Représentation des connaissances et systèmes d'inférence floue

S. Guillaume

Université Paul Sabatier, Toulouse III
École doctorale Systèmes

Mémoire d'Habilitation à Diriger des Recherches

Section 61

Génie informatique, Automatique, Traitement du signal

présenté par

Serge GUILLAUME

Représentation des connaissances et systèmes d'inférence floue

Soutenu le 21 novembre 2005 devant le jury composé de :

Isabelle Bloch	Professeur ENST	Paris	Rapporteur
Laurent Foulloy	Professeur ESIA	Annecy	Rapporteur
Pierre-Yves Glorennec	Professeur INSA	Rennes	Rapporteur
Luis Magdalena	Professeur ETSIT	Madrid	
Jean-Louis Calvet	Professeur UPS	Toulouse	
Brigitte Charnomordic	Ingénieur INRA	Montpellier	
Didier Dubois	DR IRIT-CNRS	Toulouse	
André Titli	Professeur INSA	Toulouse	Directeur de recherches

Merci

L'homme est animal curieux, non seulement il se construit tout au long de sa vie, mais de plus il ne peut se construire que dans sa relation avec d'autres. Aussi, à une étape aussi significative de ma vie, je ne peux que penser à toutes celles et à tous ceux avec qui j'ai fait un bout de chemin sur la route de l'existence. C'est à vous tous, famille, amis, compagnons de route de la vie militante, collègues de travail, que je dois ce que je suis devenu aujourd'hui.

Ce mémoire consacre une évolution qui a été possible grâce au Cemagref, qui m'a offert un environnement propice, et à ses responsables qui m'ont fait confiance tout au long de mes 20 ans de carrière. Ils m'ont permis d'accéder à des responsabilités sans cesse renouvelées, tout en encourageant des efforts de formation.

Ce document capitalise le travail d'une équipe dont les deux piliers sont Brigitte Charnomordic et Luis Magdalena. Je mesure la chance d'avoir bénéficié de deux années de mise à disposition et d'avoir pu, ainsi, établir des collaborations durables avec mes hôtes. La permanence dans l'équipe est également assurée par Jean-Luc Lablée, sans qui FisPro ne serait pas aussi convivial. A tous, ainsi qu'à l'ensemble de ceux qui sont cités dans ce mémoire, stagiaires, thésards ou post-doctorants, français ou espagnols, je veux redire le plaisir que j'ai eu à travailler avec vous, et vous adresse un sincère merci.

Cette habilitation doit également beaucoup à ceux qui m'ont soutenu dans cette démarche, et notamment à André Titli qui, après avoir été mon directeur de thèse, a accepté de parrainer ce travail.

Caminante, son tus huellas
el camino, y nada más ;
caminante, no hay camino,
se hace camino al andar.
Al andar se hace camino,
y al volver la vista atrás
se ve la senda que nunca
se ha de volver a pisar.
Caminante, no hay camino,
sino estelas en la mar.

ANTONIO MACHADO
(chanté par Joan Manuel Serrat)

Le vrai point de vue sur les choses est celui de l'opprimé.

JEAN-PAUL SARTRE

Table des matières

1 Introduction

Ce mémoire traite de la représentation des connaissances, dans un but opérationnel, sous la forme de systèmes d'inférence floue.

La représentation des connaissances couvre un domaine très vaste de l'intelligence artificielle. Notre contribution est limitée à la mise en forme du raisonnement humain avec un objectif d'action. Plusieurs raisons motivent le choix de la logique floue et des systèmes d'inférence floue.

La logique floue a été proposée par Zadeh [66] pour modéliser le langage naturel et pour rendre compte du caractère vague des connaissances que nous, les humains, manipulons au quotidien. Nombre de propositions, comme "Le raisin est à peu près mûr", ne peuvent pas être évaluées, comme vraies ou fausses, par la logique classique. La logique floue[1] autorise l'utilisation de concepts linguistiques comme *chaud*, *un peu* ou *très bon*, lesquels sont modélisés par des ensembles flous. Son utilisation dans les systèmes experts, sous la forme de règles de raisonnement floues, 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 [46]. 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.

Une deuxième génération de systèmes d'inférence floue a été construite par apprentissage automatique. Sugeno [58] a été l'un des premiers à proposer de tels systèmes conçus à partir des données. Leurs propriétés d'approximateurs universels démontrées [61], ces systèmes ont été utilisés en compétition avec d'autres techniques d'approximation de fonction, comme les réseaux de neurones. Le seul objectif était alors d'améliorer un critère de performance numérique, l'erreur quadratique moyenne, sans tenir aucun compte de la spécificité de la logique floue. Un retour aux sources est en train de s'opérer. Glorennec [25] a proposé des algorithmes d'apprentissage, qui grâce à des contraintes, garantissent l'interprétabilité des règles. Nous-même [28] avons utilisé le critère de l'interprétabilité comme guide d'analyse et de comparaison des différents méthodes d'induction de règles floues. Ce souci du compromis entre l'interprétabilité et la performance est maintenant partagé par plusieurs équipes qui essaient soit d'induire des règles interprétables [8], soit d'améliorer la précision des systèmes de règles linguistiques [7].

Un des enjeux de la période actuelle est la coopération, au sein d'un même cadre, de ces deux approches historiques. Nous essaierons de montrer que cette combinaison est bénéfique à la condition que les formes de la coopération soient convenablement définies, Cette intégration est susceptible d'ouvrir de nouvelles perspectives dans l'utilisation des systèmes d'inférence floue.

Dès l'origine, les règles linguistiques dans les systèmes d'inférence floue ont été reconnues pour leurs avantages : conception simple et rapide d'un système, facilité de maintenance et d'adaptation. Leur utilisation s'inscrit dans la perspective de proposer des modèles utilisant un langage de description aussi proche que possible de celui employé par les experts humains. L'utilisation de modèles transparents permet à l'expert du domaine de comprendre le raisonnement mis en œuvre sans investir dans la compréhension de la technique qui a été mobilisée pour le construire. Il s'agit là d'un atout capital ! Interrogeons-nous sur les objectifs de la modélisation. Bien sûr, il s'agit en premier lieu de proposer un modèle suffisamment précis, capable de reproduire les relations entre les entrées et les

[1] Le lecteur peu familier avec les concepts de la logique floue et des systèmes d'inférence floue est invité à consulter le glossaire en annexe ou bien, pour une présentation plus complète, les ouvrages [18, 4, 23, 59].

sorties du système étudié. Mais, nous pouvons être plus ambitieux : nous aimerions aussi que ce modèle rende compte de nos connaissances. En effet, lorsqu'on s'intéresse à un problème, il est très rare qu'on ne sache rien. C'est pourquoi, même dans les cas où des techniques de modélisation plus classiques peuvent s'avérer performantes, on attend des systèmes d'inférence floue qu'ils apportent des éléments de connaissance qualitative sur le phénomène étudié.

En effet de tels systèmes peuvent être utilisés avec différents objectifs. Comme simulateurs du procédé, ils permettent de mieux comprendre les effets d'un groupe de variables, ils peuvent permettre aussi d'explorer d'autres voies technologiques. Les systèmes d'inférence floue, en coopération avec les techniques d'apprentissage, permettent aussi la formalisation du savoir-faire en vue de sa pérennisation ou même de sa transmission. En supervision, ils peuvent reproduire le raisonnement humain pour gérer, de façon graduelle, l'articulation entre des modèles locaux, issus par exemple de l'automatique. De même, ils sont parfaitement adaptés à la construction d'attributs, par agrégation de variables, au sein d'un système de décision. Ils permettent de bâtir des modèles linguistiques, par opposition au modèles numériques, pour lesquels la sortie est un terme linguistique, comme un niveau de qualité ou de risque qui peut être par exemple *nul*, *faible*, *moyen* ou bien *élevé*.

De par leur nature, les systèmes d'inférence floue sont donc utilisables dans des environnements imprécis et incertains. Quiconque s'intéresse à la modélisation est confronté à cette imprécision. La variabilité des produits biologiques ou bien le fait d'évoluer dans un environnement ouvert pour un engin mobile rendent nécessaires des prises de décision dans l'incertain. D'autre part, ces systèmes offrent de nombreuses possibilités pour tous ceux qui s'interrogent sur les moyens d'utiliser au mieux des informations hétérogènes, de nature numérique, qualitative, sensorielle ou experte. Enfin, leur capacité de modélisation du langage naturel, les rend particulièrement efficaces pour la représentation du raisonnement approché. Ils prennent donc toute leur place dans les situations où l'humain joue un rôle clé.

Nous proposons, dans la section suivante, 2, un cadre général d'intégration des différents types et sources de connaissance utilisables dans un objectif opérationnel. Il s'agit de favoriser leur coopération, notamment celle de la connaissance experte et des données expérimentales, à toutes les étapes de la conception d'un système. Le développement de cette proposition s'est appuyé sur des applications variées et se traduit par deux implémentations logicielles.

Les applications ont joué un rôle très important dans l'élaboration de ce travail. Loin d'être considérées comme une simple utilisation d'une méthode pré-existante, les applications, par les questions qu'elles soulèvent, ont stimulé ces développements.

Historiquement, la première classe d'applications traitées, la simulation des procédés alimentaires, répondait à une préoccupation du Cemagref et de l'Inra. 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 souvent mal modélisées. Les difficultés de modélisation et de quantification des interactions sont un obstacle à leur compréhension et à leur automatisation. Ainsi, l'expert est au cœur du système. Ses décisions sont prises à partir d'une multitude d'informations qui ne sont pas toutes à notre disposition. Deux procédés ont été étudiés : la fabrication du fromage Comté [29] et l'amélioration de la couleur au moment de la vinification du vin rouge [30, 35]. Dans les deux cas les informations prises en compte sont des mesures, mais aussi des évaluations qualitatives ou encore des actions de l'opérateur.

Les données sensorielles, du fait de l'absence de référence absolue, sont par nature imprécises. L'exemple que nous avons traité [33, 32], vise à mettre en relation les appréciations de consommateurs sur un lot de biscuits avec les caractéristiques de ces produits évaluées par un panel d'experts

entraînés. Les méthodes d'induction, moins gourmandes en données que les statistiques, se sont révélées efficaces dans ce cadre particulier où l'acquisition des données est coûteuse.

Mon séjour en Espagne m'a permis de travailler sur des applications très différentes, de robotique sous marine et terrestre. Nous nous sommes plus particulièrement investis dans la conception d'un module de diagnostic pour la navigation de ces robots. L'objectif est de diagnostiquer des dysfonctionnements causés par des obstacles non détectés par les capteurs et de proposer des actions de récupération. Ce travail, d'abord réalisé pour un robot terrestre, a mobilisé des compétences diverses, électroniciens, mécaniciens et informaticiens notamment [45]. Il constitue une belle illustration de la complémentarité entre l'expertise et les données expérimentales. En effet, le point de départ est une base de règles expertes décrivant le comportement du système formé par le robot et l'obstacle. Celle-ci, incomplète, a ensuite été enrichie par des règles extraites de données acquises dans des situations représentatives de celles que l'on veut détecter. Enfin l'expert, guidé par des algorithmes, a finalement simplifié la base de règles résultante.

Les logiciels, en particulier les logiciels libres, représentent une valorisation importante de ce travail de recherche.

Le logiciel *FisPro* [36] constitue une boîte à outils qui permet à un expert d'un domaine d'application de concevoir des systèmes d'inférence floue soit à partir de sa propre connaissance, soit à partir d'un ensemble de données d'apprentissage. Ce logiciel implémente nos propres développements ainsi que les méthodes connues que nous avons jugées les plus compatibles avec nos critères d'interprétabilité. Il est portable, testé sur les principales plate-formes existantes, localisé, son interface est disponible en trois langues, et documenté. Ouvert, il peut être personnalisé et est susceptible d'accueillir des contributions.

KBCT [49], dont les initiales signifient *Knowledge Base Configuration Tool*, est développé par l'équipe espagnole. Basé sur *FisPro*, il implémente le dialogue entre la connaissance experte et les données expérimentales. Comme *FisPro*, il est portable, localisé, ouvert et documenté.

Ces deux logiciels constituent un vecteur de la diffusion des travaux que nous allons maintenant présenter en détail.

2 Différents types de connaissance

Plusieurs typologies de la connaissance sont possibles. La connaissance superficielle, connaissance empirique possédée par un expert dans un domaine spécifique, est basée sur les faits. A l'opposé, la connaissance profonde correspond à une connaissance théorique et abstraite. L'archétype de la connaissance profonde est le modèle mathématique.

Les éléments de connaissance peuvent être de type déclaratif, décrivant des concepts, leurs propriétés et les relations entre eux. Ce type de connaissance statique est également appelé connaissance du domaine. Le pôle correspondant est la connaissance de type procédural, permettant l'inférence de nouvelles propriétés ou relations. Celui-ci inclut la connaissance opératoire, la connaissance contenue dans les règles, qui indique la manière d'utiliser les faits et représente le savoir-faire sur le domaine.

Notre objectif étant opérationnel, nous nous concentrons sur la connaissance de type opératoire. Nous allons la représenter sous forme de règles floues au sein de systèmes d'inférence floue.

Une autre typologie va particulièrement nous intéresser c'est celle qui oppose la connaissance positive, les faits avérés, à la connaissance négative, les contraintes à respecter. En effet, à ces différents types de connaissance correspondent différents types de règles floues.

Pour la conception de systèmes, nous disposons principalement de deux sources de connaissance : la connaissance experte et celle contenue dans des données expérimentales. Les modèles mathématiques représentent un cas particulier de la connaissance experte négative.

L'objectif de cette section est de présenter les caractéristiques de chacune d'elles pour souligner leur caractère complémentaire, puis, après avoir envisagé différents modes de coopération possibles, de proposer un cadre de travail.

La première source de connaissance est la connaissance experte. Le dictionnaire, comme le note Larichev [44], donne de l'expert la définition suivante : "une personne ayant une grande connaissance, de la compétence et de l'expérience dans un domaine". Cornelissen *et al.* [11] soulignent qu'un expert est une personne dont la connaissance dans un domaine particulier est acquise progressivement, au cours d'une période d'apprentissage et d'expérimentation. Pour Roy [55], la pratique n'est rien d'autre que le mécanisme d'apprentissage du cerveau à partir des exemples qui lui sont présentés.

L'acquisition de la connaissance experte est une tâche difficile.

Dans l'approche multi-modèles proposée dans [57], cette acquisition est un procédé itératif. Dans ce cadre le système expert n'est pas réduit à un moteur d'inférence : il est vu comme un modèle qui exhibe un comportement. Ce comportement, qui coïncide ou non avec celui de l'expert, résulte de la coopération de plusieurs modèles, l'un d'entre eux étant le modèle de l'expertise. Le système expert aide l'expert à formaliser de manière cohérente sa connaissance en réalisant des tâches qui sont hors de portée d'un esprit humain : conserver un grand nombre d'hypothèses dans la mémoire immédiate, expliciter le raisonnement mis en œuvre, ...

Mais le mode d'acquisition le plus usuel, parce que le moins onéreux, consiste à demander à l'expert d'écrire ses propres règles de raisonnement. Cette forme de transfert est bien évidemment limitée. Des travaux montrent qu'une part essentielle de la connaissance experte dans différents champs professionnels se situe à un niveau inconscient et ne peut pas être recueillie en questionnant l'expert [44].

Cependant, généralement, l'expert connait les grandes tendances de son système et est capable de décrire, de manière qualitative, le comportement des variables les plus influentes. La connaissance experte inclut des éléments de connaissance positive, qui correspondent à des situations rencontrées et qui sont mobilisés au sein des des raisonnements par analogie, ou encore des raisonnements à base de cas. S'agissant d'un procédé de vinification, un des éléments bien connus de l'œnologue s'exprime ainsi : "Les effets de la dose d'enzyme, de la température et de la durée de fermentation se compensent". Comme les règles sont du type "Si l'entrée est de type A alors il se peut que la sortie soit de type B", cette connaissance s'implémentera au sein d'un système visant à mettre en relation la couleur du vin rouge avec les actions de l'œnologue sous la forme d'un ensemble de règles comme : *"Si la vitesse de fermentation est grande et la température en fin de fermentation est faible et la dose d'enzyme est moyenne alors l'intensité colorante est moyenne"* et "*Si la vitesse de fermentation est grande et la température en fin de fermentation est élevée et la dose d'enzyme est faible alors l'intensité colorante est moyenne"*. Ces deux règles ne diffèrent que par les labels des variables *température* et *dose d'enzyme* : dans la première ils sont respectivement *faible* et *moyenne* tandis que dans la seconde ils sont *élevée* et *faible*. Ces différences illustrent la compensation décrite lorsque la fermentation est rapide puisque les deux règles concluent à une intensité colorante *moyenne*.

La connaissance experte comprend également des éléments de connaissance négative. A la différence des éléments de connaissance positive, dérivés d'observations, ceux-ci représentent des certitudes et se traduisent par des contraintes à imposer. Ces éléments permettent donc d'écarter des situations qui ne les respectent pas. Les restrictions peuvent porter sur le domaine d'une variable, *Dans telle situation la température ne doit pas excéder 14 à 15 ^oC* ou sur une action. Les contraintes peuvent provenir également de la confrontation de points de vue sur un même problème. L'ingénieur chargé de la conception d'un véhicule cherche à en diminuer la masse, *Plus les pièces sont légéres meilleures elles sont*, mais parfois la masse est justifiée par d'autres usages. Ainsi le mécanicien

pourra dire : "*Cette pièce sert d'appui au levier pour en démonter une autre, il ne faut surtout pas la fragiliser*".

Il est clair que ces deux types de connaissance, la connaissance positive et la connaissance négative, ne peuvent pas être modélisés par le même type de règle. Weisbrod [63] et surtout l'équipe de Dubois et Prade [14, 15, 17, 19] ont proposé des formes de raisonnement adaptées à la connaissance négative.

L'autre source de connaissance dont nous disposons est constituée par les données expérimentales qui peuvent être enregistrées pendant le fonctionnement du système que nous voulons modéliser. Les données sont, par définition, uniquement des éléments de connaissance positive. Elles peuvent servir de base à l'élaboration d'éléments de connaissance négative, mais entre les observations et la loi il y a nécessairement un travail de modélisation. La principale caractéristique des données est l'incomplétude : un ensemble de données ne peut pas prétendre couvrir l'ensemble des situations, et ceci est particulièrement vrai pour les systèmes complexes et les systèmes de grande dimension.

L'extraction de la connaissance à partir d'un jeu de données implique une phase d'induction. L'induction est le type de raisonnement qui consiste à tirer une connaissance générale à partir de faits particuliers : $(p(a)\ ;\ q(a)) \Rightarrow \forall x \quad p(x) \rightarrow q(x)$. Dans quelle mesure ayant observé les propriétés p et q sur certains individus, puis-je en conclure que pour l'ensemble de la classe si l'on observe la propriété p on observera la propriété q ? Contrairement au raisonnement déductif, $(p \wedge (p \rightarrow q)) \Rightarrow q$, le raisonnement inductif suppose un risque, mais il s'agit du risque de la création. Donc, l'induction de règles, et particulièrement de règles floues, consiste à générer des règles générales à partir d'un ensemble d'exemples limité. Les règles expertes sont des règles universelles, car elles sont basées sur une expérience cumulée importante, qui couvre un grand nombre de situations. Par opposition, les règles induites ne sont pas universelles et leur caractère général dépend de la qualité de l'ensemble d'apprentissage. Plus les exemples sont représentatifs du système et de son comportement, plus les règles sont générales. Dans la pratique les exemples correspondant à des défauts ou à des conditions de fonctionnement extrêmes sont rares.

La modélisation de systèmes pour lesquels l'humain fait partie du processus de décision, avec l'objectif de conception d'un système d'aide à la décision par exemple, nous intéresse particulièrement. Nombre de procédés non automatisés tels les procédés agroalimentaires ainsi que les applications de robotique télé-opérée entrent dans cette catégorie. Dans ce cas, les données contiennent, de façon implicite, l'état du système mais aussi les conséquences des décisions prises par l'opérateur. Les règles induites contribuent ainsi à la formalisation du savoir-faire de l'opérateur.

Les données présentent certains avantages : elles rendent compte des phénomènes et représentent une bonne image des interactions entre les variables au sein du système.

Les caractéristiques des deux sources de connaissance montrent que leur coopération ne peut être que bénéfique pour la conception de systèmes. En effet, le savoir-faire de l'expert doit être valorisé mais sa formalisation, pour des systèmes complexes, se heurte à deux types de difficultés : l'identification des variables susceptibles d'influer sur le procédé et la quantification des interactions. Ainsi, les systèmes contenant seulement des règles expertes sont généralement peu performants dans le sens où ils ne permettent pas de reproduire avec suffisamment de précision les phénomènes observés. Les données, bien qu'intrinséquement limitées, sont susceptibles d'apporter la matière permettant de les compléter.

L'expertise réside principalement au niveau qualitatif, les données, quant à elles, sont susceptibles d'apporter un éclairage quantitatif. Le niveau de coopération, entre l'expertise et les données, dépendra du caractère interprétable des règles induites. Celui-ci n'est nullement garanti par le formalisme flou, et les méthodes d'induction traitent cet aspect de façon très inégale [28].

En fonction du coût respectif de l'acquisition des données et du recueil de l'expertise plusieurs types de coopération sont envisageables. Ces coûts respectifs dépendent du problème, parfois ce sont

les données qui sont onéreuses dans certaines zones de fonctionnement comme les conditions extrêmes, les points de rupture ou encore l'acquisition de données correspondant à des défauts ; mais nous avons vu que le recueil du savoir-faire, lorsqu'il existe, est également coûteux.

Trois types de coopération peuvent être envisagés.

Dans le premier cas, le système est conçu par l'expert et les données sont utilisées pour la validation des règles ou pour l'optimisation de certaines parties du système : bornes des fonctions d'appartenance ou conclusions des règles par exemple. Cette option est limitée à des systèmes simples car il est contre productif de demander à un expert de définir un grand nombre de paramètres : ce n'est pas son terrain de compétence.

La seconde option consiste à concevoir le système par apprentissage automatique à partir d'un ensemble de données. Les principaux algorithmes d'apprentissage pour les systèmes d'inférence floue sont présentés dans [25]. Le rôle de l'expert consiste dans ce cas à valider, a posteriori, la connaissance induite. Symétrique de la première, cette forme de coopération est moins coûteuse pour l'expert du domaine. Lire est toujours moins difficile qu'écrire. Soulignons toutefois que cette démarche n'est possible qu'à la condition que la base de règles induites soit interprétable.

La troisième solution envisageable est d'impliquer les deux ressources lors de la conception du système et non seulement pour une validation. Ce processus, qui doit être sous le contrôle total de l'expert, pourrait s'appeler induction guidée par l'expert. Il est présenté dans la section 4.

La figure 1 résume les formes de coopération envisagées. Plusieurs itinéraires sont proposés pour aller de la connaissance à une base de règles. La première bifurcation est celle du type de connaissances. Les données sont le point de départ des procédures d'induction, les contraintes d'interprétabilité ainsi que notre contribution dans ce domaine seront présentées dans la section 3.

L'objectif pour la connaissance experte est sa formalisation pour qu'elle soit utilisable par le système informatique. Deux branches sont à distinguer. La partie correspondant à la connaissance positive est susceptible de coopérer avec les données, c'est le sens de la flèche horizontale qui relie les règles conjonctives à la branche induction. La section 4 présente une voie possible de coopération ainsi que la problématique de fusion de deux bases de règles de même nature, les règles conjonctives à possibilité.

La branche correspondant à la connaissance négative est l'objet de notre travail de recherche actuel. Les différentes sémantiques de règles, ainsi que les perspectives à court terme sont présentées dans la section 5.

L'objectif étant l'intégration, au sein d'une même base de règles de règles de différentes nature, la question de la maintenance de celle-ci sera également évoquée dans les perspectives.

Nous nous attacherons à souligner, au cours de ce mémoire, notre contribution dans chacune des étapes de ces itinéraires, mais aussi les points qui demeurent perfectibles et ceux qui, à l'état de perspective, sont ouverts.

3 Induction

Les procédures d'induction sont habituellement pilotées par un critère d'erreur tel que l'erreur quadratique moyenne. L'objectif d'extraction de connaissances en vue de leur coopération avec l'expertise nous impose une propriété supplémentaire : le respect de la sémantique.

Trois conditions nous semblent nécessaires pour rendre les règles induites interprétables. En premier lieu le partitionnement des variables doit être lisible pour les experts, les ensembles flous doivent correspondre à des termes linguistiques, qui ont un sens pour le procédé. Pour que les règles s'expriment complètement sous forme linguistique, leurs conclusions sont des ensembles flous, système de Mamdani, ou bien des constantes, système de Sugeno d'ordre 0. Ainsi les règles peuvent être comparées. Ensuite, le nombre de règles doit rester raisonnablement petit. Cette réduction s'accompagne

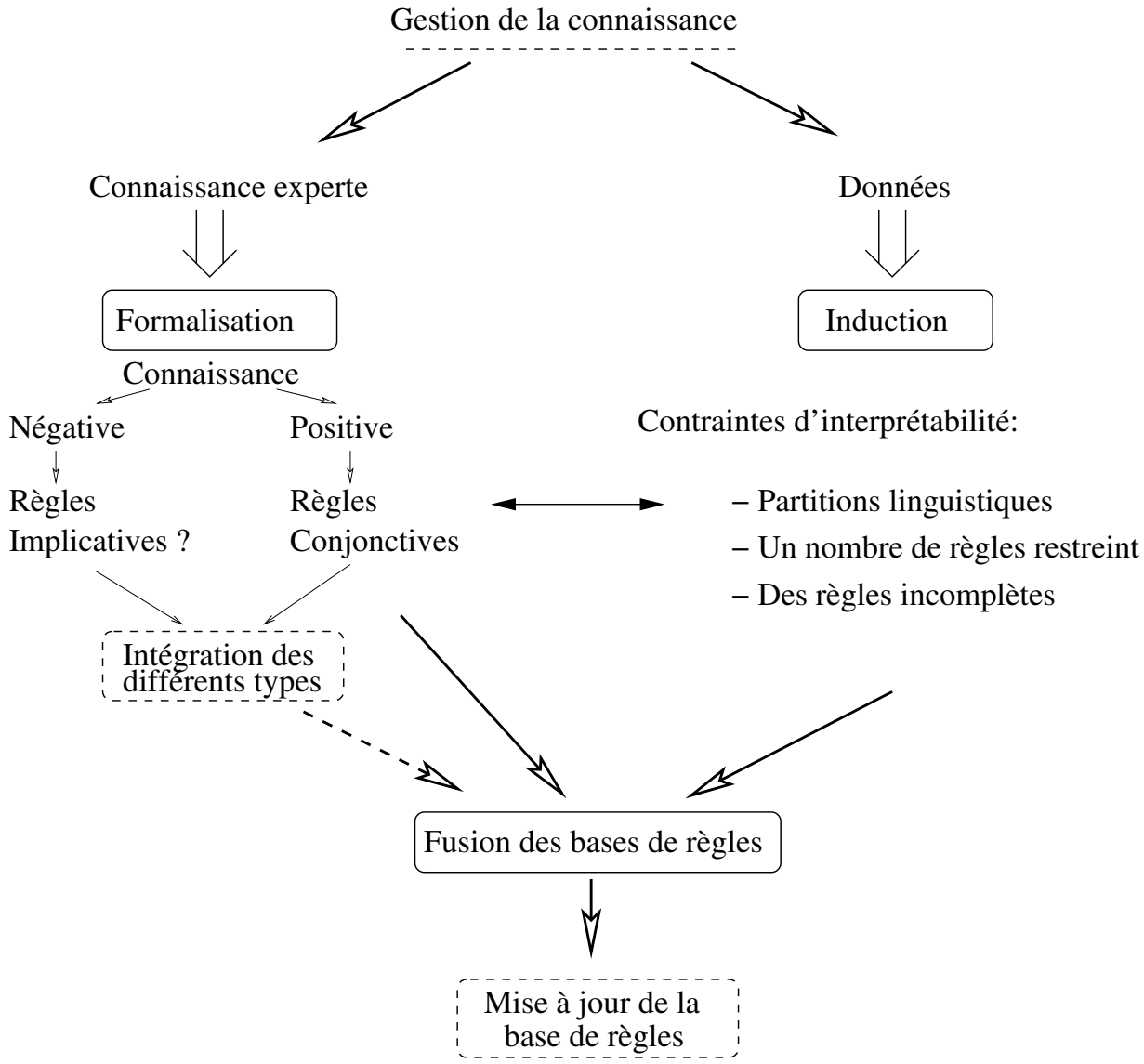


FIG. 1 – Un cadre de représentation des connaissances

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.

Dans l'approche que nous proposons la conception des partitions est indépendante de la génération des règles. Cela permet d'appliquer l'ensemble des méthodes de génération des règles sur n'importe quelle partition. La section 3.1 présentera notre travail sur le partitionnement des variables. Comme nous travaillons avec une famille de partitions de taille différente, la section 3.2 présente des outils pour sélectionner la granularité. La section 3.3 traite des méthodes d'induction de règles. Quelle que soit la méthode d'induction choisie, nous proposons une procédure de simplification de la base de règles dans la section 3.4. Enfin la section 3.5 ouvre quelques perspectives.

3.1 Conception des partitions

La méthode de partitionnement que nous proposons [34] est conduite sans aucune référence à la valeur de sortie du système : seule est prise en compte la distribution des données de la variable. L'intérêt est de la regarder pour elle-même sans aucun a priori sur son utilisation future. Par rapport aux méthodes connues [47], elle présente l'originalité de fournir une famille de partitions emboitées, au lieu d'une partition unique. Cette propriété permet de différer le choix de la granularité.

Les fonctions d'appartenance gaussiennes sont très utilisées par les automaticiens du fait de leurs propriétés, notamment leur dérivabilité. Les experts utilisent plus spontanément des formes trapézoïdales ou triangulaires. Pedrycz a montré l'intérêt de ces dernières pour les systèmes d'inférence floue [51].

Les qualités d'une partition interprétable ont été étudiées par plusieurs auteurs [56, 25, 12, 20]. Nous les rappelons : les fonctions d'appartenance sont en nombre justifiable, elles sont normales et distingables, elles présentent un recouvrement significatif avec les voisines et la partition est complète (chaque élément appartient de manière significative, $\mu(x) > \epsilon$, à au moins l'un des termes).

Avec les partitions floues fortes, comme celle illustrée dans la figure 2, le niveau de couverture est $\epsilon = 0.5$

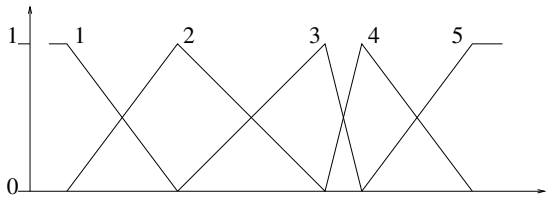


FIG. 2 – Une partition floue forte

Le point de départ de notre algorithme est une partition floue forte qui comprend autant de sous-ensembles flous que de données distinctes dans la distribution. A chaque étape deux sous-ensembles flous sont fusionnés, ils sont choisis en fonction d'un critère basé sur une notion de distance adaptée au partitionnement flou [37]. La figure 3 montre le résultat des derniers pas de la procédure ainsi que la distribution des données.

Le concept de hiérarchie peut être étendu à d'autres méthodes de partitionnement. Le logiciel FisPro propose deux types de familles supplémentaires : celle constituée par des grilles régulières et celle résultant d'un algorithme de groupage monodimensionnel, les $k-means$, exécuté plusieurs fois avec un nombre de centres croissant.

Ces algorithmes de partitionnement à partir d'une distribution s'appliquent aussi bien aux variables d'entrée que de sortie. Néanmoins, à des rôles différents doivent correspondre des propriétés et des contraintes spécifiques. Considérons les extrémités des partitions formées par les demi-trapèzes inférieur et supérieur.

S'agissant des entrées, la partition est utilisée pour mesurer le degré d'appartenance d'une valeur de l'univers à un terme linguistique. Ainsi, le domaine de variation qui correspond au noyau des demi-trapèzes est considéré comme typique du label linguistique associé, avec un niveau d'appartenance maximum pour cet ensemble et nul pour les autres.

La partition de la variable de sortie n'est pas utilisée de cette manière : elle permet d'inférer une valeur de sortie, en réalisant ou non une interpolation entre les différentes étiquettes. L'objectif dans ce cas est de pouvoir inférer toutes les valeurs du domaine de variation, y compris les valeurs extrêmes. Pour cela il est nécessaire de prendre en compte le type d'opérateur de défuzzification. Deux types principaux sont à considérer.

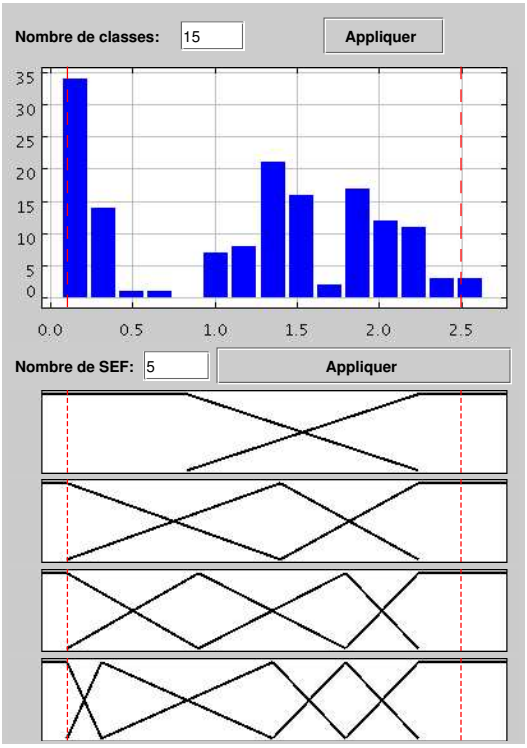


FIG. 3 – Une hiérarchie de partitions floues fortes

La figure 4 illustre le cas de défuzzication par la moyenne des maxima sur le demi-trapèze inférieur, défini par les points a, b, c, et celui du centre de gravité sur le demi-trapèze supérieur, défini par les points d, e, f.

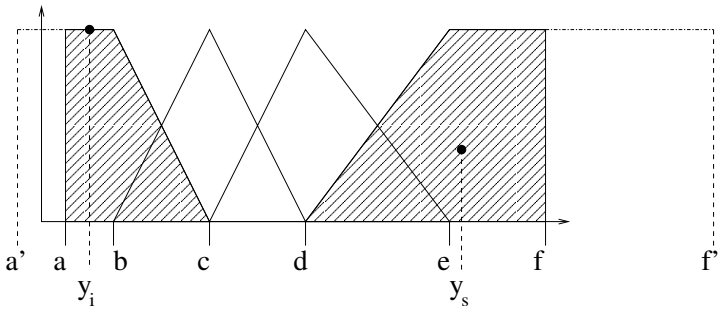


FIG. 4 – Inférer les valeurs extrêmes de la sortie

Les procédures de génération des partitions limitent les demi-trapèzes aux bornes du domaine de variation de la variable. Ce domaine est donc représenté sur la figure par le segment $[a, f]$. Dans ce cas, le domaine de variation des valeurs inférées ne couvre pas l'ensemble du domaine de variation de la variable de sortie. La borne inférieure correspondant à l'opérateur de moyenne des maxima, y_i, est le milieu du segment $[a, b]$. La borne supérieure accessible par l'opérateur du centre de gravité est y_s, la projection du centre de gravité du demi-trapèze supérieur sur le domaine de variation.

Pour pouvoir inférer les bornes du domaine, les limites des demi-trapèzes sont reportées au-delà du domaine de variation, sans aucune altération de la sémantique. La position des limites corrigées, a' et f' sur la figure, se calcule très simplement dans les deux cas présentés.

3.2 Granularité des partitions

Le fait de disposer d'une famille de partitions floues fortes pour chacune des variables, nous permet d'envisager plusieurs options pour le choix du nombre de sous-ensembles flous par variable.

Ce choix peut résulter de l'optimisation, minimisation ou maximisation, d'un critère. Les premiers critères proposés l'ont été dans le cadre d'approches de "clustering", ou coalescence. L'objectif étant d'organiser les données en un nombre de groupes déterminé. Les critères visent à qualifier la partition : une bonne partition sera formée de groupes bien séparés les uns des autres et dont les éléments sont rassemblés autour de leur centre de gravité. Les critères les plus communément utilisés sont ceux de Bezdek [2], appelés coefficient de partition (PC) et entropie de la partition (PE), ainsi que ceux de Xie-Beni [64] (XB) et Fukuyama-Sugeno [22] (FS).

Dans les formules qui suivent, c est le nombre de groupes, n le cardinal de l'ensemble d'apprentissage, u_{ik} le degré d'appartenance de l'exemple k au groupe i, x_k les coordonnées de l'exemple k, v_i les coordonnées du centre du groupe i.

$$PC = \frac{\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^2}{n}$$

$$PE = -\frac{1}{n}\left\{\sum_{k=1}^{n}\sum_{i=1}^{c}\left[u_{ik}\log(u_{ik})\right]\right\}$$

$$XB = \frac{\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^2 \|x_k - v_i\|^2}{n(\min_{i\neq j}\{\|v_i - v_j\|^2\}}$$

$$FS = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \left(\|x_k - v_i\|_M^2 - \|v_i - \overline{v}\|_M^2\right)$$

D'autres critères, comme ceux proposés par Pedrycz [52, 53] ou bien nous-même [34], sont inspirés de l'égalisation d'histogramme, et visent à qualifier l'harmonie des densités.

Malgré les efforts récents, une difficulté demeure pour l'utilisation de ces indices : ils sont mal adaptés à la comparaison de partitions de taille différente. Nous proposons, comme alternative à ces indices, un algorithme de raffinement, algorithme 1, qui permet de déterminer le nombre de sous-ensembles flous par variable en fonction de la performance du système. p est la dimension de l'espace d'entrée (nombre de variables), n_j^{max} (respectivement n_j) est le nombre maximum (respectivement courant) de sous-ensembles flous pour la variable j, $Perf$ est la valeur de l'indice de performance pour le système.

La procédure de raffinement consiste à introduire les seules variables nécessaires pour la définition de la base de règles, et ce avec le nombre minimum de sous-ensembles flous. Le point de départ est le système le plus simple : il n'est formé d'une seule règle. A chaque itération, un sous-ensemble flou est ajouté dans la partition d'une variable. La sélection est réalisée suivant l'amélioration de la performance. Le résultat n'est pas un seul système, mais une famille de systèmes de complexité croissante. Le choix final, qui résulte du compromis entre la complexité du système et sa performance, est laissé à l'utilisateur.

Remarquons que cet algorithme n'est pas un algorithme glouton : seule une partie de l'espace des possibles est explorée.

Algorithm 1 Procédure de raffinement

iter = 1 ; $\forall j\ n_j = 1$
CALL *FIS Generation*
while $iter \leq iter_{max}$ **do**
  Store system as base system
  for $1 \leq j \leq p$ **do**
    if $n_j = n_j^{max}$ **then** next j (partition size limit reached for input j)
    n_j = n_j + 1
    CALL *FIS Generation*
    $Perf_j = Perf$
    n_j = n_j - 1
    Restore base system
  end for
  if $\forall j\ n_j = n_j^{max}$ **then** exit (no more inputs to refine)
  s = argmin $\{Perf_j,\ j = 1, \ldots, p, \quad n_j < n_j^{max}\}$ (Select input to refine)
  n_s = n_s + 1
  CALL *FIS Generation*
  keep FIS_{iter}
  iter = iter + 1
end while

Cet algorithme permet donc la construction des systèmes les plus performants parmi les plus compacts. Le nombre de variables introduites est, en règle générale, petit, ce qui contribue à une meilleure lisibilité des règles. La procédure présentée est néanmoins perfectible. Lorsque, à une étape donnée, plusieurs solutions conduisent à une amélioration de la performance voisine, nous ne disposons d'aucun critère pour effectuer un choix qui est irréversible : aucun retour en arrière n'est prévu. Une perspective d'amélioration consiste à explorer l'espace des solutions avec un niveau de profondeur paramétrable pour différer ce choix.

Cet algorithme fait appel à une procédure de génération du système, *FIS Generation*. Celle-ci consiste simplement à implémenter toutes les règles correspondant à l'ensemble des combinaisons des entrées, puis à ne retenir que celles qui sont supportées par des données d'apprentissage. La conclusion d'une règle est initialisée en prenant en compte les valeurs de sortie des exemples qui l'activent le plus.

3.3 Induction des règles

Il existe de nombreuses méthodes d'induction de règles [28]. Nous avons choisi de nous intéresser seulement à celles qui sont susceptibles de conduire à des résultats interprétables. Ce choix élimine les méthodes de groupage multi-dimensionnel, ou "clustering". En effet, dans ce cas un groupe correspond à une règle et le partitionnement des entrées est dérivé de la projection du groupe sur chacun des axes. Les sous-ensembles flous de la partition peuvent être différents dans chaque règle, ce qui interdit toute comparaison pour étudier l'influence d'une variable.

En plus de l'indice de performance, nos procédures d'induction sont régies par un indice de couverture. Celui-ci mesure la capacité du système à gérer la diversité des situations présentes dans l'ensemble d'apprentissage. Un exemple sera déclaré inactif s'il n'active pas au moins une règle avec un niveau suffisant. Cela signifie que la base de règles manque d'éléments de connaissance pour représenter correctement les exemples inactifs, et nous avons donc choisi de ne pas prendre en compte ces exemples dans le calcul de l'indice de performance. Par conséquent, une erreur faible peut dissimuler

le fait que des parties importantes de notre espace ne soient pas couvertes par les règles. Nous avons donc besoin d'une information complémentaire pour faire la différence entre deux systèmes qui ont un indice d'erreur comparable. Le nombre I d'exemples inactifs est utilisé pour définir un indice de couverture. Sa formule est la suivante, N étant le nombre total d'exemples : $Couv = 1 - \frac{I}{N}$

Le choix important que nous avons fait, séparer les phases de conception des partitions et d'induction des règles, nous a conduit à revisiter des algorithmes qui incluent les deux étapes. C'est le cas notamment de celui de Wang et Mendel [62] ou de celui de génération des règles par orthogonalisation des moindres carrés (OLS) [9, 10, 61, 42].

Ce dernier algorithme est intéressant car il réalise une sélection des règles. Les règles retenues dans le système sont celles qui expliquent une part significative de l'inertie de la sortie. Leur ordre, par pouvoir explicatif décroissant, facilite l'analyse de la base de règles. La sélection est effectuée par régression linéaire. Pour utiliser des techniques linéaires dans un cadre d'optimisation non linéaire, il est nécessaire de reformuler le problème. Un système d'inférence floue, formé de règles dont la conclusion est un scalaire (système de Sugeno d'ordre 0), 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, les fonctions de base floues [61].

La première étape de cet algorithme consiste en la définition du partitionnement des variables : chaque valeur distincte de la distribution est considérée comme la moyenne d'un sous-ensemble flou gaussien d'écart type fixé. Ainsi, la situation est paradoxale : la régression demande un volume de données important, or plus le volume de données est important moins le partitionnement sera lisible.

Nous avons également montré [13] que ce grand nombre de fonctions d'appartenance engendre des problèmes de couverture dès lors que le nombre de variables augmente (ordre de grandeur 4 à 5). Ce phénomène est caché dans l'algorithme d'origine car le degré d'appartenance à un sous-ensemble flou gaussien est toujours strictement positif. Pour le mettre en évidence, il suffit de remplacer les fonctions d'appartenance gaussiennes par des triangles de forme semblable. Dans ce cas, nombre d'exemples n'activent plus aucune règle, et ne sont donc pas gérés par le système.

La première modification que nous proposons est donc l'utilisation de partitions floues fortes définies par un nombre raisonnable [48] de sous-ensembles flous.

Les conclusions des règles sont optimisées par les moindres carrés. Cela conduit à un nombre distinct de conclusions voisin du nombre de règles. Il est possible, pour gagner en interprétabilité, de réduire ce nombre de valeurs comme cela a été proposé dans un autre cadre par Glorennec [25]. Les nouvelles valeurs résultent d'un groupage mono-dimensionnel, les conclusions des règles sont simplement remplacées par la nouvelle valeur la plus proche. Dans nombre d'applications, cette réduction peut être réalisée sans perte significative de performance numérique.

Nous utilisons également une méthode très simple d'induction de règles, proposée par Glorennec [24, 25], appelée algorithme de prototypage rapide. Elle consiste à générer l'ensemble des règles correspondant aux combinaisons des entrées, puis à initialiser les conclusions des seules règles supportées par les données. La conclusion d'une règle est calculée à partir des sorties des exemples qui vérifient le mieux à la règle. Cette méthode, très efficace pour un prototypage, ne remplace en aucun cas une procédure d'optimisation.

Nous proposons d'étendre cet algorithme à différents types de sortie, comme une sortie floue, et de choisir les exemples à retenir en fonction de plusieurs paramètres. Soit E_r l'ensemble des exemples correspondant à la règle r.

Considérons d'abord le cas d'une sortie nette. Si l'on travaille en classification, la conclusion de la règle est simplement la classe majoritaire dans E_r. Dans le cas d'une 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* :

$$C_r = \frac{\sum_{i \in E_r} \mu_r(x_i) * y_i}{\sum_{i \in E_r} \mu_r(x_i)} \quad (1)$$

L'initialisation d'une conclusion floue se fait en deux temps. Tout d'abord une sortie nette est calculée suivant l'équation 1, comme pour une sortie continue. Puis, la conclusion de la règle est choisie comme le numéro de l'ensemble flou pour lequel le degré d'appartenance de la sortie nette est maximum.

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 *décrémentale*, 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 un maximum (*0.7*, par exemple). 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* 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 exemples proches des prototypes de la règle, dont la définition est rappelée dans le glossaire en annexe.

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*.

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 arbres de décision flous [5, 54, 43], parce qu'ils permettent de générer des règles incomplètes, restent une méthode de référence que nous avons implémentée. Notre version actuelle n'inclut aucune procédure d'optimisation. Une version future pourrait exploiter les familles de partitions pour déterminer, au cours de la construction de l'arbre, le meilleur niveau de granularité pour chacune des variables.

3.4 Simplification de la base de règles

Les méthodes d'induction présentées dans la section précédente incluent généralement une étape de sélection. L'OLS sélectionne les règles les plus représentatives de l'ensemble d'apprentissage, dans les arbres de décision, seules les variables nécessaires sont introduites dans chacune des branches, enfin les paramètres de l'algorithme de prototypage rapide permettent de garantir un niveau de généralité aux règles induites, en éliminant certaines combinaisons peu fréquentes dans l'ensemble d'apprentissage.

La sélection de variables [41] 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 [31].

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. L'objectif est de ne retenir que les variables indispensables pour la définition du contexte d'application d'une règle.

Si la variable qui différencie les règles du groupe n'a qu'une faible influence dans le contexte défini par les autres variables, alors elle peut être supprimée de la prémisse de chacune des règles. Dans ce cas les règles du groupe deviennent identiques et une seule d'entre elles est conservée.

La règle donnée en exemple ci-dessous montre la simplification des règles R1, R2 et R3, suite à l'élimination de la variable numéro deux, notée E_2, dont la partition compte trois sous-ensembles flous. La notation MF_j^i se réfère au $i^{ème}$ sous-ensemble flou de la $j^{ème}$ variable. L' élimination de la variable E_2 est faite dans le contexte où la variable numéro un est représentée par le MF^3, et la variable numéro trois par le MF^2 :

Règle *3 1 2* $\iff$ Si E_1 est $MF_1^{\mathbf{3}}$ ET E_2 est $MF_2^{\mathbf{1}}$ ET E_3 est $MF_3^{\mathbf{2}}$ Alors ...

$$\begin{cases} \text{R1 :} & 3 & 1 & 2 \\ \text{R2 :} & 3 & 2 & 2 \\ \text{R3 :} & 3 & 3 & 2 \end{cases} \iff R_g : 3\ 0\ 2 \quad \text{Si } E_1 \text{ est } MF_1^{\mathbf{3}} \text{ ET } E_3 \text{ est } MF_3^{\mathbf{2}} \text{ Alors ...}$$

Cette fusion induit une perte de spécificité : le nombre, et la diversité, des exemples qui activent la règle sont susceptibles d'augmenter. Le critère de fusion des règles du groupe vérifie l'homogénéité des valeurs de sortie (observées dans l'ensemble d'apprentissage) des exemples attirés par la règle. La fusion sera acceptée si l'indice d'hétérogénéité des sorties est inférieur à un seuil donné.

L'indice d'hétérogénéité dépend du type de la sortie. Pour une sortie continue, il s'exprime comme : $h = \frac{\sigma_r}{\sigma}$. E_r est l'ensemble des exemples qui activent le plus la règle, σ_r est l'écart type des sorties dans E_r pondéré par les degrés de vérité de la règle pour chacun des exemples, σ l'écart type dans l'ensemble complet. Pour une sortie de type classification, il s'agit d'une entropie normalisée par le logarithme du nombre de classes.

A chaque étape de l'algorithme, dans un premier temps, l'ensemble des groupes satisfaisant ce critère sont identifiés et ordonnés suivant la perte de performance que leur fusion individuelle occasionne. Dans un deuxième temps, les groupes qui engendrent la baisse de performance la plus petite sont fusionnés jusqu'à ce que la perte cumulée soit jugée tolérable, c'est-à-dire inférieure à un seuil donné.

La phase de fusion des groupes est complétée par une sélection des règles. Afin d'éliminer la redondance, nous voulons vérifier que la présence de chacune des règles dans la base est 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.

Cet algorithme permet donc une simplification du système avec une perte de performance contrôlée. Les règles incomplètes, plus faciles à analyser dès que le nombre de variables est supérieur à trois, renforcent l'interprétabilité du système. Cependant, la version actuelle manque de nuance. Deux voies

d'amélioration sont possibles. La fusion des règles d'un groupe est binaire : toutes les règles sont fusionnées ou bien aucune. Il serait bon d'introduire un peu de gradualité. D'autre part, des règles qui sont peu, voire pas, supportées par les données d'apprentissage peuvent être fusionnées au sein d'un groupe. Ce comportement est quelque peu abusif.

L'absence d'une variable dans une règle, du fait de la fusion d'un groupe peut correspondre à deux sémantiques très différentes. Soit la valeur de cette variable n'a aucune importance pour le contexte, soit cette variable est, dans le contexte de la règle, corrélée à une autre. Dans le deuxième cas, l'une ou l'autre des variables peut être utilisée pour la description de la règle. Des outils statistiques devraient permettre de distinguer ces deux situations.

3.5 Extensions et améliorations

L'ensemble des procédures d'apprentissage présentées dans cette section travaillent avec des entrées numériques. Il est indispensable de les étendre aux entrées floues. Ceci permettra de représenter l'imprécision de la valeur numérique et, plus largement, de manipuler des concepts intriséquement flous, qui ne résultent pas d'une mesure.

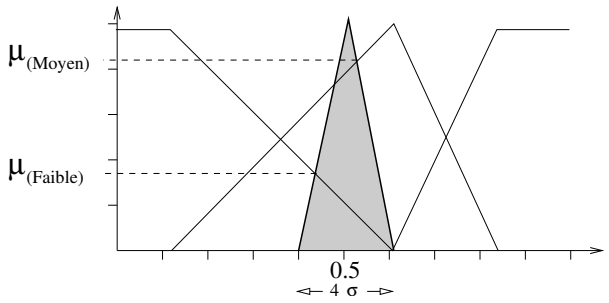


FIG. 5 – Un nombre flou

La figure 5 illustre une représentation possible d'un nombre flou, A, qui caractérise une distribution de médiane 0.5 et d'écart type $\sigma = 0.05$. Les niveau de comparabilité de l'ensemble A avec les termes linguistiques sont calculés par la traditionnelle composition Sup-Min :

$$Comp(A, M) = \sup_x \left(min\left(\mu_A(x), \mu_M(x)\right)\right)$$

.

Les systèmes d'inférence flous sont caractérisés par deux indices qui guident les procédures d'apprentissage : l'un, indice de de performance, relatif à l'erreur entre les sorties observées et les sorties inférées, l'autre, indice de couverture, qui caractérise la capacité du système à gérer la diversité des exemples de l'ensemble d'apprentissage. Ces deux indices sont globaux : ils ne font aucune distinction entre les exemples. Cette gestion pénalise les cas rares, ou exceptions. En effet, comme ils sont peu nombreux, s'ils sont mal traités par le système, cela n'aura que peu d'incidence sur les indices. Une pondération locale forcera le système à prendre ces cas rares en considération.

4 La connaissance experte

Cette section s'intéresse à la partie positive de la connaissance experte. Deux volets sont abordés : son recueil et la coopération avec les données expérimentales. L'objectif de la coopération est de tirer parti du meilleur de chacune des sources d'information. Le cadre général proposé [39] est présenté

dans la figure 6. La partie supérieure de la figure indique les tâches réalisées par l'expert, la partie inférieure l'exploitation des données. Nous retrouvons la séparation, en séquence, entre la définition des partitions et l'induction des règles. La première étape, section 4.2, consiste à définir un partitionnement qui soit cohérent avec la connaissance experte disponible et les données expérimentales. Puis, dans un deuxième temps, section 4.3, les règles expertes sont comparées avec les règles induites en vue de la fusion des deux bases. Comme les deux bases de règles utilisent les mêmes partitions, la comparaison peut se faire au seul niveau symbolique. La fusion est réalisée en vérifiant les propriétés de la base résultante : cohérence, absence de redondance, complétude.

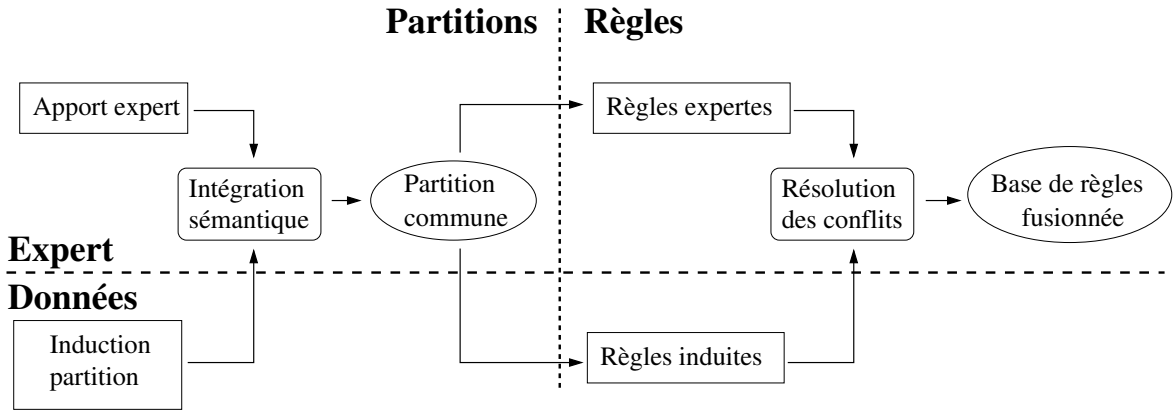


FIG. 6 – Un cadre de coopération

L'ensemble de la démarche est implémentée dans le logiciel KBCT[2]. Les principales fonctionalités du logiciel sont décrites dans [1].

Intéressons nous d'abord au recueil de la connaissance experte.

4.1 La sémantique des opérateurs OU et NON

L'acquisition de la connaissance experte est une tâche très difficile. La façon de procéder la plus simple, et la plus économique, consiste à demander à l'expert d'écrire ses propres règles de raisonnement. Le rôle des interfaces homme machine est de faciliter ce travail, notamment en proposant des modes de représentation indépendants des contraintes de leur implémentation informatique.

La base de règles doit refléter la connaissance de l'expert de la manière la plus simple et la plus interprétable possible. Parmi les approches envisagées, les règles qui utilisent des formes normales disjonctives (DNF) dans la prémisse sont particulièrement populaires. L'usage de l'opérateur *OU* permet effectivement de réduire le nombre de règles de la base tout en conservant l'interprétabilité linguistique : les experts utilisent de façon naturelle les opérateurs *OU* et *NON*. Cela rejoint la préoccupation de conception de systèmes compacts, et notamment de bases de règles compactes, partagée par de nombreuses équipes, voir [65] par exemple.

L'interprétabilité linguistique n'est pas suffisante, le niveau de l'inférence est également à considérer [40]. Pour illustrer notre propos, considérons une partition de 5 sous-ensembles comme celle de la partie supérieure de la figure 7, et la proposition $(MF_4\ OU\ MF_5)$, qui peut s'interpréter comme *Elevé OU Très elevé* si l'univers sémantique de cette variable varie de *Très faible* à *Très elevé*. Même si cette expression est parfaitement lisible, son sens dépendra étroitement de la façon de la calculer. La partie centrale de la figure montre le résultat obtenu par le maximum, qui reste la méthode la plus utilisée pour calculer une disjonction.

[2]http ://www.mat.upm.es/projects/advocate/kbct.htm

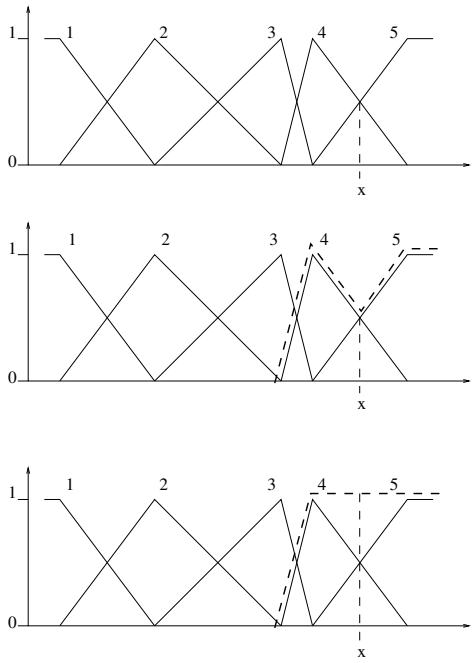


FIG. 7 – Opérateur OU

Ce résultat correspond-il vraiment à ce que nous avons en tête en écrivant $(MF_4\ OU\ MF_5)$? Le problème provient de la non convexité de l'ensemble résultant. Ainsi pour la valeur x le degré d'appartenance à l'ensemble $(MF_4\ OU\ MF_5)$ est de 0.5, alors que dans l'esprit de la personne qui a écrit cette expression, il est de 1. En effet, l'expert en utilisant la disjonction pense probablement davantage à un méta terme linguistique, qui respecte toutes le contraintes d'interprétabilité dont la convexité, qu'à une simple énumération d'options possibles. Certes, cette énumération émule bien le cas net *(crisp)* pour lequel la définition d'un intervalle discret est l'union des éléments discrets qui appartiennent à cet intervalle. Mais, dans notre contexte, l'expression disjonctive est porteuse d'un tout autre sens, qui serait, dans ce cas, exprimé par une inégalité du type *"plus grand que"*.

La sémantique attachée à l'usage de la négation peut être discutée de façon analogue.

Notons que les partitions considérées sont celles pour lesquelles les termes sont ordonnés. Ce sont les seules pour lesquelles le recouvrement a un sens. En effet, lorsque les termes sont incomparables, ce qui est le cas des variables qualitatives, par exemple des types d'aliments de bétail comme *fourrage*, *céréales* ou *protéagineux*, il n'est pas possible de définir une relation d'ordre qui prenne en compte l'ensemble de leurs caractéristiques.

Le concept de méta terme

Un des moyens de gérer ce problème est de considérer l'enveloppe convexe des termes intervenant dans la combinaison, comme illustré dans la partie inférieure de la figure 7. Pour éviter que l'enveloppe inclue des termes linguistiques indésirables, et donc préserver l'intégrité sémantique des règles, la combinaison ne peut comprendre que des termes linguistiques adjacents. Cette implémentation correspond, dans le cas des partitions floues fortes, à la t-conorme de Łukasiewicz définie comme : $S_{a,b} = min(1, a + b)$.

Au niveau numérique, calculatoire, la combinaison *OU*, $(MF_i\ OU\ MF_{i+1})$, est implémentée sous la forme d'un nouvel ensemble flou, défini comme l'enveloppe convexe des termes élémentaires de

la combinaison. Cela nous conduit au concept de partition hybride illustré sur la figure 8 : la partition inclut d'une part des termes élementaires, 5, qui définissent une partition floue forte, et d'autre part, des méta termes correspondant à l'enveloppe convexe des termes élémentaires intervenant dans une combinaison. Dans la partie médiane de la figure un nouvel ensemble flou a été construit : il correspond à la combinaison $(MF_2\ OU\ MF_3)$. Son étiquette est le premier numéro plus grand que 5 disponible, soit 6.

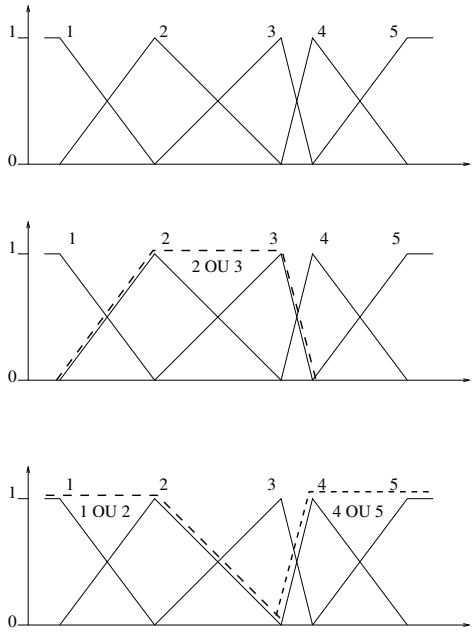


FIG. 8 – Une partition hybride

Ainsi la règle linguistique :

$$Si\ X\ est\ (MF_2\ OU\ MF_3)\ ET \ldots\ Alors\ Y\ est\ C$$

se traduit au moment de l'inférence par :

$$Si\ X\ est\ MF_6\ ET \ldots\ Alors\ Y\ est\ C$$

Le niveau de correspondance entre une valeur numérique d'entrée, x, et le terme linguistique $(MF_2\ OU\ MF_3)$ est calculé comme le degré d'appartenance de x à l'ensemble flou MF_6.

L'implémentation la plus courante de l'opérateur *NON* consiste à utiliser le complément du degré d'appartenance : $\mu_{NON(A)}(x) = 1 - \mu_A(x)$.

Pour s'assurer de la convexité de tous les ensembles flous de la partition, le complément d'un terme linguistique est défini comme deux sous-ensembles flous. Soit pour une partition qui comprend m sous-ensembles flous :

$$NON\ (MF_i) \iff \left\{ \begin{array}{l} MF_1\ OU \ldots MF_{i-1} \\ MF_{i+1}\ OU \ldots\ MF_m \end{array} \right. \tag{2}$$

Leur signification respective est *plus petit que* MF_i pour le premier et *plus grand que* MF_i pour l'autre.

La partie inférieure de la figure 8 illustre l'opérateur de négation. Au terme linguistique $NON\ (MF_3)$ correspondent deux méta termes $(MF_1\ OU\ MF_2)$ et $(MF_4\ OU\ MF_5)$, étiquetés comme MF_7 et MF_8.

La règle :

$$Si\ X\ est\ NON(MF_3)\ ET \ldots\ Alors\ Y\ est\ C$$

est implémentée sous la forme des deux règles suivantes :

$$Si\ X\ est\ MF_7\ ET \ldots\ Alors\ Y\ est\ C$$

$$Si\ X\ est\ MF_8\ ET \ldots\ Alors\ Y\ est\ C$$

Remarquons, que compte tenu des restrictions imposées aux partitions élémentaires, une seule de ces règles, au plus, est activée pour un exemple donné. L'augmentation résultante du nombre de règles n'est que peu pénalisante du point de vue calculatoire et n'affecte en rien l'interprétabilité du système.

Les méta termes ajoutés à la partition floue forte initiale ne sont pris en compte qu'au seul niveau de l'inférence.

4.2 Définir un partitionnement commun

Replaçons nous maintenant dans le cadre de la coopération entre la connaissance experte, particulièrement l'évidence positive qui la compose, et les données. La première étape, qui correspond à la partie gauche de la figure 6, est relative au partitionnement des variables. Comme présenté dans la section induction, le partitionnement est conduit au sein d'une variable, sans faire intervenir les données d'une autre, qu'elle soit d'entrée ou de sortie.

La figure 9 résume le type d'information qu'est susceptible d'apporter chacune des sources. Les niveaux d'interaction sont matérialisés par des flèches horizontales.

L'expert, dont le raisonnement est principalement basé sur des prototypes, apporte une information essentiellement qualitative. Nous lui demandons de préciser la zone d'intérêt du domaine de variation de la variable, le nombre de termes linguistiques dont il (ou elle) a besoin pour exprimer sa connaissance ou son raisonnement ainsi que les étiquettes linguistiques associées. Dans certains cas il sera capable de fournir la valeur d'un prototype, ou point modal. Par exemple 12 V pour la tension nominale d'une batterie de voiture, ou bien 37 ^{o}C pour la température corporelle d'un sujet adulte. Nous pensons qu'il est contre productif de lui demander de définir les limites et formes des sous-ensembles flous parce que ce serait l'obliger à s'exprimer dans un langage qui n'est pas le sien et parce que ces valeurs peuvent être extraites des données.

Les deux sources doivent être cohérentes sur l'ensemble des points proposés par l'expert : le domaine d'intérêt, la granularité de la partition et la valeur des points modaux. Le processus est sous le contrôle total de l'expert, c'est à lui que revient l'arbitrage en cas de conflit. Ce choix repose sur l'hypothèse de la fiabilité de la connaissance experte : les éléments qu'ils apportent sont considérés comme sûrs et ont priorité sur ceux issus des données. Comme son savoir peut également être questionné par les processus d'induction, l'expert peut, à tout moment, modifier ses saisies.

Détaillons maintenant les trois points qui doivent faire l'objet d'un accord :

– Le domaine de variation : La coopération ne pourra porter que sur la partie commune du domaine couvert par les données et du domaine d'intérêt spécifié par l'expert. Il est possible qu'une part de ce domaine ne soit pas couverte par des données, ces zones peuvent correspondre à des parties de l'espace pour lesquelles la collecte des données est coûteuse. Il appartiendra à l'expert, dans ce cas, de compléter la partition en fournissant un point modal. Il se peut également que le domaine couvert par les données ne soit pas complètement inclus dans celui indiqué par l'expert. Ces données atypiques peuvent correspondre à des valeurs erronées mais aussi à des étapes particulières du processus que l'expert ne souhaite pas modéliser, par exemple la phase de démarrage d'un procédé industriel continu est souvent considérée de façon distincte.

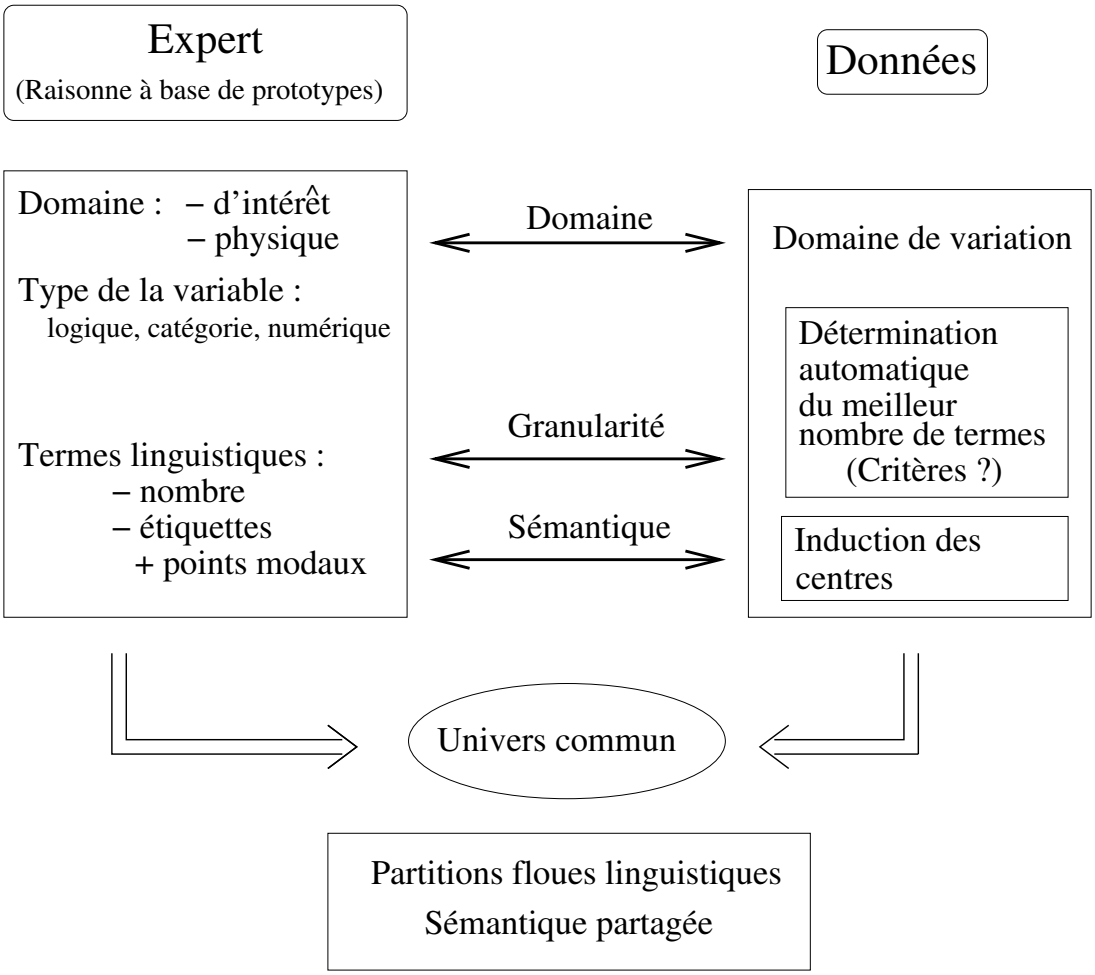


FIG. 9 – Définir un univers commun

– La granularité de la partition : Le nombre de termes dont il est question concerne la partie commune du domaine de variation. Deux options sont possibles : ou bien induire directement les centres du nombre d'ensembles défini par l'expert, ou bien induire les hiérarchies présentées dans la section précédente et ordonner les partitions suivant un critère de qualité. Dans ce cas le résultat peut être différent du nombre proposé initialement par l'expert. La décision finale est, comme pour les étapes précédentes, de la responsabilité de l'expert.
– Les centres des sous-ensembles flous : Une fois l'accord acquis sur le nombre de sous-ensembles flous, la dernière vérification est purement sémantique. Comme une partition floue forte est complètement définie par les centres des ensembles, la vérification est limitée à leur position. La valeur du centre est-elle un prototype possible pour l'étiquette considérée ? La différence éventuelle avec le point modal est-elle acceptable ? C'est à l'expert d'en décider.

Le résultat de cette procédure est donc un partitionnement, pour chacune des variables d'entrée ou de sortie, défini en accord avec les données et porteur de la sémantique de l'expert. Ces partitions sont maintenant utilisables par les procédures d'induction de règles comme pour l'expression des règles expertes.

4.3 Fusionner les bases de règles

La base de règles résultante doit vérifier certaines propriétés fondamentales, en particulier la cohérence des règles entre elles et l'absence de redondance [16].

Dans le cadre de la coopération entre l'expertise et les données, la complétude, une autre propriété fondamentate de la base de règles, ne peut être assurée que par l'expert car elle ne peut pas être garantie par les règles induites.

Du point de vue de la sémantique des règles, comme chaque règle ouvre une possibilité sans garantir aucune nécessité, il ne peut y avoir de contradiction. Cependant, si l'on se place du point de vue de la représentation des connaissances deux règles ayant les mêmes prémisses et des conclusions différentes sont contradictoires. C'est pourquoi nous avons proposé une typologie des conflits [38] dans ce type de systèmes, les systèmes de Mamdani, susceptibles d'affecter l'interprétabilité de la base de règles.

La détection des conflits s'appuie sur une comparaison des règles et en particulier des espaces couverts par chacune des règles. Grâce au partionnement commun présenté dans la section précédente, la comparaison des règles se fait à un niveau purement symbolique.

Chacune des règles de la base opère dans une région de l'espace définie par les variables d'entrée et de sortie. Le tableau 1 présente les différentes relations possibles, pour deux règles, entre l'espace d'entrée couvert par chacune d'elle ainsi que deux cas pour les conséquences : elles peuvent être identiques ou différentes. Il serait bien entendu possible d'étendre cette étude en considérant la relation d'inclusion dans l'espace de la sortie.

Comme la détection des conflits s'effectue à un niveau linguistique, la règle doit être interprétée au seul niveau symbolique. Traitant de la même variable X, les concepts "$X\ est\ MF_1$" et "$X\ est\ MF_2$" sont distincts, même si les ensembles flous correspondants ont une intersection non vide. La partie commune ne constitue en rien une zone redondante : du fait de la multi-appartenance, elle permet l'interpolation entre les règles.

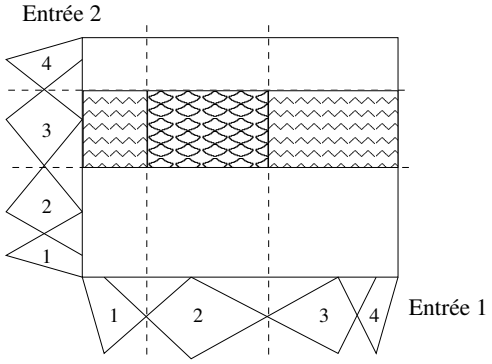
Entrée \ Sortie	$=$	$\neq$
$=$	1	2
$\subset$	3	4
$\cap \neq \emptyset$	5	6
$\cap = \emptyset$	Pas de problème	

TAB. 1 – Les situations prises en compte

Il est évident que si les espaces d'entrée couverts par deux règles sont identiques, alors l'un est inclus dans l'autre et leur intersection n'est pas vide. Aussi dans le tableau 1 *inclus* signifie différent mais l'un inclus dans l'autre, *intersection non vide* se rapporte à des espaces qui ne sont pas inclus l'un dans l'autre mais qui couvrent une partie commune de l'espace. La figure 10 montre un exemple de la relation *inclus*, tandis que la figure 11 illustre le cas de l'*intersection non vide*.

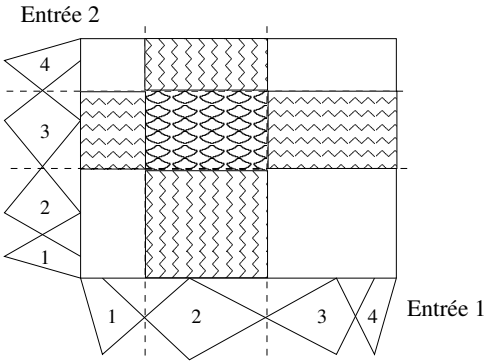
Les cas répertoriés dans le tableau 1 portant des numéros impairs correspondent à des situations de redondance partielle ou totale, les numéros pairs à des situations d'incohérence potentielle ou établie. Une gestion des ces différentes situations, proposée dans [38] est implémentée dans KBCT.

L'analyse de la base de règles pour détecter ces situations de conflit doit prendre en compte les règles utilisant les opérateurs *OU* ou *NON*. Comme cette analyse est réalisée par une procédure automatique pour laquelle l'interprétabilité, et particulièrement le nombre de règles, n'est pas très importante, nous proposons d'écrire chacune des règles sous la forme d'un ensemble de règles canoniques.



R_1 : Si X_1 est MF_2 Et X_2 est MF_3 Alors Y est ...
R_2 : Si X_2 est MF_3 Alors Y est ...

FIG. 10 – L'espace d'entrée couvert par R1 est inclus dans celui couvert par R2



R_1 : Si X_1 est MF_2 Alors Y est ...
R_2 : Si X_2 est MF_3 Alors Y est ...

FIG. 11 – Deux règles couvrant une partie d'espace d'entrée commune

Une règle canonique est définie seulement à partir de l'opérateur *ET* et comporte au plus un terme linguistique par variable.

Ainsi une règle contenant une combinaison *OU* de k termes sera exprimée sous la forme de k règles. Un *NON* appliqué à un terme d'une partition qui en compte m générera $m-1$ règles.

Si la partition de la variable X_2 est formée de 4 sous-ensembles flous, la règle :

$$Si\ X_1\ est\ (MF_3\ OU\ MF_4)\ ET\ X_2\ est\ NON\ MF_3\ Alors\ Y\ est\ C_1$$

sera transcrite sous la forme des 6 règles canoniques suivantes :

$$Si\ X_1\ est\ MF_3\ ET\ X_2\ est\ MF_1\ Alors\ Y\ est\ C_1$$

$$Si\ X_1\ est\ MF_3\ ET\ X_2\ est\ MF_2\ Alors\ Y\ est\ C_1$$

$$Si\ X_1\ est\ MF_3\ ET\ X_2\ est\ MF_4\ Alors\ Y\ est\ C_1$$

$$Si\ X_1\ est\ MF_4\ ET\ X_2\ est\ MF_1\ Alors\ Y\ est\ C_1$$

$$Si\ X_1\ est\ MF_4\ ET\ X_2\ est\ MF_2\ Alors\ Y\ est\ C_1$$
$$Si\ X_1\ est\ MF_4\ ET\ X_2\ est\ MF_4\ Alors\ Y\ est\ C_1$$

Ce principe d'expansion permet la détection des conflits, au niveau linguistique, en mobilisant seulement les termes élémentaires de la partition et l'opérateur *ET*.

4.4 Simplification

La procédure de simplification présentée dans la section induction ne s'applique pas à un système contenant de la connaissance experte. En effet le nombre d'exemples qui confirment une règle experte peut être nul, mais il faut prendre en compte le nombre de ceux qui la contredisent à un certain niveau.

D'autre part, la procédure devra également simplifier l'expression des règles par l'usage des opérateurs *OU* et *NON* présentés dans cette section. Cette expression pourra même conduire à la fusion permanente de sous-ensembles flous au sein d'une partition.

Ce travail sera abordé par José María Alonsa Moral dans la cadre de sa thèse.

5 Perspectives

La figure 1, qui résume le cadre de représentation que nous proposons, n'a pas été complètement commentée dans ce mémoire. Certaines parties sont, pour l'instant, à l'état de perspectives. La prise en compte de la connaissance négative est l'objet d'une thèse en cours, la mise à jour de la base de règles sera envisagée à moyen terme.

D'autre part, la part croissante prise par les approches spatiales au sein des problématiques du Cemagref nous conduit à nous intéresser à la gestion de variables spatiales dans le raisonnement flou ainsi qu'à la fusion de données.

5.1 La connaissance négative

Les règles induites des données, comme les règles expertes qui modélisent la connaissance positive, sont des règles qui ouvrent une possibilité. La sémantique de la règle $A \rightarrow B$ est la suivante : si je suis dans la situation A, alors B est une conclusion garantie possible. Ce type de règles est à la base du raisonnement par similarité : la sortie est garantie possible car elle a déjà été observée dans un cas semblable. En l'absence de règles, aucune valeur de sortie n'est garantie possible, et plus un exemple active de règles, plus l'ensemble des valeurs possibles s'élargit. L'agrégation des règles est disjonctive et s'effectue après l'inférence.

Dans le cas des règles implicatives, qui sont utilisées pour la prise en compte des contraintes, le processus est différent. En l'absence de règles la totalité de l'univers de la variable de sortie constitue l'ensemble des conclusions possibles et chacune des règles restreint cet ensemble. Dans ce cas, il faut d'abord agréger les règles, de manière conjonctive, puis, inférer une valeur de sortie parmi celles qui restent possibles.

Les règles conjonctives et les règles implicatives constituent deux grandes familles. Dubois et al. [19] proposent plusieurs types de règles au sein de ces deux familles. Le premier travail d'Hazaël Jones, qui réalise ces travaux dans le cadre de sa thèse, sera d'élaborer une typologie des règles expertes susceptibles de nous intéresser. Cette élaboration n'a rien de trivial. Elle suppose un recueil, et une mise en forme, de la connaissance experte disponible dans le cadre d'applications représentatives de celles que nous voulons traiter. D'autre part, le mécanisme d'inférence de ce type de règles nécessite le développement d'algorithmes spécifiques efficaces pour rendre leur utilisation possible dans des situations réelles.

La difficulté du processus d'intégration réside dans la coexistence de ces différents types de règles au sein d'une base de règles floues, dans la définition d'un raisonnement approprié, leur hiérarchisation et leurs représentations différentes de l'ignorance.

5.2 La mise à jour de la base de règles

Lorsque la base de règles contient des connaissances issues de différentes sources et résultant d'un processus d'intégration, il est nécessaire de prévoir des procédures de mise à jour incrémentales.

Ces procédures doivent pouvoir être exécutées lorsque l'utilisateur dispose d'éléments nouveaux, des données ou bien de la connaissance experte. Elles pourraient aussi être utilisées, en analysant le degré de vérité des règles, pour la détection de dérives ou le diagnostic de dysfonctionnements du système.

5.3 La prise en compte du caractère spatial

Nous proposons d'élargir l'approche présentée au cas des données spatialisées. La spatialisation des données est en effet un élément clé des agro et éco-systèmes sur lesquels se concentre aujourd'hui l'activité du Cemagref, dans le cadre notamment du TR CASYS au sein de de l'UMR Itap.

Nos projets s'incrivent dans une double dynamique.

D'une part, le stage de Mathieu Grelier a montré l'intérêt de méthodes d'induction de règles floues pour la viticulture de précision [27]. Il propose des systèmes d'inférence floue pour modéliser la relation entre le taux de sucre au moment de la vendange et des variables explicatives, pédologiques et physiologiques. Les règles induites sont très expressives pour l'expert. Cependant, dans sa configuration actuelle, le logiciel FisPro ne peut pas gérer la variable zone, qui correspond au domaine spatial de validité d'une donnée.

D'autre part, Jean-Noël Paoli vient de soutenir une thèse sur la fusion de données géoréférencées. Une donnée est représentée par deux valeurs incertaines : la valeur d'un paramètre, mesure ou label linguistique attribué par un expert, et la zone de validité de celle-ci. La première composante est modélisée par un ensemble flou. Pour représenter la seconde, il propose le concept de sous-ensemble flou géographique, une fonction d'appartenance en deux dimensions [50].

Notre objectif, à travers le séjour post doctoral de Laurent Lardon, est donc l'implémentation dans FisPro de la variable zone. Les zones pourront être définies manuellement par l'expert. Mais également des algorithmes d'induction, à partir de la cartographie d'un ou plusieurs indices, seront développés. Ils pourront s'inspirer de méthodes de segmentation en régions [21], ou bien de méthodes de triangulation multi-dimensionnelle, comme celle de Delaunay par exemple [3], ou encore des algorithmes de ligne de partage des eaux [60]. Ils devront surtout respecter des contraintes d'ordre sémantique. En effet, une variable spatiale ne peut pas être réduite à une combinaison de deux variables, x et y. Elle implique une sémantique, une relation avec l'expert humain qui va l'utiliser dans son raisonnement. Les contraintes sémantiques relatives aux partitions floues mono-dimensionnelles sont à adapter à un espace de dimension 2, avant de les imposer à l'algorithme.

Enfin, une utilisation conviviale demandera la réalisation de modules de visualisation et d'analyse spécifiques, comme par exemple la cartographie de l'influence d'une règle ou de chacun des labels d'une partition mono-dimensionnelle, ainsi que l'interface avec les principaux systèmes d'information géographiques. Ces développements s'appuyeront sur des applications agronomiques.

5.4 La fusion de données

La fusion de données est un de nos centres d'intérêt actuels. Nous l'abordons à travers un outil générique, l'intégrale de Choquet [26], dans deux cadres distincts.

Avec Jean-Noël Paoli nous l'utilisons dans une perspective d'estimation spatiale de la valeur d'un paramètre [50]. Il s'agit d'agréger différentes sources d'information disponibles, à des degrés divers, pour une zone de requête afin d'en déduire une valeur caractérisant la zone. Deux évolutions sont à envisager au cours du séjour post doctoral de Jean-Noël. La première concerne la prise en compte de l'interaction entre les sources. En effet, dans sa version actuelle, le poids d'une donnée dans le processus d'estimation est seulement fixé par sa position relative par rapport à la zone de requête, sans tenir compte de son voisinage. Il paraît légitime d'accorder un niveau de confiance plus élevé à des sources proches qui indiquent la même tendance et d'en accorder moins à celles, également proches, qui se contredisent. La seconde vise à prendre en compte la répartition des données au sein de la zone de requête. Actuellement, seul est pris en compte le nombre de données pondéré par le niveau d'intersection avec la zone de requête. Cela ne permet pas de différencier la situation qui correspond à des sources uniformément distribuées sur l'ensemble de la zone de celle pour laquelle les données sont regroupées dans une petite portion de l'espace.

Avec Pilar Bulacio, le même formalisme est utilisé pour intégrer les sorties de plusieurs systèmes de décision. Les premiers résultats montrent l'efficacité de la coopération [6]. Deux voies sont à explorer dans le cadre d'un système multi-agents distribué, du fait du nombre important de systèmes de décision susceptibles d'être présents. Il s'agira d'une part de vérifier dans quelles conditions les algorithmes proposés peuvent supporter ce grand nombre d'acteurs. Ce problème est identifié sous le nom de "scalability". D'autre part, il faudra proposer le sous-ensemble optimal de systèmes pour réduire les temps de calcul et, surtout, éviter la redondance.

Références

[1] José Alonso, Luis Magdalena, and Serge Guillaume. Kbct : A knowledge extraction and representation tool for fuzzy logic based systems. In IEEE, editor, *FUZZ'IEEE 2004*, pages 989–994, July 2004.

[2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Functions Algorithms*. Plenum Press, New York, 1981.

[3] Houman Borouchaki and Paul Louis George. Encore delaunay. Rapport de recherche 2461, Inria, France, 1995.

[4] Bernadette Bouchon-Meunier and Christophe Marsala, editors. *Logique floue, principes, aide à la décision*. Lavoisier, 2003.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont CA, 1984.

[6] Pilar Bulacio, Luis Magdalena, and Serge Guillaume. *Arquitectura Abierta para la Toma de Decisión Conjunta*, pages 1–12. In Press, 2005.

[7] Jorge Casillas, Oscar Cordon, Francisco Herrera, and Luis Magdalena. *Accuracy Improvements in Linguistic Fuzzy Modeling*, volume 129 of Studies in Fuzziness and Soft Computing. Springer, 2003.

[8] Jorge Casillas, Oscar Cordon, Francisco Herrera, and Luis Magdalena. *Interpretability Issues in Fuzzy Modeling*, volume 128 of Studies in Fuzziness and Soft Computing. Springer, 2003.

[9] S. Chen, S. A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *Int. J. Control*, 50 :1873–1896, 1989.

[10] S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2 (No 2) :302–309, March 1991.

[11] A.M.G. Cornelissen, J. van der Bergand W.J. Koops, and U. Kaymak. Elicitation of expert knowledge for fuzzy evaluation of agricultural production systems. *Agriculture, Ecosystems and Environment*, 95 :1–18, 2003.

[12] J. Valente de Oliveira. Semantic constraints for membership functions optimization. *IEEE Transactions on Systems, Man and Cybernetics. Part A*, 29(1) :128–138, 1999.

[13] Sébastien Destercke, Serge Guillaume, and Brigitte Charnomordic. Amélioration de l'interprétabilité d'un algorithme classique d'induction de règles floues. In *LFA'04*, pages 247–254, Toulouse, France, Novembre 2004. Cépaduès Editions.

[14] D. Dubois and H. Prade. Fuzzy sets in approximate reasoning, part 1 : Inference with possibility distributions. *Fuzzy Sets and Systems*, 40 :143–202, 1991.

[15] D. Dubois and H. Prade. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems*, 84(2) :169–186, 1996.

[16] D. Dubois, H. Prade, and L. Ughetto. Checking the coherence and redundancy of fuzzy knowledge bases. *IEEE Trans. on Fuzzy Systems*, 5 (5) :398–417, 1997.

[17] Didier Dubois, Eyke Hullermeier, and Henri Prade. Fuzzy set-based methods in instance-based reasoning. *IEEE Transactions on Fuzzy Systems*, 10 (3) :322–332, 2002.

[18] Didier Dubois and Henri Prade, editors. *Fundamentals of fuzzy sets*. Kluwer Academic Publishers, 2000.

[19] Didier Dubois, Henri Prade, and Laurent Ughetto. A new perspective on reasoning with fuzzy rules. *International Journal of Intelligent Systems*, 18 :541–567, 2003.

[20] Jairo Espinosa and Joos Vandewalle. Constructing fuzzy models with linguistic integrity from numerical data-afreli algorithm. *IEEE Transactions on Fuzzy Systems*, 8 (5) :591–600, October 2000.

[21] Christophe Fiorio and Jens Gustedt. Two linear time union-find strategies for image processing. *Theoretical Computer Sciences*, 54 :165–181, 1996.

[22] Y. Fukuyama and M. Sugeno. A new method of choosing the number of clusters for fuzzy c-means method. In *Proceedings of the 5th Fuzzy system symposium (in Japanese)*, pages 247–250, 1989.

[23] Louis Gacogne. *Eléments de logique floue*. Editions Hermès, Paris, 1997.

[24] Pierre-Yves Glorennec. Quelques aspects analytiques des systèmes d'inférence floue. *Journal Européen des Systèmes automatisés*, 30 (2-3) :231–254, 1996.

[25] Pierre-Yves Glorennec. *Algorithmes d'apprentissage pour systèmes d'inférence floue*. Editions Hermès, Paris, 1999.

[26] M. Grabisch, T. Murofushi, and M. Sugeno. *Fuzzy Measures and Integrals. Theory and Applications (edited volume)*. Studies in Fuzziness. Physica Verlag, 2000.

[27] Mathieu Grelier, Serge Guillaume, Bruno Tisseyre, and Thibaut Scholasch. Interpretable fuzzy rule induction in precision agriculture. *Computers and electronics in agriculture*, Submited, 2005.

[28] Serge Guillaume. Designing fuzzy inference systems from data : an interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9 (3) :426–443, June 2001.

[29] Serge Guillaume and Brigitte Charnomordic. Knowledge discovery for control purposes in food industry databases. *Fuzzy sets and Systems*, 122 (3) :487–497, September 2001.

[30] Serge Guillaume and Brigitte Charnomordic. La logique floue au service de la vinification : relations entre pratiques et couleur du vin. In *Colloque Automatique et Agronomie, Texte des conferences*, pages 131–142, Montpellier, France, Janvier 2003. INRA.

[31] Serge Guillaume and Brigitte Charnomordic. *A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference systems from data*, pages 148–175. Volume 128 of Studies in Fuzziness and Soft Computing of [8], 2003.

[32] Serge Guillaume and Brigitte Charnomordic. *Fuzzy Inference Systems to Model Sensory Evaluation*, pages 197–216. Intelligent Sensory Evaluation- Methodologies and Applications, Springer, 2004.

[33] Serge Guillaume and Brigitte Charnomordic. Fuzzy models to deal with sensory data in food industry. *Journal of Donghua University*, 21 (3) :43–48, June 2004.

[34] Serge Guillaume and Brigitte Charnomordic. Generating an interpretable family of fuzzy partitions. *IEEE Transactions on Fuzzy Systems*, 12 (3) :324– 335, June 2004.

[35] Serge Guillaume and Brigitte Charnomordic. La logique floue pour l'extraction de connaissances à partir de données en oenologie. application à la couleur du vin rouge. *Revue française de l'oenologie*, 211 :24–31, mars/avril 2005.

[36] Serge Guillaume, Brigitte Charnomordic, and Jean-Luc Lablée. Fispro : An open source portable software for fuzzy inference systems. *http ://www.inra.fr/bia/M/fispro*, 2002.

[37] Serge Guillaume, Brigitte Charnomordic, and André Titli. A distance metric suitable for fuzzy partitioning. In IEEE, editor, *FUZZ'IEEE 2001*, pages 264–267, December 2001.

[38] Serge Guillaume and Luis Magdalena. Typologie des conflits dans les systèmes de mamdani. In *LFA'03*, pages 73–80, Toulouse, France, Novembre 2003. Cépaduès Editions.

[39] Serge Guillaume and Luis Magdalena. Expert guided integration of induced knowledge into a fuzzy knowledge base. *Soft computing*, In Press, 2005.

[40] Serge Guillaume and Luis Magdalena. An or and not implementation that improves linguistic rule interpretability. In *IFSA World Congress, In Press*, Beijing, China, July 2005.

[41] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[42] J. Hohensohn and J. M. Mendel. Two pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems. In *Proc. IEEE Conf. Fuzzy Syst.*, pages 696–700, Orlando, Florida, June 1994.

[43] H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi. Neuro-fuzzy id3 : a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning. *Fuzzy Sets and Systems*, 81 :157–167, 1996.

[44] Oleg I. Larichev. Close imitation of expert knowledge : the problem and methods. *International Journal of Information Technology & Decision Making*, 1(1) :27–42, 2002.

[45] Luis Magdalena, Miguel A. Sotelo, Serge Guillaume, Luis M. Bergasa, and José M. Alonso. A fuzzy-based system for intelligent diagnosis of motion problems on ground robots. In *IPMU 04*, pages 151–158, Italy, July 2004. Editrice Universita La Sapienza.

[46] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of Man-Machine Studies*, 7 :1–13, 1975.

[47] Swarup Medasani, Jaeseok Kim, and Raghu Krishnapuram. An overview of membership function generation techniques for pattern recognition. *International Journal of Approximate Reasoning*, 19 :391–417, 1998.

[48] G. Miller. The magical number seven, plus or minus two. *The Psychological Review*, 63 :81–97, 1956.

[49] Jose Maria Alonso Moral, Serge Guillaume, and Luis Magdalena. Kbct : A knowledge management tool for fuzzy inference systems. *http ://www.mat.upm.es/projects/advocate/kbct.htm*, 2003.

[50] Jean-Noel Paoli, Olivier Strauss, Bruno Tisseyre, Jean-Michel Roger, and Serge Guillaume. Fusion de données géoréférencées. In *LFA'04*, pages 77–84, Toulouse, France, Novembre 2004. Cépaduès Editions.

[51] Witold Pedrycz. Why triangular membership functions ? *Fuzzy sets and Systems*, 64 (1) :21–30, 1994.

[52] Witold Pedrycz. Fuzzy equalization in the construction of fuzzy sets. *Fuzzy Sets and Systems*, 119(2) :329–335, 2001.

[53] Witold Pedrycz and George Vukovich. On elicitation of membership functions. *IEEE Trans. Systems, Man and Cybernetics-Part A :Systems and humans*, 32 (6) :761–767, 2002.

[54] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1 :81–106, August 1986.

[55] Asim Roy. On connectionism, rule extraction, and brain-like learning. *IEEE Transactions on Fuzzy Systems*, 8 (2) :222–227, April 2000.

[56] Enrique H. Ruspini. A new approach to clustering. *Information and Control*, 15 :22–32, 1969.

[57] Guus Shreiber, Bob Wielinga, and Joost Breuker. *KADS- A principled approach to knowledge-based system development*. Knowledge-based systems. Academic Press, Harcourt Brace Jovanovich, Publishers, 1993.

[58] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on System Man and Cybernetics*, 15 :116–132, 1985.

[59] André Titli and Laurent Foulloy, editors. *Logique floue*. Observatoire français des techniques avancés, n^{o}14, Editions Masson, 1994.

[60] Luc Vincent and Pierre Soille. Watersheds in digital spaces : An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (6) :583–598, 1991.

[61] Li-Xin Wang and Jerry M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE Transactions on Neural Networks*, 3 :807–814, 1992.

[62] Li-Xin Wang and Jerry M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22 (6) :1414–1427, November/December 1992.

[63] J. Weisbrod. A new approach to fuzzy reasoning. *Soft Computing*, 2 :89–99, 1998.

[64] X. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (8) :841–847, 1991.

[65] Tang-Kai Yin. A characteristic-point-based fuzzy inference system aimed to minimize the number of fuzzy rules. *IEEE Transactions on Fuzzy Systems*, 12 (2) :250–273, April 2004.

[66] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8 :338–353, 1965.

A Petit glossaire du flou

Ce glossaire est extrait de la documentation de FisPro.

– 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 12 montre un ensemble de forme triangulaire.

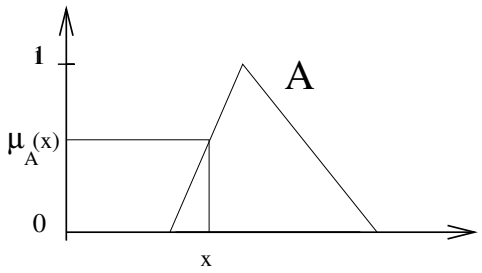


FIG. 12 – Un ensemble flou de forme triangulaire

– 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 appelé 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 $\forall x \in X_i, \sum_j \mu_{A_j^i}(x) = 1$.

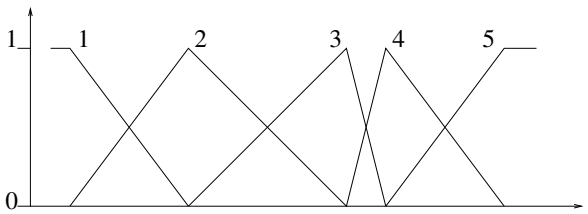


FIG. 13 – Exemple de partition floue forte

– 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 :

$$SI\ x_1\ est\ A_1^i\ ET\ \ldots\ ET\ x_p\ est\ A_p^i\ ALORS\ y_1\ est\ C_1^i\ \ldots\ ET\ y_q\ est\ C_q^i$$

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 : $y_j^i = f_j^i(x)$.
Pour des raisons d'interprétabilité, dans FisPro, la sortie est une constante au lieu de la traditionnelle combinaison linéaire des entrées.

– 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.

– 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 : $w_i = \mu_{A_1^i}(x_1) \wedge \ldots \wedge \mu_{A_p^i}(x_p)$, où $\mu_{A_j^i}(x_j)$ est le degré d'appartenance de la valeur x_j à l'ensemble flou A_j^i.

– Prototype d'une règle : un exemple est un prototype d'une règle si son degré de vérité pour cette règle vaut un, donc si les valeurs de l'exemple sont des prototypes de tous les sous-ensembles flous de la prémisse.

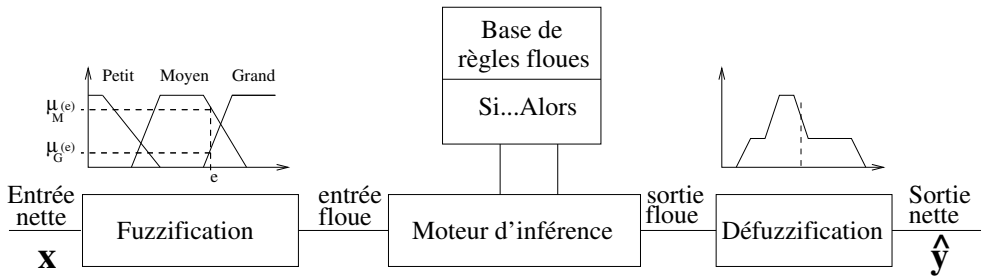


FIG. 14 – Un système d'inférence floue

– Système d'inférence floue : Un système d'inférence floue est formé de trois blocs comme indiqué sur la figure 14. 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.

– Sortie inférée par le système : Soit $\widehat{y_i}$ la sortie inférée pour l'exemple i. Elle dépend bien entendu de la base de règles mais aussi des opérateurs d'agrégation et de défuzzification, et de la nature de la sortie : nette ou floue.

L'agrégation des règles est disjonctive, signifiant que chaque règle ouvre une nouvelle possibilité pour la sortie. Les deux principaux opérateurs sont le *maximum* et la *somme*. Les conclusions des règles pour la sortie sont des valeurs numériques pour une sortie nette ou bien des numéros de sous-ensembles flous pour une sortie floue. On obtient ainsi un ensemble de valeurs possibles.

Soient m le nombre de termes linguistiques de la partition de la variable de sortie (sortie floue) ou le nombre de valeurs différentes des conclusions des règles (sortie nette), et C^r la conclusion de la règle r.
Les niveaux d'activation résultant de l'agrégation sont :
- max : $\forall j = 1, \ldots, m$
  $W^j = \left\{ \underset{r}{max} \left(w^r(x) \right) \mid C^r = j \right\}$
- sum :
  - sortie nette $\forall j = 1, \ldots, m$
    $W^j = \sum\limits_{r} \left(w^r(x) \right) \mid C^r = j$
  - sortie floue $\forall j = 1, \ldots, m$
    $W^j = min \left(1, \left\{ \sum\limits_{r} \left(w^r(x) \right) \mid C^r = j \right\} \right)$
- La défuzzification : Cette opération utilise les résultats de l'agrégation.

Les opérateurs de défuzzification sont différents selon le type de sortie, nette ou floue.
- sortie nette
  1. opérateur de *sugeno*
  $$\widehat{y}_i = \frac{\sum\limits_{j=1}^{m} W^j C^j}{\sum\limits_{j=1}^{m} W^j} \tag{3}$$
  2. opérateur *max net*
  $$\widehat{y}_i = \left\{ j = argmax \left(W^j \right) \mid j = 1 \ldots m \right\}$$
- sortie floue
  La figure 15 illustre le processus de défuzzification pour une sortie floue.

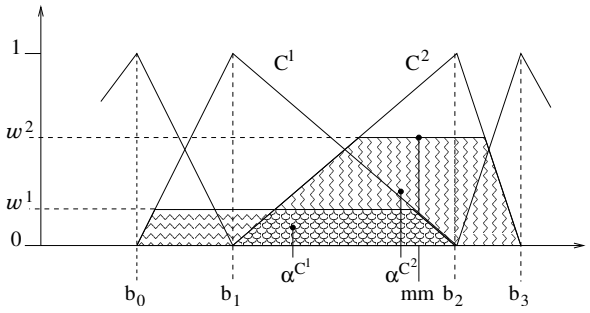


FIG. 15 – Un exemple de défuzzification

1. *pondération par les aires* Cet opérateur favorise l'interpolation entre termes linguistiques. La sortie est calculée suivant l'équation 4.

$$\widehat{y}_i = \frac{\sum_{j=1}^{m} \alpha^{C^j}\ area(C_\alpha^j)}{\sum_{j=1}^{m} area(C_\alpha^j)} \tag{4}$$

où m est le nombre de sous-ensembles flous dans la partition, $\alpha = W^j$ est le niveau d'activation résultant de l'ensemble j, α^{C^j} est l'abscisse du centre de gravité de C_α^j, et C_α^j un nouvel ensemble flou, défini à partir de C^j comme :

$$\mu^{C_\alpha^j}(x_i) = \begin{cases} \mu(x_i) & if\ \mu^{C^j}(x_i) \leq \alpha \\ \alpha & sinon \end{cases}$$

2. *moyenne des maxima* La sortie vaut $\widehat{y}_i = mm$ (figure 15). Cet opérateur considère seulement le segment correspondant au niveau d'activation maximum. Aussi, il travaille principalement au sein d'un terme linguistique.

– 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 : notée EQM, elle est calculée comme :

$$EQM = \frac{1}{n} \sum_{i=1}^{n} \|\widehat{y}_i - y_i\|^2$$

– Indice de performance. Son expression en régression est :

$$PI = \frac{1}{n} \sqrt{\sum_{i=1}^{n} \|\widehat{y}_i - y_i\|^2} = \frac{\sqrt{EQM}}{\sqrt{n}}$$

Il est homogène à une mesure, contrairement à l'EQM.
En classification, $mc(i)$ vaut 0 si l'exemple est bien classé, 1 sinon, il s'exprime comme :

$$PI = \sum_{i=1}^{n} mc(i)$$