

Bouchon-Meunier, B. (1995) – *La logique floue et ses applications* – Addison-Wesley france.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984) – *Classification and Regression Trees* – Wadsworth International Group, Belmont CA.

Burrough, P., van Gaans, P., & MacMillan, R. (2000) – High-resolution landform classification using fuzzy k-means – *Fuzzy Sets and Systems, 113, Issue 1*, 37–52.

Chaudhuri, B. B. & Rosenfeld, A. (1996) – On a metric distance between fuzzy sets – *Pattern Recognition Letters, 17*, 1157–1160.

Chaudhuri, B. B. & Rosenfeld, A. (1999) – A modified hausdorff distance between fuzzy sets – *Information Sciences*, *118*, 159–171.

Chen, M.-S. & Wang, S.-W. (1999) – Fuzzy clustering analysis for optimizing fuzzy membership functions – *Fuzzy Sets and Systems*, *103*, 239–254.

Chen, S., Billings, S. A., & Luo, W. (1989) – Orthogonal least squares methods and their application to non-linear system identification – *Int. J. Control*, *50*, 1873–1896.

Chen, S., Cowan, C. F. N., & Grant, P. M. (1991) – Orthogonal least squares learning algorithm for radial basis function networks – *IEEE Transactions on Neural Networks*, *2 (No 2)*, 302–309.

Chiang, C.-K., Chung, H.-Y., & Lin, J.-J. (1997) – A self-learning fuzzy logic controller using genetic algorithms with reinforcements – *IEEE Transactions on Fuzzy Systems*, *5 (3)*, 460–467.

Chiu, S. L. (1994) – Fuzzy model identification based on cluster estimation – *Journal of Intelligent and Fuzzy Systems*, *2*, 267–278.

Cho, K. B. & Wang, B. H. (1996) – Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction – *Fuzzy Sets and Systems*, *83*, 325–339.

Chow, M.-Y., Altug, S., & Trussel, H. J. (1999) – Heuristic constraints enforcement for training of and knowledge extraction from a fuzzy/neural architecture- part i : Foundation – *IEEE Transactions on Fuzzy Systems*, *7 (2)*, 143–150.

Clergue, G. (1997) – *L'apprentissage de la complexité* – Hermès, Paris.

Cordon, O. & Herrera, F. (2000) – A proposal for improving the accuracy of linguistic modeling – *IEEE Transactions on Fuzzy Systems*, *8 (3)*, 335–344.

Cordon, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001) – *GENETIC FUZZY SYSTEMS Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* – Advances in Fuzzy Systems - Applications and Theory Vol. 19, World Scientific.

de Oliveira, J. V. (1999) – Semantic constraints for membership functions optimization – *IEEE Transactions on Systems, Man and Cybernetics*, *29*, 128–138.

Dubois, D. & Prade, H. (1980) – *Fuzzy Sets and Systems : Theory and Applications* – Academic Press, New York.

Dubois, D. & Prade, H. (1996) – What are fuzzy rules and how to use them – *Fuzzy Sets and Systems*, *84(2)*, 169–186.

Dubois, D. & Prade, H. (1997) – The three semantics of fuzzy sets – *Fuzzy Sets and Systems*, *90*, 141–150.

Dunn, J. C. (1973) – A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters – *J. Cybernetics*, *3 (3)*, 32–57.

Emami, M. R., Türksen, I. B., & Goldenberg, A. A. (1998) – Development of a systematic methodology of fuzzy logic modeling – *IEEE Transactions on Fuzzy Systems*, *6 (3)*, 346–361.

Espinosa, J. & Vandewalle, J. (2000) – Constructing fuzzy models with linguistic integrity from numerical data-afreli algorithm – *IEEE Transactions on Fuzzy Systems*, *8 (5)*, 591–600.

Fan, J. & Xie, W. (1999) – Distance measure and induced fuzzy entropy – *Fuzzy Sets and Systems*, *104*, 305–314.

Fiordaliso, A. (2001) – A constrained takagi-sugeno fuzzy system that allows for better interpretation and analysis – *Fuzzy Sets and Systems*, *118*, 307–318.

Foulloy, L., Galichet, S., & Benoit, E. (1994) – Fuzzy control with fuzzy state sensors. *EUFIT'94*, (pp. 1156–1160)., Aachen, Germany.

Gacogne, L. (1997) – *Elements de logique floue* – Editions Hermes, Paris.

Glorennec, P.-Y. (1991) – Un reseau "neuro-flou" evolutif. *Neuro-Nimes, Fourth International Conference, Neural Networks and their Applications*, Nanterre, France. EC2.

Glorennec, P.-Y. (1999) – *Algorithmes d'apprentissage pour systèmes d'inférence floue* – Editions Hermès, Paris.

Goldberg, D. E. (1989) – *Genetic Algorithm in Search, Optimization and Machine Learning* – Addison Wesley.

Gomez-Skarmeta, A., Delgado, M., & Vila, M. (1999) – About the use of fuzzy clustering techniques for fuzzy model identification – *Fuzzy Sets and Systems*, *106*, 179–188.

Gonzalez, A. & Perez, R. (1999) – Slave : A genetic learning system based on an iterative approach – *IEEE Transactions on Fuzzy Systems*, *7 (2)*, 176–191.

Guedj, D. (1998) – *Le théorème du perroquet* – Editions du Seuil, France.

Guillaume, S. (2001) – Designing fuzzy inference systems from data : an interpretability-oriented review – *IEEE Transactions on Fuzzy Systems*, *9 (3)*, 426–443.

Guillaume, S. & Charnomordic, B. (2001) – Knowledge discovery for control purposes in food industry databases – *Fuzzy sets and Systems*, *122 (3)*, 487–497.

Guély, F., La, R., & Siarry, P. (1999) – Fuzzy rule base learning through simulated annealing – *Fuzzy Sets and Systems*, *105*, 353–363.

Gustafson, D. E. & Kessel, W. C. (1979) – Fuzzy clustering with a fuzzy covariance matrix. *Proc. IEEE CDC*, (pp. 761–766)., San Diego, CA, USA.

Hammah, R. E. & Curran, J. H. (1999) – On distance measures for the fuzzy k-means algorithm for joint data – *Rock Mechanics and Rock Engineering*, *32 (1)*, 1–27.

Hathaway, R. & Bezdek, J. (1993) – Switching regression model and fuzzy clustering – *IEEE Transactions on Fuzzy Systems*, *1*, 195–204.

Hoffmann, F. & Pfister, G. (1996) – *Learning a fuzzy control rule base using messy genetic algorithms in Genetic algorithms and soft computing, Number 8 in Studies in Fuzziness and Soft Computing* – (pp. 279–305)., Physica-Verlag.

Hohensohn, J. & Mendel, J. M. (1994) – Two pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems. *Proc. IEE Conf. Fuzzy Syst.*, (pp. 696–700)., Orlando, Florida.

Holland, J. H. (1975) – *Adaptation in natural and artificial systems* – The University of Michigan Press, Ann Arbor.

Hong, T.-P. & Chen, J.-B. (1999) – Finding relevant attributes and membership functions – *Fuzzy Sets and Systems*, *103*, 389–404.

Huang, Y.-P. & Chu, H.-C. (1999) – Simplifying fuzzy modeling by both gray relational analysis and data transformation methods – *Fuzzy Sets and Systems*, *104*, 183–197.

Ichihashi, H., Shirai, T., Nagasaka, K., & Miyoshi, T. (1996) – 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*, 157–167.

Ishibuchi, H., Nozaki, K., Tanaka, H., Hosaka, Y., & Matsuda, M. (1994) – Empirical study on learning in fuzzy systems by rice test analysis – *Fuzzy Sets and Systems*, *64*, 129–144.

Ishibuchi, H., Nozaki, K., Yamamoto, N., & Tanaka, H. (1995) – Selecting fuzzy if-then rules for classification problems using genetic algorithms – *IEEE Transactions on Fuzzy Systems*, *3 (3)*, 260–270.

Jang, J.-S. R. (1993) – Anfis :adaptive-network-based fuzzy inference systems – *IEEE Transactions on Systems, Man ad Cybernetics*, *23 (03)*, 665–685.

Jang, J.-S. R. & Sun, C.-T. (1993) – Functional equivalence between radial basis function network and fuzzy inference systems – *IEEE Transactions on Neural Networks*, *4*, 156–159.

Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997) – *Neuro-Fuzzy and Soft Computing* – Prentice Hall.

Jin, Y. (2000) – Fuzzy modeling of high-dimensional systems : complexity reduction and interpretability improvement – *IEEE Transactions on Fuzzy Systems*, *8 (2)*, 212–221.

Jouffe, L. (1997) – *Apprentissage de systèmes d'inférence floue par des méthodes de renforcement : application à la régulation d'ambiance dans un bâtiment d'élevage porcin* – Thèse de Doctorat, Université de Rennes I.

Karr, C. L. (1991) – Design of a cart-pole balancing fuzzy logic controller using a genetic algorithm. SPIE (édité par), *Conf. on applications of Artificial Intelligence*. Bellingham, WA.

Kaymak, U. & Babuska, R. (1995) – Compatible cluster merging for fuzzy modeling. *Proc. of the Fourth IEEE International Conference on Fuzzy Systems*, (pp. 897–904)., Yokohama, Japan.

Kim, E., Park, M., Ji, S., & Park, M. (1997) – A new approach to fuzzy modeling – *IEEE Transactions on Fuzzy Systems*, *5*, 328–337.

Kim, E., Park, M., Kim, S., & Park, M. (1998) – A transformed input-domain approach to fuzzy modeling – *IEEE Transactions on Fuzzy Systems*, *6 (4)*, 596–604.

Klawonn, F. & Keller, A. (1997) – Fuzzy clustering and fuzzy rules. *Seventh IFSA World Congress*, Prague.

Koczy, L. & Hirota, K. (1993) – Ordering, distance and closeness of fuzzy sets – *Fuzzy Sets and Systems*, *59*, 281–293.

Krishnapuram, R. & Freg, C.-P. (1992) – Fitting an unknown number of lines and planes to image data through compatible cluster merging – *Pattern Recognition*, *25 (4)*, 385–400.

Krishnapuram, R. & Keller, J. M. (1993) – A possibilistic approach to clustering – *IEEE Transactions on Fuzzy Systems*, *(1)*, 98–110.

Krishnapuram, R. & Kim, J. (1999) – A note on the gustafson-kessel and adaptative fuzzy clustering algorithms – *IEEE Transactions on Fuzzy Systems*, *7 (4)*, 453–461.

Krone, A., Krause, P., & Slawinski, T. (2000) – A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. *Proc. of the Nineth IEEE International Conference on Fuzzy Systems*, San Antonio, Texas, USA.

Lacrose, V. (1997) – *Réduction de la complexité des contrôleurs flous : application a la commande multivariable* – Thèse de Doctorat, LAAS-CNRS, Institut National des Sciences Appliquées, Toulouse, France.

Leitch, D. & Probert, P. (1998) – New techniques for genetic development of fuzzy controllers – *IEEE Transactions on Systems, Man and Cybernetics : Part C, Applications and Reviews*, *28 1*, 112–123.

Li, R.-P. & Mukaidono, M. (1999) – Gaussian clustering method based on maximum-fuzzy-entropy interpretation – *Fuzzy Sets and Systems*, *102*, 253–258.

Lin, Y. & Cunningham, G. A. (1994) – A fuzzy approach to input variable identification. *Proc. IEEE Conf. Fuzzy Syst.*, (pp. 2031–2036)., Orlando, Florida.

Lin, Y., Cunningham, G. A., & Coggeshall, S. V. (1997) – Using fuzzy partitions to create fuzzy systems from input-output data and set the initial weights in a fuzzy neural network – *IEEE Transactions on Fuzzy Systems*, *5 (4)*, 614–621.

Lowen, R. & Peeters, W. (1998) – Distance between fuzzy sets representing grey level images – *Fuzzy Sets and Systems*, *99*, 135–149.

Magdalena, L. (1997) – Adapting the gain of an flc with genetic algorithms – *International Journal of Approximate Reasoning, 17 (4)*, 327–349.

Mamdani, E. H. & Assilian, S. (1975) – An experiment in linguistic synthesis with a fuzzy logic controller – *International journal of Man-Machine Studies, 7*, 1–13.

Marsala, C. (1998) – Construction d'arbres de decision flous : le systeme salammbo. *Rencontres francophones sur la logique floue et ses applications*, Toulouse, France. Cepadues Editions.

Martens, H. & Naes, T. (1989) – *Multivariate calibration* – John Wiley and sons.

Mitra, S., De, R. K., & Pal, S. K. (1997) – Knowledge-based fuzzy mlp for classification and rule generation – *IEEE Transactions on Neural Networks, 8 (6)*, 1338–1350.

Mitra, S. & Hayashi, Y. (2000) – Neuro-fuzzy rule generation : Survey in soft computing framework – *IEEE Transactions on Neural Networks, 11 (3)*, 748–768.

Moody, J. & Darken, C. (1989) – Fast learning in networks of locally-tuned process units – *Neural Computation, 1*, 281–294.

Morin, E. (1990) – *Introduction à la pensée complexe* – ESF éditeur, Paris.

Nozaki, K., Ishibuchi, H., & Tanaka, H. (1997) – A simple but powerful heuristic method for generating fuzzy rules from numerical data – *Fuzzy Sets and Systems, 86*, 251–270.

OFTA (1994) – *Logique floue* (André Titli et Laurent Foulloy^{e} éd.) – Observatoire francais des techniques avancées, n°14, Editions Masson.

Pal, N. R. (1999) – Soft computing for feature analysis – *Fuzzy Sets and Systems, 103*, 201–221.

Pedrycz, W. (1994) – Why triangular membership functions ? – *Fuzzy sets and Systems, 64 (1)*, 21–30.

Perrier, X. (1998) – *Analyse de la diversité génétique : mesures de dissimilarités et représentations arborées* – Thèse de Doctorat, Université Montpellier II.

Quinlan, J. R. (1986) – Induction of decision trees – *Machine Learning, 1*, 81–106.

Rojas, I., Pomares, H., Ortega, J., & Prieto, A. (2000) – Self-organized fuzzy system generation from training examples – *IEEE Transactions on Fuzzy Systems, 8 (1)*, 23–36.

Roy, A. (2000) – On connectionism, rule extraction, and brain-like learning – *IEEE Transactions on Fuzzy Systems, 8 (2)*, 222–227.

Ruck, D. W., Rogers, S. K., & Kabrisky, M. (1990) – Feature selection using a multilayer perceptron – *J. Neural Network Comput., (1)*, 40–48.

Runkler, T. A. & Bezdek, J. C. (1999) – Alternating cluster estimation : A new tool for clustering and function approximation – *IEEE Transactions on Fuzzy Systems, 7 (4)*, 377–393.

Russo, M. (1998) – Fugenesys- a fuzzy genetic neural system for fuzzy modeling – *IEEE Transactions on Fuzzy Systems, 6 (3)*, 373–388.

Saporta, G. (1990) – *Probabilités, analyse des données et statistiques* – Editions Technip, France.

Setnes, M., Babuska, R., Kaymak, U., & van Nauta Lemke, H. R. (1998) – Similarity measures in fuzzy rule base simplification – *IEEE Transactions on Systems, Man and Cybernetics, 28(3)*, 376–386.

Shafer, G. (1976) – *A Mathematical Theory of Evidence* – Princeton University Press, Princeton, NJ.

Spire, A. (1999) – *La pensée-Prigogine* – Desclée de Brouwer, France.

Subasic, P. & Hirota, K. (1998) – Similarity rules and gradual rules for analogical interpolative reasoning with imprecise data – *Fuzzy Sets and Systems, 96*, 53–75.

Sugeno, M. & Yasukawa, T. (1993) – A fuzzy-logic-based approach to qualitative modeling – *IEEE Transactions on Fuzzy Systems*, *1*, 7–31.

Takagi, T. & Sugeno, M. (1985) – Fuzzy identification of systems and its applications to modeling and control – *IEEE Transactions on System Man and Cybernetics*, *15*, 116–132.

Ughetto, L. (1998) – *Les systèmes à base de règles floues - Vérification de la cohérence et méthodes d'inférence* – Thèse de Doctorat, Université Paul Sabatier, Toulouse.

Wang, L.-X. (1992) – Fuzzy systems are universal approximators. *First IEEE conference on fuzzy systems*, (pp. 1163–1169)., San Diego.

Wang, L.-X. & Mendel, J. M. (1992a) – Fuzzy basis functions, universal approximation, and orthogonal least squares learning – *IEEE Transactions on Neural Networks*, *3*, 807–814.

Wang, L.-X. & Mendel, J. M. (1992b) – Generating fuzzy rules by learning from examples – *IEEE Transactions on Systems, Man and Cybernetics*, *22 (6)*, 1414–1427.

Warwick, K. (1995) – New ideas in fuzzy clustering and fuzzy automata. Steele, N. C. (édité par), *Proceedings of the International ICSC Symposium on Fuzzy logic*, (pp. XXI–XXVII)., Canada / Switzerland. ICSC Academic Press.

Werber, B. (2000) – *L'empire des anges* – Albin Michel.

Wong, C.-C. & Her, S.-M. (1999) – A self-generating method for fuzzy systems design – *Fuzzy Sets and Systems*, *103*, 13–25.

Xie, X. & Beni, G. (1991) – A validity measure for fuzzy clustering – *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13 (8)*, 841–847.

Yager, R. R. & Filev, D. P. (1994) – Generation of fuzzy rules by mountain clustering – *Journal of Intelligent and Fuzzy Systems*, *2*, 209–219.

Yam, Y., Baranyi, P., & Yang, C.-T. (1999) – Reduction of fuzzy rule base via singular value decomposition – *IEEE Transactions on Fuzzy Systems*, *7 (2)*, 120–132.

Yen, J., Wang, L., & Gillepsie, C. W. (1998) – Improving the interpretability of tsk fuzzy models by combining global learning and local learning – *IEEE Transactions on Fuzzy Systems*, *6 (4)*, 530–537.

Zadeh, L. A. (1965) – Fuzzy sets – *Information and Control*, *8*, 338–353.

Annexes

Annexe A

Petit glossaire du flou

- Ensemble flou : Un ensemble flou est défini par sa fonction d'appartenance. Un point de l'univers, x, appartient à un ensemble, A avec un degré d'appartenance, $0 \leq \mu_A(x) \leq 1$.

  La figure A.1 montre un ensemble de forme triangulaire.

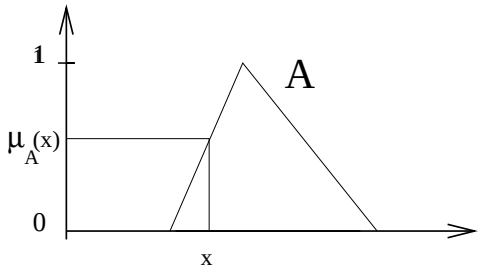


FIG. A.1 – Un ensemble flou de forme triangulaire

- Prototype d'un ensemble : un point est un prototype d'un ensemble flou si son degré d'appartenance à cet ensemble vaut un.

- Opérateurs :
  - *ET* : opérateur de conjonction, noté $\wedge$, les plus employés sont le minimum et le produit.
  - *OU* : opérateur de disjonction, les plus employés sont le maximum et la somme.
  - *est* : la relation x *est* A est quantifiée par le degré d'appartenance de la valeur x au sous-ensemble flou A.

- Partitionnement : Le découpage du domaine de définition d'une variable en sous-ensembles flous est appelée partitionnement. Ces ensembles sont notés A_1, A_2, ...

- Partition floue forte : La partition de la variable X_i sera appelée une partition floue forte si $\forall x \in X_i, \sum_j \mu_{A_j^i}(x) = 1$ et $\forall j \quad \exists x \quad \mu_{A_j^i}(x) = 1$.

- Exemple : un exemple ou individu est formé d'un vecteur d'entrée x de dimension p et, dans le cas général, d'un vecteur y, de dimension q.

- Règle floue : Une règle floue est de la forme ***Si*** *je rencontre telle situation* ***Alors*** *j'en tire telle conclusion.* La situation, appelée prémisse ou antécédent de la règle, est définie par une combinaison de relations de la forme x *est* A pour chacune des composantes du vecteur

d'entrée. La partie conclusion de la règle est appelée conséquence, ou encore simplement conclusion.

- Les règles sont de deux types :

  1. Mamdani : Une règle floue de type Mamdani, dont la conclusion est un ensemble flou, s'écrit :

     $$SI\ x_1\ est\ A_1^{i_1}\ ET\ x_2\ est\ A_2^{i_2}\ \ldots\ ET\ x_p\ est\ A_p^{i_p}\ ALORS\ y_1\ est\ C_1^i\ \ldots\ ET\ y_q\ est\ C_q^i$$

     où A_j^i *et* C_j^i sont des ensembles flous qui définissent le partitionnement des espaces d'entrée et de sortie.

  2. Takagi-Sugeno : Dans le modèle de Sugeno la conclusion de la règle est nette. Celle de la règle i pour la sortie j est calculée comme une fonction linéaire des entrées : $y_j^i = b_{jo}^i + b_{j1}^i x_1 + b_{j2}^i x_2 + \cdots + b_{jp}^i x_p$, également notée : $y_j^i = f_j^i(x)$.

- Règle incomplète : Une règle floue sera dite incomplète si sa prémisse est définie par un sous-ensemble des variables d'entrée seulement. La règle, $SI\ x_2\ est\ A_2^1\ ALORS\ y\ est\ C_2$, est incomplète car la variable x_1 n'intervient pas dans sa définition. Les règles formulées par les experts sont principalement des règles incomplètes. Formellement, une règle incomplète est définie par une combinaison implicite de connecteurs logiques *ET* et *OU* opérant sur l'ensemble des variables. Si l'univers de la variable x_1 est découpée en 3 sous-ensembles flous, la règle incomplète ci-dessus peut aussi s'écrire de la façon suivante :
  $$SI\ \left(x_1\ est\ A_1^1\ OU\ x_1\ est\ A_1^2\ OU\ x_1\ est\ A_1^3\right)\ ET\ x_2\ est\ A_2^1\ ALORS\ y\ est\ C_2.$$

- Degré de vérité : Pour une règle donnée, i, son degré de vérité pour un exemple, également appelé poids, et noté w_i, résulte d'une opération de conjonction des éléments de la prémisse : $w_i = \mu_{A_1^i}(x_1) \wedge \mu_{A_2^i}(x_2) \wedge \ldots \wedge \mu_{A_p^i}(x_p)$, où $\mu_{A_j^i}(x_j)$ est le degré d'appartenance de la valeur x_j à l'ensemble flou A_j^i.

- Activité : Un exemple active une règle, ou bien une règle active un exemple, si le degré de vérité de la règle pour l'exemple est non nul.

- Prototype d'une règle : un exemple est un prototype d'une règle si son degré de vérité pour cette règle vaut un.

- Système d'inférence floue : Un système d'inférence floue est formé de trois blocs comme indiqué sur la figure A.2. Le premier, l'étage de fuzzification transforme les valeurs numériques en degrés d'appartenance aux différents ensembles flous de la partition. Le second bloc est le moteur d'inférence, constitué de l'ensemble des règles. Enfin, un étage de défuzzification permet, si nécessaire, d'inférer une valeur nette, utilisable en commande par exemple, à partir du résultat de l'agrégation des règles. Le nombre de règles du système est noté r.

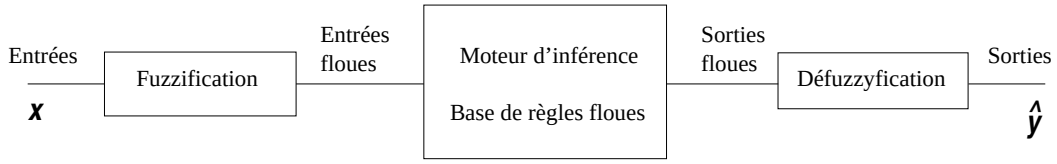


FIG. A.2 – Un système d'inférence floue

- Sortie inférée par le système : elle est notée $\widehat{y_i}$ pour l'exemple i.

- Apprentissage supervisé : L'apprentissage supervisé consiste à induire des relations entre les entrées et la sortie, de dimension un, d'un système à partir d'un ensemble d'exemples. L'ensemble d'apprentissage comprend n exemples.

- Erreur quadratique moyenne : elle est calculée comme $EQM = \frac{1}{n}\sum_{i=1}^{n} \|\widehat{y_i} - y_i\|^2$

- Erreur Moyenne : Contrairement à la précédente, elle est homogène à une mesure. Son expression est $ErM = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \|\widehat{y_i} - y_i\|^2} = \frac{\sqrt{EQM}}{\sqrt{n}}$

Annexe B

Petit glossaire de vinification

Les principaux termes techniques relatifs à la vinification employés dans ce mémoire sont définis ci-dessous :

- fermentation alcoolique : processus de transformation des sucres en alcool. Son évolution est mesurée par la densité, elle est initialement voisine de 1100 et atteint une valeur légèrement inférieure à 1000.
- anthocyanes : composés phénoliques responsables de la couleur des vins rouges et rosés.
- phase pré-fermentaire : phase au cours de laquelle sont extraites les anthocyanes. Durant cette phase la densité évolue lentement.
- phase de fermentation : durant la fermentation proprement dite la densité évolue rapidement. La fermentation est terminée lorsque les sucres sont épuisés, la densité est alors voisine d'une valeur de 997.
- phase post-fermentaire : elle favorise l'extraction de certains tanins.
- phase aqueuse : délimitée par la densité initiale, environ 1100, et la densité de 1050.
- phase alcoolique : succède à la phase aqueuse. Les limites de la densité sont 1050 et 997.
- levures : elles consomment le sucre et le transforment en alcool. Elles se trouvent, à l'état naturel sur la peau du raisin ; elles peuvent aussi être ajoutées.
- enzymage : opération qui consiste à ajouter des enzymes pour favoriser l'extraction en phase aqueuse.
- chapeau : ensemble consistant formé des grappes et enveloppes des grains de raisin. Il flotte sur le moût en fermentation.
- pigeage : opération qui consiste à casser le chapeau à l'aide de râteaux afin de favoriser l'extraction de couleur et d'arômes fruités. Cette opération est automatisée.
- remontage : il s'agit de pomper au bas de la cuve et de réinjecter le jus au dessus du chapeau. Le passage à travers le chapeau favorise l'extraction. Comme le pigeage, le remontage est automatisé.
- délestage : la cuve est entièrement vidée dans une cuve tampon, seul le chapeau repose sur le fond. Le jus est alors réintroduit, et le chapeau, pour remonter, traverse l'ensemble de la cuve.

Annexe C

Le logiciel Fispro

***Fispro* est un logiciel libre de droits, dont la diffusion est assurée par l'INRA : fispro@ensam.inra.fr**

Le logiciel *Fispro* permet la création et la manipulation d'un système d'inférence floue (sif). La bibliothèque de fonctions est écrite en $C++$, une interface en *java* est en cours de développement. La figure C.1 en montre la fenêtre principale. Ce logiciel est portable, il peut s'exécuter sous différentes plates-formes comme Unix, Linux ou Windows. Il destiné à un large public. Les professionnels non informaticiens, par exemple les experts d'un procédé agro-alimentaire, apprécieront la convivialité de l'interface qui permet notamment de créer graphiquement un système d'inférence floue, de saisir des règles, et de visualiser les résultats de l'inférence. C'est un logiciel localisable : les menus, boutons et messages d'erreur sont disponibles en plusieurs langues. Le fichier de configuration est au format texte, il est lisible et modifiable par n'importe quel éditeur, les fichiers d'entrée et de sortie sont compatibles avec les tableurs usuels.

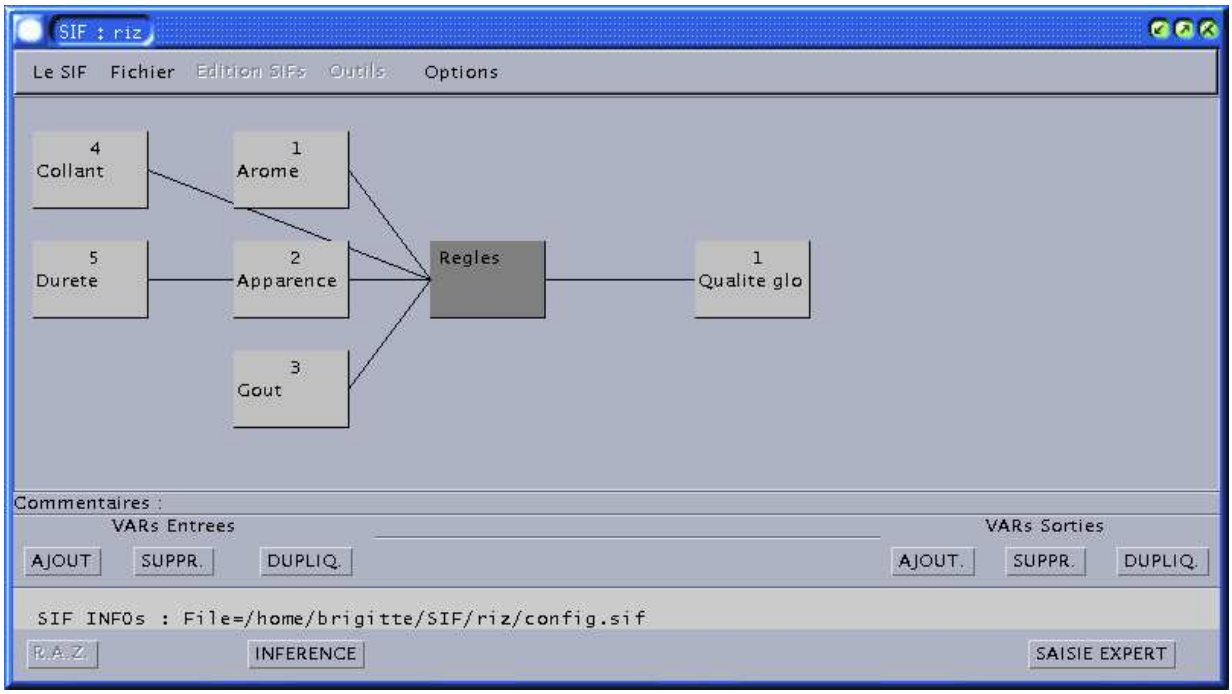


FIG. C.1 – La fenêtre principale

La présente documentation est principalement destinée aux programmeurs. La bibliothèque est formée d'un module de base et de modules spécifiques. Les classes du module de base sont présentées en détail, leur hiérarchie est illustrée sur la figure C.2. Des classes dérivées, dédiées à l'apprentissage à partir de données, ont été définies. Elles implémentent des méthodes d'induction de règles floues telles que celle basée sur le partitionnement ascendant hiérarchique, les arbres de décision flous ou encore la sélection des règles par régression linéaire. Elles sont sommairement présentées dans la section C.4.

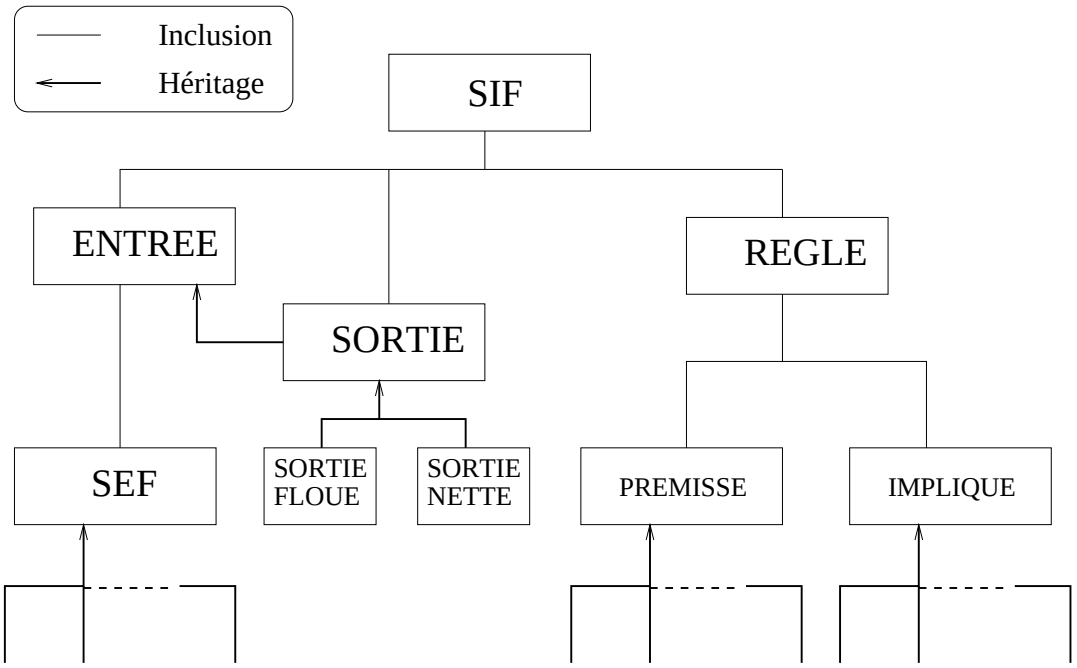


FIG. C.2 – La hiérarchie des classes de base

C.1 Le module de base

Les classes de base sont définies dans les fichiers suivants :

- $sif.h$: pour la description des classes SEF, $ENTREE$, $SORTIE$, $REGLE$, SIF. Les classes SEF et $SORTIE$ sont des classes abstraites. La sortie peut être soit *nette*, soit $floue$. Ces deux sous-classes sont définies dans $sif.h$.
- $sef.h$: la description des différents types de sous-ensembles flous disponibles, les classes dérivées de la classe abstraite SEF.
- *regle.h* : la description des classes nécessaires pour la classe $REGLE$, les classes $PREMISSE$ et $IMPLIQUE$.
- *commun.h* : la description des fonctions communes, hors classe, ainsi que les paramètres de base de configuration du SIF.

C.2 Fonctionnement du SIF

Les SIF gérés par cette bibliothèque sont de type $MIMO$, Multiple Input Multiple Output. Le SIF est construit à partir des informations contenues dans le fichier de configuration. La fonction principale du module de base est la fonction $Infere()$ de la classe SIF. Pour un vecteur d'entrée

d'un exemple du fichier de données elle infère un vecteur de sortie. Les étapes de l'inférence floue sont les suivantes :

1. Fuzzification : cette étape est réalisée au sein de la classe $ENTREE$. Ainsi, pour une valeur donnée, la fonction $ValeurAppartenance()$ remplit le tableau $VecteurAppartenance$ qui contient les coefficients d'appartenance de la valeur à chacun des sous-ensembles flous de l'entrée. Ce tableau est public.

2. Inférence : la conséquence de chacune des règles est pondérée par le degré de vérité de la règle pour l'exemple. Les règles, pour un système de deux entrées et une sortie, peuvent être de la forme :

   SI Entrée 1 est Sef 2 ET Entrée 2 est Sef 1 ALORS Sortie 1 est Valeur 1

   SI Entrée 1 est Sef 1 ET Entrée 2 est Sef 1 ALORS Sortie 1 est Sef 2

   Cette étape est assurée au sein de la classe $REGLE$ par la fonction $ExecRegle()$. Cette fonction est composée de deux parties.

   La première partie calcule le degré de vérité de la règle pour l'exemple, au sein de la classe $PREMISSE$. Il s'agit d'une opération de conjonction des degrés d'appartenance des valeurs des différentes variables d'entrée ($Entre$ 1 et $Entre$ 2) pour les sous-ensembles flous qui décrivent la prémisse de la règle (2 et 1 pour la première règle ; 1 et 1 pour la seconde). La classe $PREMISSE$ est construite avec un pointeur sur le tableau des entrées qui lui permet d'accéder au champ $VecteurAppartenance$ des différentes entrées.

   La seconde calcule la conclusion de la règle pour l'exemple au sein de la classe $IMPLIQUE$. Si la sortie est une sortie de nature *nette*, la valeur de sortie qui décrit la règle est interprétée comme une constante qui est simplement multipliée par le degré de vérité. Si la sortie est de nature $floue$, la valeur de sortie de la règle indique le numéro d'ordre de l'ensemble flou de la partition. Dans ce cas, la pondération renvoie l'alpha coupe, du niveau du degré de vérité, du sous-ensemble flou activé par la règle. Cette alpha coupe est caractérisée par un centre de gravité, une masse qui correspond à son aire et quatre points qui définissent un trapèze. Quelle que soit la forme du sous-ensemble flou, le polygone défini par le support et le noyau du niveau de l'alpha coupe est modélisé par un trapèze. Le constructeur de $IMPLIQUE$ contient un pointeur sur le tableau des sorties.

   Les valeurs calculées sont des données publiques de la classe $REGLE$:
   - $Poids$: le degré de vérité ;
   - $ValActions$: la conclusion pondérée pour la sortie nette, le centre de gravité de l'alpha coupe pour la sortie floue ;
   - $ValMasses$: l'aire de l'alpha coupe pour la sortie floue, le degré de vérité pour la sortie nette ;
   - $TrapCoupes$: les coordonnées du trapèze de l'alpha coupe, elles sont toutes nulles pour une sortie nette.

3. Défuzzification : La fonction de défuzzification consiste à agréger les conclusions des différentes règles activées par l'exemple et d'en déduire une valeur numérique, la sortie inférée par le SIF pour l'exemple courant.

   Le type d'opérateur de défuzzification est un des paramètres de la classe $SORTIE$. Deux sont actuellement définis au sein de la classe SIF :
   - $sugeno$: sa formule est la suivante : $\frac{\sum_{Regles} ValActions}{\sum_{Regles} Poids}$;

– *aires* : il s'agit de la méthode des aires dont la formule est :

$$\frac{\sum_{Regles} ValMasses * ValActions}{\sum_{Regles} ValMasses}$$

Cet opérateur ressemble à la méthode du centre de gravité. Dans la méthode des aires, l'intersection de deux alpha coupes est comptabilisée deux fois dans la masse, l'aire est donc surestimée.

Les opérateurs de défuzzification sont définis pour être compatibles avec l'un ou l'autre des types de sortie : sortie nette pour l'opérateur de Sugeno, sortie floue pour la méthode des aires. En cas d'incompatibilité, un message d'erreur est affiché et les sorties inférées sont égales à la sortie par défaut ('`SORTIE_DEFAUT`' défini dans *commun.h*, actuellement comme -1.0).

La fonction $Performance()$ s'utilise lorsque l'on veut comparer la sortie réellement observée à celle inférée par le système pour un ensemble d'exemples. Elle est appelée avec deux arguments principaux : le premier est le numéro de la sortie à tester, le second est l'ensemble des données d'entrée et de sortie. Celui peut être fourni sous la forme d'un fichier ou bien d'un pointeur sur un tableau de '`double`'. La performance d'un système est mesurée par la moyenne de la racine carrée de la somme des carrés des erreurs, lorsque la sortie est continue ; ou bien par le nombre d'exemples mal classées en classification. La sortie est un tableau, compatible avec les tableurs usuels, de trois colonnes indiquant pour chacun des exemples la sortie observée, la sortie inférée ainsi que l'écart entre les deux. Cette fonction renseigne également le champ *NbBlancs* correspondant au nombre d'exemples non gérés par le système, c'est-à-dire dont le degré de vérité cumulé sur l'ensemble des règles est inférieur au paramètre *MuSeuil*.

C.3 Structure du fichier de configuration

Un SIF est complètement décrit par un fichier de configuration lisible. Ce fichier est composé de plusieurs sections séparées par une ou plusieurs lignes vides. Attention il faut que ces lignes ne contiennent aucun caractère, pas même un blanc. Chacune des sections démarre par une balise, celle ci doit être placée en début de ligne et être rigoureusement identique à la description.

Les chaînes de caractères sont délimitées par le symbole simple quote `'chaîne'`, le séparateur de champs est la virgule. Ces délimiteurs sont paramétrables, ils sont définis dans "commun.h" par les constantes `DELIM_CHAINE` et `SEPARE`.

C.3.1 L'interface

La première section gère les paramètres de l'interface utilisateur. La balise est **[Interface]**. Comme seuls ces caractères sont utilisés pour l'authentifier, ils peuvent être suivis de commentaires.

Exemple d'interface :

[Interface]
DelimChaine="
Separe=,
LongueurMaxNom=40
LongueurMaxType=20
CaracteresMaxLigne=100
NbBornesMax=10

Les deux premiers champs correspondent aux séparateurs utilisés dans le fichier de données. Ils sont définis par un caractère situé immédiatement après le signe égale. Le caractère `ESPACE` n'est pas un séparateur valide.

`LongueurMaxNom` et `LongueurMaxType` délimitent la longueur maximale des chaînes de caractères utilisées dans la description des entrées, des sorties et des sous-ensembles flous.

`CaracteresMaxLigne` indique la dimension du tampon utilisé pour la lecture d'une ligne du fichier de configuration ou du fichier de données. Attention : ces valeurs ont des incidences sur les allocations mémoire, elles doivent être correctement dimensionnées.

`NbBornesMax` est le nombre de maximal d'arguments pour la construction des sous-ensembles flous, il correspond de fait au nombre maximal de valeurs possibles pour les sous-ensembles flous discrets.

C.3.2 L'entête

La deuxième section est l'entête. La balise est **[Systeme]**.

Exemple d'entête :
[Systeme]
Nom='Mon système'
NbEntrees=2
NbSorties=2
NbRegles=5
NbExceptions=0
Conjonction='produit'
ValeursManquantes='moyenne'

La fonction de l'entête est de décrire la structure du système d'inférence floue, comme le nombre des entrées et des sorties, mais aussi le type des opérateurs paramétrables. Ces informations déterminent le contenu du fichier de configuration. Le début de chaque ligne commence par une chaîne de caractères à laquelle est accolé un signe égale. Il est impératif de respecter le contenu de ces chaînes et de les faire suivre sans espace du signe '='. La première ligne correspond au nom.

Suivent les lignes indiquant le nombre des entrées, des sorties, des règles et des exceptions. La signification de chacun de ces chiffres (des entiers) est donnée dans la section correspondante.

'`Conjonction=`' décrit le type de l'opérateur de conjonction choisi pour calculer le degré de vérité de la règle. Trois sont disponibles, le produit ('`produit`'), le minimum ('`min`') et celui de Lukasiewicz ('`luka`').

'`ValeursManquantes=`' définit la stratégie de gestion des valeurs manquantes. Deux sont possibles : soit `moyenne`, la valeur d'appartenance à chacune des modalités est égale et leur somme vaut 0.5 ; soit `aleatoire`, la valeur manquante résulte d'un tirage aléatoire uniforme dans le domaine de variation, puis est ensuite projetée dans les différents sous-ensembles flous.

C.3.3 Les entrées

Après l'entête, le fichier doit être composé d'un nombre de sections d'entrée correspondant au nombre d'entrées spécifié dans l'entête.

Les numéros d'ordre dans ce fichier, pour les entrées, les sous-ensembles flous ou les sorties, commencent toujours par 1. Le caractère '`?`' représente le numéro d'ordre.

Exemple de section d'entrée :
[Entree1]
Active='oui'
Nom='Var 1'
Domaine=[10 , 25.5]
NbSef=3
Sef1='petit','triangle',[10 , 12 , 14]
Sef2='moyen','gauss',[1.2 , 16]
Sef3='grand','trapeze',[15, 18 , 22 , 24]

La balise est **[Entree ?]**. L'entrée peut être active ou inactive, `Active='non'`. Elle est identifiée par son nom. La ligne suivante spécifie le domaine de variation, également appelé univers du discours. Les bornes du domaine sont utilisées pour limiter le tirage aléatoire dans le cas de valeurs manquantes. D'autre part, le logiciel signalera les valeurs des exemples situées hors domaine comme une alarme. Le domaine est défini par deux nombres en virgule flottante séparés par un séparateur de champ (actuellement une virgule) et encadrés par des crochets droits. Comme pour le délimiteur de chaîne, les délimiteurs de suites de nombres, `DEBUT_NOMBRE` et `FIN_NOMBRE`, et le délimiteur de champ, `SEPARE`, sont définis dans "commun.h".

Puis viennent les lignes qui définissent le partitionnement de l'entrée. Le nombre de sous-ensembles flous est un entier. Cette ligne est suivie d'un nombre variable de lignes, chacune décrivant un sous-ensemble flou. Un sous-ensemble flou est identifiée par '`Sef ?=`'. Sur la même ligne, figurent dans l'ordre une chaîne de caractères correspondant au nom (label linguistique), une chaîne de caractères correspondant au type (à la forme) de la fonction d'appartenance et une suite de nombres en virgule flottante pour indiquer les paramètres de la fonction d'appartenance. Chacun de ces champs est délimité par `SEPARE`.

La longueur maximale des noms ainsi que celle des types sont définies dans la section `Interface`.

Les types disponibles sont (le nombre entre parenthèses indique le nombre d'arguments nécessaires pour les instancier) : '`triangle`' (3 ou 2 pour le triangle symétrique), '`trapeze`' (4), '`DemiTrapezeInf`' (3), '`DemiTrapezeSup`' (3), '`gbell`' (3), '`gauss` (2)' et '`discrete`'. Les arguments attendus pour ce dernier type sont un nombre variable d'entiers. Ce nombre est limité par la valeur `NbBornesMax`, définie dans la section `Interface`.

C.3.4 Les sorties

La section correspondante à la sortie commence par la balise **[Sortie ?]**. Les quatre lignes immédiatement suivantes sont spécifiques à la sortie :

La ligne '`Nature=`' indique s'il s'agit d'une sortie '`nette`' ou '`floue`'. La ligne '`Defuzzification=`' indique l'opérateur de défuzzification associé à la sortie : '`sugeno`', '`aires`', ou plus tard '`centroid`'. La ligne '`ValeurDefaut=`' indique la valeur, nombre en virgule flottante, à affecter à la sortie lorsqu'aucune règle n'est activée par l'exemple en cours. La ligne '`Classif=`' indique si la sortie est une classe, valeur '`oui`', ou bien continue, '`non`'. Cette information sera utilisée par la fonction '`Performance`'.

Comme la structure de la sortie hérite de celle de l'entrée, les lignes suivantes sont identiques à celles décrites dans la section entrée. Pour une sortie nette il est naturel d'indiquer un nombre de sous-ensembles flous nul. Toutefois, si ce nombre n'est pas nul, l'absence du nombre de ligne correspondant à la description de chacun des sous-ensembles flous provoquera une erreur. Imposer un nombre de sous-ensembles flous nul dans le cas d'une sortie nette nous aurait privé du lien d'héritage entre la sortie et l'entrée, et de la réutilisation des fonctions correspondantes. Attention les messages d'erreur indiquant *Erreur dans le constructeur de ENTREE %d* peuvent se rapporter aussi bien à l'entrée qu'à la sortie portant le numéro en question.

C.3.5 Les règles

La balise **[Regles]** indique le début de la section correspondante. Une règle est décrite par une suite d'entiers et une suite de nombres en virgule flottante correspondant aux actions à associer à chacune des sorties. La suite des entiers définit la prémisse de la règle : le premier entier correspond au numéro d'ordre du sous-ensemble flou de l'entrée 1, et ainsi de suite pour les autres entrées. Chacun des nombres est séparé de son voisin par le délimiteur `SEPARE`. Une ligne décrit une, et une seule, règle.

Il y a deux façons de définir la base de règles actives :

1. Première possibilité : Insérer la description des règles, après la balise, dans le fichier de configuration. Le nombre de lignes, donc de règles, doit être en accord avec le nombre indiqué en entête.
2. Deuxième possibilité : Indiquer un nom de fichier, une chaîne délimitée `'NomDuFichierRegles'`, contenant les règles stockées sous la même forme. Le nombre de lignes du fichier doit être en accord avec le nombre de règles spécifié dans l'entête.

Il est parfois souhaitable de définir des règles incomplètes, c'est-à-dire dont la prémisse n'implique qu'un sous-ensemble des variables d'entrées disponibles. Il suffit pour cela d'indiquer le numéro le numéro d'ordre 0 pour la variable inactive dans la règle. La règle 2, 0, 1, 23.5 correspondant à un système à trois entrées et une sortie s'énonce de la façon suivante, l'entrée deux étant inactive :

SI Entrée 1 est Sef 2 ET Entrée 3 est Sef 1 ALORS Sortie 1 vaut 23.5

Nous disposons d'un utilitaire permettant de générer les règles correspondant à un partitionnement des entrées défini par un fichier de configuration. Dans ce cas le nombre de règles indiqué dans l'entête n'est pas pris en compte. Il doit tout de même être non nul. Ce programme, *genere*, permet de générer les prémisses des règles correspondant à l'ensemble des combinaisons des entrées. Il a besoin du nom du fichier de configuration en argument. Il admet comme argument optionnel le nom d'un fichier de données. Si celui-ci est fourni, le programme ne conservera que les règles qui seront activées par au moins un exemple du fichier de données. La sortie de ce programme est un fichier de configuration du sif complet, incluant la description des règles dans un fichier séparé.

C.3.6 Les exceptions

Une exception est une règle inactive. Il peut être utile de gérer des exceptions sans avoir à modifier le reste de la configuration. Leur nombre est indiqué dans l'entête '`NbExceptions=?`'. La balise, **[Exceptions]**, doit toujours figurer dans le fichier même s'il n'y a aucune exception. Une exception est donc décrite comme une règle. Une erreur de lecture d'une exception sera indiquée comme une erreur dans la construction d'un objet *REGLE*.

L'utilitaire *genere* tient compte des exceptions pour générer les règles et les exclut du fichier des règles générées. L'utilitaire *genere* ne travaille qu'avec les prémisses. Une exception peut, comme

dans le cas précédent, décrire complètement la règle, ou bien, en utilisant la valeur 0, qui correspond à la notion de variable inactive, être relative à une famille de règles. L'exception suivante : `1, 2, 0` pour un système à 3 entrées, indique que toutes les règles pour lesquelles l'entrée 1 aura la valeur correspondant au sous-ensemble flou 1, et l'entrée 2 au sous-ensemble flou 2, quelles que soient les combinaisons possibles avec l'entrée 3, ne seront pas générées. Il peut y avoir autant de variables inactives que d'entrées dans la même exception, sur une même ligne.

C.4 Les classes dérivées de la classe SIF

Le module de base est conçu pour faciliter l'implémentation de classes dérivées. Les modules disponibles, en plus de celui déjà mentionné utilisé pour la génération des règles, visent principalement à la construction de systèmes d'inférence floue à partir de données.

La classe $SIFOPT$ permet d'initialiser la conclusion de règles, celle ci pouvant être *nette* ou $floue$, de type $classif$ ou non, à partir d'un ensemble d'apprentissage.

La classe $SIFCLUST$ implémente la génération d'une famille de partitions pour chacune des variables d'entrée et permet de construire le sif, multidimensionnel, formé de règles définies par les mêmes variables le plus performant parmi les plus compacts.

La classe $SIFSIMPLE$ permet de simplifier une base de règles, pour obtenir un nombre de règles plus réduit et des règles incomplètes, en acceptant une dégradation de la performance dans des limites paramétrables.

La classe $SIFOLS$ initialise une règles par exemple et sélectionne les plus influentes par régression linéaire.

La classe $SIFTREE$ implémente l'induction de règles par les arbres de décision flous.

La classe $LIENSIF$ fournit un utilitaire de travail qui permet d'établir les liens existants entre les règles et les exemples. Le résultat consiste en la création de plusieurs fichiers, notamment le fichier *regles.exemples* qui indique pour chacune des règles la liste des exemples qui l'activent ; et le fichier *exemples.regles* qui, pour chaque exemple, indique la liste des règles activées.

Annexe D

Publications liées à ce mémoire

Guillaume, S. (2001) – Designing fuzzy inference systems from data : an interpretability-oriented review – *IEEE Transactions on Fuzzy Systems*, *9 (3)*, 426–443.

Guillaume, S. & Charnomordic, B. (2001c) – Knowledge discovery for control purposes in food industry databases – *Fuzzy sets and Systems*, *122 (3)*, 487–497.

Guillaume, S. & Charnomordic, B. (2001a) – Generating an interpretable family of fuzzy partitions – *IEEE Transactions on Fuzzy Systems*, *Submitted March 2001.*

Congrès LFA :

Guillaume, S. & Charnomordic, B. (1999) – Dégager des règles de conduite d'un procédé agro-alimentaire à partir d'une base d'exemples. *LFA'99*, (pp. 49–55)., Toulouse, France. Cépaduès Editions.

Guillaume, S. & Charnomordic, B. (2000b) – Une revue des principales méthodes d'induction de règles à partir des données suivant le critère de l'interprétabilité. *LFA'2000*, (pp. 309–316)., Toulouse, France. Cépaduès Editions.

Guillaume, S. & Charnomordic, B. (2001b) – Induire un partitionnement flou interprétable. *LFA'01*, En cours de publication.

Congrès Fuzz'Ieee :

Guillaume, S., Charnomordic, B., & Titli, A. (2001) – A distance metric suitable for fuzzy partitioning. *FUZZ'IEEE 2001*, En cours de publication.

Congrès Agro alimentaire :

Guillaume, S. & Charnomordic, B. (2000a) – Process simulation using fuzzy rules inferred from data. Thiel, D. (édité par), *FoodSim2000*, (pp. 29–32). Society for Computer Simulation International.

Annexe E

Orthogonalisation des moindres carrés

Soit le modèle de régression :

$$d(k) = \sum_{i=1}^{M} p_i(k)\ \theta_i + \epsilon(k)$$

où $d(k)$ est la sortie désirée, θ_i sont les paramètres à optimiser et $p_i(k)$ les régresseurs. Ce sont des fonctions de $x(k)$, $p_i(k) = p_i(x(k))$.

La fonction d'erreur, $\epsilon(t)$ est supposée non corrélée aux régresseurs.

Un régresseur correspond à la partie prémisse d'une règle du SIF et les paramètres à optimiser correspondent à la partie conséquence. Au départ, il y a autant de régresseurs que d'exemples ($M = N$). Pour un exemple donné, les centres des gaussiennes correspondantes sont les coordonnées de cet exemple dans les différentes dimensions de l'espace d'entrée, l'écart type est calculé pour chaque dimension en fonction de la distribution. Pour une coordonnée donnée, il est identique pour l'ensemble des exemples.

En écriture matricielle, l'équation devient :

$$d = P\theta + E \tag{E.1}$$

avec d et E, vecteurs colonnes de dimension N, θ vecteur colonne de dimension M et P, une matrice de dimension $N \times M$.

$$d = \begin{bmatrix} d(1) & \cdots & d(N) \end{bmatrix}^T$$

$$P = \begin{bmatrix} p_1 & \cdots & p_M \end{bmatrix} \quad avec \quad p_i = \begin{bmatrix} p_i(1) & \cdots & p_i(N) \end{bmatrix}^T$$

L'OLS transforme l'ensemble des vecteurs p_i en un ensemble de vecteurs orthogonaux. Ainsi, il sera possible d'évaluer leur contribution individuelle en terme d'énergie ou de variance.

Pour cela la matrice P est décomposée en une matrice orthogonale W et une une matrice triangulaire supérieure d'éléments diagonaux égaux à 1, A. Soit $P = WA$.

La matrice W est de dimension $N \times M$ et telle que $W^T W = H$, où H est une matrice diagonale carrée de taille M et de terme général :

$$h_i = \sum_{t=1}^{N} w_i(t) w_i(t) = w_i^T w_i$$

La matrice A est de terme général α_{ij}, où α_{ij} représente la part du vecteur j prise en compte par le vecteur i déjà construit.

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1M} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2M} \\ 0 & 0 & & & \vdots \\ \vdots & & & \ddots & \\ & & & 1 & \alpha_{(M-1)M} \\ 0 & & \cdots & 0 & 1 \end{bmatrix}$$

Le procédé de Gram-Schmidt, qui permet de construire une base de M vecteurs orthogonaux w à partir de M vecteurs de base p, est le suivant :

$$\begin{aligned} w_1 &= p_1 \\ w_2 &= p_2 - \frac{< w_1, p_2 >}{\|w_1\|^2}\, w_1 \\ &\vdots \\ w_i &= p_i - \frac{< w_1, p_i >}{\|w_1\|^2}\, w_1 - \frac{< w_2, p_i >}{\|w_2\|^2}\, w_2 - \cdots - \frac{< w_{i-1}, p_i >}{\|w_{i-1}\|^2}\, w_{i-1} \end{aligned}$$

Le terme $\frac{<w_1,p_2>}{\|w_1\|^2}\, w_1$ correspond à la projection du vecteur p_2 sur le vecteur déjà sélectionné w_1. Le quotient du produit scalaire par la norme, $\frac{<w_1,p_2>}{\|w_1\|}$, exprime la longueur signée de la projection, tandis que le quotient $\frac{w_1}{\|w_1\|}$ indique la direction du vecteur unitaire w_1. Le terme α_{12} vaut donc $\frac{<w_1,p_2>}{\|w_1\|^2}$.

L'équation (E.1) devient :

$$d = W g + E \tag{E.2}$$

La solution de l'équation (E.2) donnée par les moindres carrés est :

$$\widehat{g} = H^{-1} W^T d$$

$\widehat{g}$ est un vecteur colonne de dimension M, de terme général :

$$\widehat{g_i} = \frac{w_i^T d}{w_i^T w_i}$$

Les paramètres θ s'obtiennent facilement à partir de la relation : $A\widehat{\theta} = \widehat{g}$

La somme des carrés des sorties désirées, ou énergie de $d(k)$ s'écrit :

$$d^T d = \sum_{i=1}^{M} g_i^2 w_i^T w_i + E^T E$$

car comme les vecteurs w_i sont deux à deux orthogonaux, $w_i^T w_j = 0$, $\forall i \neq j$.

(Rappel : la variance est l'énergie divisée par le nombre de points, ici N).

Cette énergie est décomposée en deux termes : la part expliquée par les M régresseurs et la part inexpliquée, $E^T E$.

La part de variance expliquée par un régresseur i, sera normée par la variance totale et utilisée comme critère de sélection. Pour minimiser l'erreur, il faut maximiser le critère suivant :

$$[err]_i = \frac{g_i^2 \; w_i^T w_i}{d^T d}$$

La composante ajoutée, w_i, sera donc la composante orthogonale aux précédentes qui maximise la variance expliquée sur les sorties.

Dans le cas des deux exemples identiques, si le descripteur p_i correspondant à l'un deux a été sélectionné, le second ne le sera pas car le terme α_{ij} est égal au descripteur lui-même et le vecteur $w_k^{(i)}$ associé, donc le critère de réduction d'erreur, est nul.

Les vecteurs w_i sont donc :

$$\begin{aligned} w_1 &= p_1 \\ w_k &= p_k - \sum_{i=1}^{k-1} \alpha_{ik} w_i, \qquad k = 2, \ldots, M \end{aligned}$$

$$avec \quad \alpha_{ik} = \frac{w_i^T p_k}{w_i^T w_i}, \qquad 1 \leq i \leq k$$

L'algorithme est le suivant :

A la première étape, sélectionner parmi les M vecteurs celui qui explique le plus de variance. Pour cela calculer pour $1 \leq i \leq M$:

$$\begin{aligned} w_1^{(i)} &= p_i \\ g_1^{(i)} &= (w_1^{(i)})^T d / (w_1^{(i)})^T (w_1^{(i)}) \end{aligned}$$

$$[err]_1^{(i)} = (g_1^{(i)})^2 (w_1^{(i)})^T (w_1^{(i)}) / d^T d$$

et sélectionner $w_1 = w_1^{(i_1)} = p_{i_1}$ tel que $[err]_1^{(i_1)} = max\{[err]_1^{(i)}, \quad 1 \leq i \leq M\}$

A l'étape k, $k \geq 2$, pour $1 \leq i \leq M, i \neq i_1, \cdots, i \neq i_{k-1}$, calculer :

$$\begin{aligned} \alpha_{jk}^{(i)} &= w_j^T p_i / w_j^T w_j \\ w_k^{(i)} &= p_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} w_j \\ g_k^{(i)} &= (w_k^{(i)})^T d / (w_k^{(i)})^T (w_k^{(i)}) \end{aligned}$$

$$[err]_k^{(i)} = (g_k^{(i)})^2 (w_k^{(i)})^T (w_k^{(i)}) / d^T d$$

et sélectionner

$$w_k = w_k^{(i_k)} = p_{i_k} - \sum_{j=1}^{k-1} \alpha_{jk} w_j$$

tel que $[err]_k^{(i_k)}$ est maximum.

La procédure s'arrête à l'étape M_s lorsque :

$$1 - \sum_{j=1}^{M_s} [err]_j < seuil$$

Annexe F

Complément sur l'application riz

L'étude du système de décision pour le précédé de vinification nous a conduit à proposer une autre voie pour la construction de systèmes formés de règles définies par les mêmes variables à partir des hiérarchies de partitions. Le système initial ne comprend plus deux sous-ensembles flous par entrée, comme dans l'approche étudiée dans la section 4.4.2, mais un seul. Les variables sont donc introduites progressivement dans le système.

Les résultats sont rassemblés dans la tableau F.1[a]. Ils sont à rapprocher de leurs homologues, obtenus avec un système initial formé de deux sous-ensembles flous par variable d'entrée, synthétisés dans le tableau 4.16.

	Indicateurs des différents systèmes								
	Nb Sef	Nb Sef Sortie	Max R	#R	ErM	ErMax	#I	#E	Max E
C1	2 0 3 3 0	5	18	10	0.00800	-0.266	0		
C1 (si)				5	0.00814	0.252	0	0	0
C2	2 0 4 2 0	5	16	9	0.00806	0.250	1		
C2 (si)				7	0.00873	0.259	16	0	0

TAB. F.1 – Résultats des configurations testées, avec conclusion floue, sur le riz

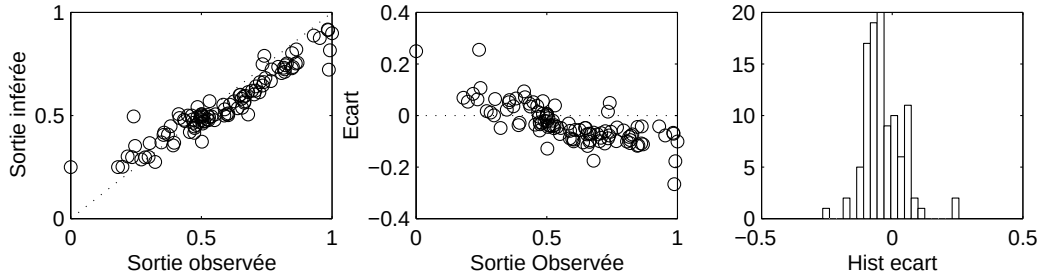
Les trois entrées sélectionnées sont l'arôme, le goût et le collant. L'apparence et la dureté semblent être de moindre importance, ce qui confirme les conclusions de l'étude réalisée dans la section 4.4.2.

Une représentation des performances des systèmes sélectionné et simplifié, respectivement dans les parties supérieure et inférieure des graphiques, issus des deux configurations est donnée sur les figures F.1 et F.2. Rappelons que l'effectif minimum pour initialiser une règle est d'un seul individu, les deux configurations différent par la valeur du degré de vérité cumulé minimum que doit présenter une règle pour la base d'exemples : 0.3 pour la première configuration, 0.5 pour la seconde.

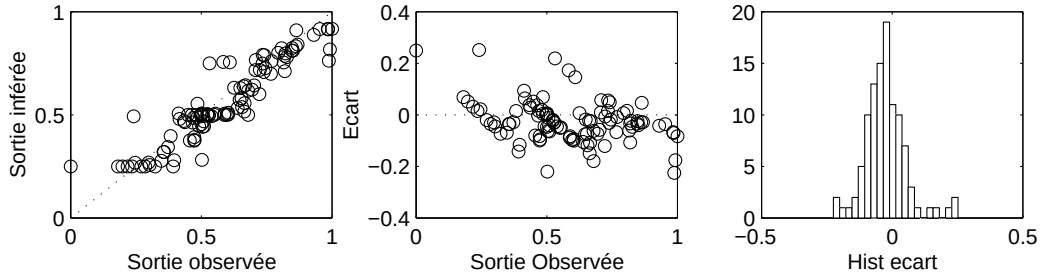
Le système correspondant à la deuxième configuration est difficile à simplifier. Le découpage trop fin de la troisième entrée, le goût, induit un nombre d'exemples inactivés important si trop de règles sont supprimées.

Le système qui nous semble réaliser le meilleur compromis entre la précision et la lisibilité est celui correspondant à la première configuration après simplification. Les valeurs des indices d'erreurs sont acceptables et le nombre de règles est petit.

[a] Les indicateurs utilisés sont définis dans le tableau 4.1.

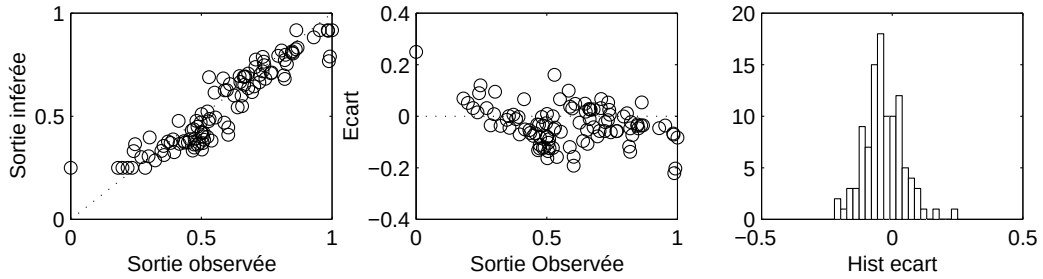


a) Système avant simplification

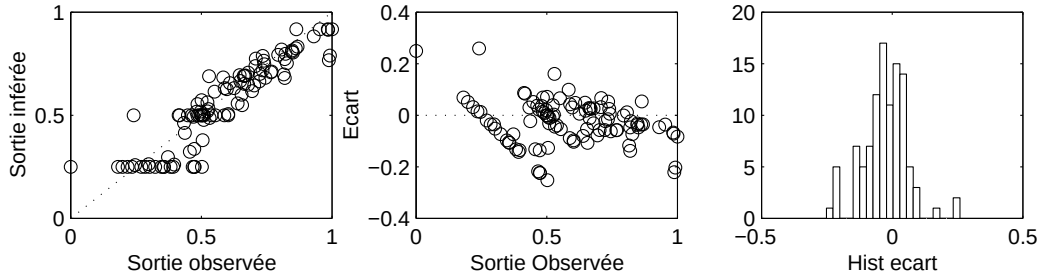


b) Système après simplification

FIG. F.1 – Représentation des erreurs de prédiction pour le système obtenu avec la première configuration lorsque l'initialisation est réalisée avec un seul sous-ensemble flou par entrée pour le riz



a) Système avant simplification



b) Système après simplification

FIG. F.2 – Représentation des erreurs de prédiction pour le système obtenu avec la deuxième configuration lorsque l'initialisation est réalisée avec un seul sous-ensemble flou par entrée pour le riz

La lisibilité des règles, indiquées dans le tableau F.2, s'en trouve nettement accrue.

Base de règles des deux configurations pour le riz :
conclusions floues et initialisation avec un sel sous-ensemble flou

	Règles complètes		Simplifié	
C1:				
	2, 0, 3, 2, 0,	4	2, 0, 3, 3, 0,	5
	2, 0, 3, 3, 0,	5	1, 0, 1, 1, 0,	2
	1, 0, 1, 1, 0,	2	1, 0, 2, 2, 0,	3
	1, 0, 1, 2, 0,	3	1, 0, 3, 3, 0,	4
	1, 0, 2, 1, 0,	3	2, 0, 2, 3, 0,	4
	1, 0, 2, 2, 0,	3		
	2, 0, 2, 1, 0,	2		
	2, 0, 2, 2, 0,	3		
	1, 0, 3, 3, 0,	4		
	2, 0, 2, 3, 0,	4		
C2:				
	2, 0, 4, 2, 0,	5	2, 0, 4, 2, 0,	5
	1, 0, 1, 1, 0,	2	1, 0, 1, 1, 0,	2
	1, 0, 2, 1, 0,	3	2, 0, 2, 1, 0,	2
	1, 0, 2, 2, 0,	2	2, 0, 2, 2, 0,	3
	2, 0, 2, 1, 0,	2	1, 0, 3, 2, 0,	3
	2, 0, 2, 2, 0,	3	2, 0, 3, 2, 0,	4
	1, 0, 3, 2, 0,	3	1, 0, 4, 2, 0,	4
	2, 0, 3, 2, 0,	4		
	1, 0, 4, 2, 0,	4		

TAB. F.2 – Règles des systèmes sélectionnés et simplifiés pour le jeu de données Riz, la sortie est découpée en cinq ensembles flous, le système le plus simple est initialisé avec un seul sous-ensemble flou par entrée.

Le tableau F.3 traduit les règles du système simplifié de la configuration $C1$ en termes linguistiques.

Les riz qui sont évalués comme globalement bons sont ceux qui ont à la fois de l'arôme, du goût et du collant. Dès qu'une valeur de l'une de ces caractéristiques s'affaiblit, la qualité est jugée moins bonne, comme en témoignent les règles quatre, l'arôme est moins bon, et cinq, le riz a moins de goût. Si ce sont deux de ces caractéristiques qui sont en baisse, la qualité globale devient moyenne, comme le montre la règle trois. La deuxième règle nous indique que si tous ces attributs présentent des valeurs faibles, alors la qualité est faible.

Les règles obtenues par Cordon et Herrera (Cordon & Herrera, 2000) sont également interprétables. Elles sont indiquées dans le tableau F.4.

L'analyse de ces règles conduit à des conclusions voisines de celles que nous connaissons. Notons tout d'abord que la sortie est découpée en deux sous-ensembles flous seulement, les étiquettes sont moins riches qu'avec un découpage plus fin. Néanmoins, ce découpage illustre aussi la capacité d'interpolation de tels systèmes. Rappelons que les performances correspondantes sont décrites dans la section 4.4.2.

Arôme	Apparence	Goût	Collant	Dureté	Qualité
Bon	×	Bon	Très collant	×	Très Élevée
Mauvais	×	Mauvais	Pas collant	×	Faible
Mauvais	×	Moyen	Un peu collant	×	Moyenne
Mauvais	×	Bon	Très collant	×	Élevée
Bon	×	Moyen	Très collant	×	Élevée

TAB. F.3 – Règles relatives à la qualité du riz éditées sous forme linguistique

Arôme	Apparence	Goût	Collant	Dureté	Qualité
Bon	Bon	Bon	Collant	Tendre	Élevée
Bon	Bon	Bon	Collant	Dur	Élevée
Mauvais	Bon	Bon	Non collant	Tendre	Élevée
Bon	Mauvais	Mauvais	Non collant	Tendre	Faible
Mauvais	Mauvais	Mauvais	Non collant	Dur	Faible
Mauvais	Bon	Mauvais	Non collant	Tendre	Faible

TAB. F.4 – Règles obtenues par Cordon et Herrera pour la qualité du riz

Ces six règles montrent une relation directe entre le goût et la qualité globale. La même remarque, un peu plus nuancée, peut être faite à propos de l'arôme, de l'apparence et du collant. Les deux premières règles diffèrent seulement par la valeur de la dureté. Ces règles pourraient être fusionnées pour former une règle incomplète. Elles montrent que la dureté, qui n'a pas été sélectionnée dans notre système, est de moindre importance pour expliquer la qualité.

Thèse INSA, École doctorale systèmes, option systèmes informatiques

Nom : Serge GUILLAUME

Date : 14 novembre 2001

Titre : Induction de règles floues interprétables

Résumé :

L'objectif du travail présenté dans ce mémoire est l'induction de règles floues interprétables à partir de données dans le but de la coopération homme machine. Dans l'état de l'art que nous avons réalisé, les méthodes d'induction de règles floues sont organisées en trois familles. Leur comparaison montre que l'interprétabilité n'est pas garantie par le formalisme flou.

La partie principale de ce mémoire décrit la méthode d'induction de règles floues que nous proposons. Elle vise à satisfaire trois conditions d'interprétabilité : lisibilité du partitionnement, nombre de règles minimal, règles incomplètes. La procédure est décomposée en trois étapes : une phase intradimensionnelle pour générer une famille de partitions par variable d'entrée, une composition multidimensionnelle pour construire un premier système performant, et une simplification de la base de règles.

Elle s'appuie sur des concepts originaux tels qu'une distance entre observations qui prenne en compte la structure de la partition, ou encore le contexte défini par un groupe de règles. Elle est guidée par des indices, indice de couverture et d'hétérogénéité, que nous avons introduits en complément de l'index de performance numérique.

Après une validation sur des exemples connus, la méthodologie est appliquée à la conception d'un système d'aide à la décision. Il s'agit d'induire les actions de conduite, sous forme de règles, qui accentuent la couleur du vin rouge au cours de la vinification.

Mots clés :

Système d'inférence floue, partitionnement, distance, interprétabilité, induction de règles, apprentissage, procédé agro-alimentaire, vinification.

Title : Induction of interpretable fuzzy rules

Abstract :

This report deals with interpretable fuzzy rule induction from data for human-computer cooperation purposes. A review of fuzzy rule induction methods shows that they can be grouped into three families. Their comparison highlights the fact that the interpretability is not guaranteed.

The central part of our work is a new fuzzy rule induction method. It aims to fulfill three interpretability conditions : readable fuzzy partitions, a number of rules as small as possible, incomplete rules. This is achieved through a three step procedure : generating a family of fuzzy partitions for each input variable, building an accurate fuzzy inference system, simplifying the rule base.

The procedure is based on original concepts such as a metric distance suitable for fuzzy partitioning, and the input context defined by a set of rules. We introduced coverage and heterogeneity related indices to guide the prodedure, complementary with a numerical performance index.

The method is first validated using well known data and then applied to decison making in a complex system. This application means to extract winemaking rules which enhance the color of red wine.

Key words :

Fuzzy Inference System, Partitioning, Distance, Interpretability, Rule Induction, Machine Learning, Food Processing, Winemaking.