FisPro : Un logiciel dédié aux systèmes d'inférence floue

contact@fispro.org
Version 3.6

15 mars 2018

FisPro (*Fuzzy Inference System Professional*)  permet de créer des systèmes d'inférence floue, et de les utiliser à des fins de raisonnement, en particulier pour la simulation d'un système physique ou biologique. Les systèmes d'inférence floue sont décrits brièvement dans le glossaire de logique floue donné dans ce document. Ils fonctionnent à partir de règles de raisonnement floues, qui ont l'avantage de gérer la progressivité des phénomènes.

L'implémentation faite dans FisPro permet tout d'abord de créer directement des systèmes à partir de la connaissance experte d'un domaine, par exemple en œnologie. Cette démarche est illustrée par un exemple donné dans le guide *Débuter avec FisPro*.

FisPro permet aussi de construire entièrement un système d'inférence floue à partir des données numériques du problème que l'on souhaite modéliser. Beaucoup de méthodes d'apprentissage automatique conduisent malheureusement à des systèmes de type "boîte noire". Dans FisPro, pour que l'utilisateur puisse comprendre le fonctionnement du système, des contraintes sont imposées aux algorithmes pour rendre les règles de raisonnement interprétables ([16]). Cette démarche novatrice constitue une des originalités du logiciel. Quelques exemples sont présentés dans le guide *Apprentissage avec FisPro*.

Les deux approches, écriture des règles par l'expert et apprentissage automatique, peuvent être combinées pour créer des systèmes plus complets et performants. FisPro intègre des outils à vocation pédagogique, pour illustrer le méca-

nisme de raisonnement, et d'autres permettant de mesurer la performance d'un système sur un jeu de données.

Ce logiciel est formé de deux parties distinctes : une bibliothèque de fonctions, écrite en C++, qui peut être utilisée de manière autonome et une interface utilisateur, écrite en Java, qui en implémente les principales fonctionnalités. Portable, il peut s'exécuter sur la majorité des plates-formes informatiques existantes.

L'utilisateur non familier avec la logique floue pourra commencer par une lecture du glossaire.

Les références suivantes sont conseillées pour une bonne prise en main de FisPro :

— Learning interpretable fuzzy inference systems with FisPro
Information Sciences, 181 :4409-4427, 2011.

— Fuzzy inference systems : An integrated modeling environment
for collaboration between expert knowledge and data using FisPro
Expert Systems with Applications 39 :8744-8755, 2012.

D'autres publications, relatives aux méthodes ou bien à des applications conduites avec *FisPro*, sont disponibles dans la page *Publications*.

Auteurs

— Conception et implémentation C++ : Serge GUILLAUME, Irstea, UMR ITAP (`http://ser.gui.free.fr/homepage`),
Brigitte CHARNOMORDIC, INRA, UMR MISTEA
(`http://www.inra.fr`)

— Interface java : Jean-Luc LABLEE, Irstea, UMR ITAP

— Contributions :
  — C++
    — François OLIVIER, module optimisation, d'après l'ouvrage de Pierre-Yves GLORENNEC, *Algorithmes d'apprentissage pour systèmes d'inférence floue,* paru aux Editions Hermès en 1999 ;
    — Sébastien DESTERCKE, induction de règles par moindres carrés (ols)
    — Vincent THERRY, avec l'appui de la société Envilys, chaînage de systèmes (superfis)
    — Russel STANDISH, version optimisée et parallélisable (standard OPENMP)
    — Hazaël JONES, conception de la partie *règles implicatives*
    — Lydie DESPERBEN, implémentation C++ de la partie *règles implicatives*
  — Java

- Pierre-Marie BOYER, interface JNI (Java-C++) ;
- Mathieu GRELIER, visualisation des données (Table, 2D et 3D) ;
- Anne TIREAU, interface java de la partie *règles implicatives*.

— Communication
- Jean-Michel FATOU, icônes de FisPro et conception graphique du site WEB
- Moacir Jr PEDROSO, traduction de l'interface en portuguais
- Juan Luis CORTI, traduction de *Débuter avec FisPro* en espagnol

Remerciements

Le développement initial de FisPro a bénéficié du soutien de fonds publics, Etat français et région Languedoc-Roussillon, dans le cadre d'un projet de recherche, COST 2000-012, coordonné par l'association TRANSFERTS LR et dont le partenaire industriel était la *cave coopérative "La Malepère", Arzens, Aude*.

Table des matières

Première partie
Un environnement convivial et puissant

Le lancement du programme ouvre la fenêtre principale. Plusieurs menus, qui seront détaillés dans cette section sont disponibles. Certaines informations de configuration : langue, répertoire de travail, sont mémorisées dans le fichier *fispro.cnf*, conservé dans le répertoire d'installation de FisPro.

L'opérateur de conjonction des prémisses, qui permet de calculer le degré de vérité des règles, est sélectionné dans la fenêtre principale. Trois possibilités offertes : le produit, le minimum et celui de Lukasiewicz, $max(0, a + b - 1)$.

Dans un système à plusieurs sorties, les opérateurs d'agrégation et de défuzzification (voir 1) peuvent changer selon la sortie. Ils ne sont donc pas définis dans la fenêtre principale, mais dans chaque fenêtre sortie.

Menus contextuels : dans les fenêtres Entrée, Sortie du menu *Sif* et la fenêtre Table du menu *Données*, des menus contextuels sont accessibles à l'aide du bouton droit de la souris.

Impression et exportation de graphiques : dans les fenêtres Entrée, Sortie du menu *Sif*, dans la fenêtre Règles, dans les fenêtres de visualisation 2D et 3D des menus *Sif* et *Données*, dans la fenêtre de visualisation des arbres de décision, deux options *Imprimer* et *Exporter* sont disponibles. Elles permettent respectivement d'imprimer le graphique visible dans la fenêtre, et de l'exporter dans un format graphique courant (eps, jpeg, gif, png, pdf).

1 Menu Sif

La fonction *Nouveau* permet de créer graphiquement un système d'inférence floue (SIF) en définissant les entrées, et pour chacune d'entre elles son partitionnement, la ou les sorties, ainsi que les règles d'inférence.

Remarque : La définition du SIF peut être manuelle, ou, si un fichier de données est ouvert (menu *Données*), semi-automatique : voir les options *Générer un SIF sans règles, Générer règles et Générer Conclusions*

Il est naturellement possible d'enregistrer et de recharger une configuration donnée. Soulignons que tous les fichiers sont créés au format texte, ils sont donc lisibles et modifiables à partir d'un éditeur quelconque.

Chacun des composants de base du SIF, entrée, sortie, ou règle, peut être à tout moment actif ou inactif. Cette fonctionnalité permet de travailler avec le même fichier de données, quelle que soit la configuration du SIF à un instant donné.

1.1 Fenêtre Entrée

Un partitionnement peut être défini à l'aide d'une grille, régulière ou irrégulière, comportant un nombre fixé de sous-ensembles flous (SEF) de forme triangulaire. Cette grille a les propriétés d'une partition floue forte. Il peut aussi être construit pas à pas. Quel que soit le mode choisi, chacun des SEF peut être édité et chacun de ses paramètres modifiés : forme et points de rupture. Les formes disponibles sont le triangle, le trapèze, les demi-trapèzes inférieur et supérieur, la gaussienne, la gaussienne généralisée, le rectangle et enfin le type discret qui correspond à une valeur nette.

Remarque : Par souci de cohérence, si un fichier de données est ouvert, son nombre de colonnes doit être au moins égal au nombre d'entrées déclaré dans le SIF.

Nouvelle Entrée - Supprimer une entrée

Suite à la remarque précédente, une fois un fichier de données ouvert, on ne peut pas déclarer dans le SIF de nouvelle entrée, ni en supprimer. Pour le faire, il faut d'abord fermer le fichier de données.

1.2 Fenêtre Sortie

Une sortie a tous les paramètres d'une entrée, plus des paramètres spécifiques. Il faut préciser sa nature, nette, floue ou floue implicative, si elle est de type classification ou non, et enfin les opérateurs d'agrégation et de défuzzification utilisés.

Nouvelle Sortie - Supprimer une sortie

Une fois un fichier de données ouvert, on ne peut pas déclarer dans le SIF de nouvelle sortie ni en supprimer. Pour le faire, il faut d'abord fermer le fichier de données.

En fonction de la nature de la sortie, différents opérateurs d'agrégation, la somme et le maximum pour les règles conjonctives, sont disponibles (voir partie II, section 1).

Lorsque la sortie est floue et conjonctive, les opérateurs de défuzzification possibles sont *aires, moyenne des max et sugeno* La méthode des aires est analogue à celle du centre de gravité, la différence réside dans le fait que les aires communes à deux SEF sont comptabilisées deux fois. L'opérateur moyenne des max retourne le milieu du segment de l'alpha-coupe correspondant au degré de vérité le plus élevé, en cas d'ambiguïté sur le maximum un message est adressé à l'utilisateur. Enfin, l'opérateur de Sugeno est également proposé : la conclusion de chaque

règle est, dans ce cas, le milieu du noyau de l'ensemble flou correspondant, et la sortie est ensuite calculée comme la somme pondérée de ces conclusions, le poids étant le degré de vérité de la règle.

Lorsque la sortie est nette, deux opérateurs de défuzzification peuvent être utilisés : MaxCrisp, analogue au Mean-Max mais adapté pour des scalaires, et Sugeno. Pour des raisons d'interprétabilité, nous avons dans ce cas limité la sortie à une constante au lieu de la traditionnelle combinaison linéaire des entrées. La sortie est la somme pondérée des conclusions des règles, le poids étant le degré de vérité de la règle.

Remarque sur la défuzzification des sorties floues

Pour tous les opérateurs de défuzzification des sorties floues, une correction est apportée à la borne supérieure (resp. inférieure) d'un demi-trapèze supérieur (resp. inférieur), situé à l'extrémité de la partition floue. Cela permet d'inférer les valeurs extrêmes (bornes du domaine de la sortie). Cette correction est automatique lors de la construction d'une partition complète : grille régulière ou irrégulière, ou génération d'un SIF sans règles. A chaque modification manuelle de ces bornes, la correction est déclenchée par le changement d'opérateur de défuzzification.
Cette correction peut aussi être appliquée à la réouverture d'un fichier de configuration SIF (dans ce cas, cocher l'option *Garantir l'inférence des valeurs extrêmes des sorties*).

Paramètre seuil d'alarme

Le paramètre *seuil* est un seuil de tolérance lié au déclenchement d'un message d'avertissement lors de la défuzzification (voir paragraphe alarme dans la section 2).

Option classification

— Pour une sortie nette, l'option classification ramène la valeur inférée à la classe la plus proche. Les classes sont déterminées à partir de la colonne du fichier de données correspondant à cette sortie, si si un fichier de données est ouvert et que la colonne existe. Dans le cas contraire, les classes sont calculées à partir des conclusions des règles.

— Pour une sortie floue, l'option classification ne change rien à la valeur inférée, qui est la valeur défuzzifiée, mais le fichier résultat contiendra, en plus de la valeur inférée, le degré d'appartenance de la valeur inférée à chacun des SEF de la sortie.

Remarque : SEF discret

Dans une sortie floue, les SEF discrets ne sont pas compatibles avec la défuzzification par la méthode des aires.

Règles implicatives

Le choix d'utiliser des règles implicatives plutôt que conjonctives s'effectue

par la paramétrisation de la sortie. En effet, les partitions des variables d'entrée, comme la méthode de calcul du degré de vérité d'une règle pour des valeurs d'entrée sont indépendantes du type de règles.

Pour choisir ce type de règles il faut cocher la case *Impli*. Dans ce cas, le seul opérateur de défuzzification possible est appelé *Impli* et le choix de l'opérateur d'implication se fait au moyen de la méthode d'agrégation des règles, soit le champ *Disjonction*. Trois sont disponibles : *Resher-Gaines*, *Goguen* et *Gödel*. Les distributions de possibilités inférées par ces trois opérateurs diffèrent seulement par les supports, elles ont le même noyau.

La méthode d'agrégation spécifique des règles implicatives impose un type de partition de sortie particulier. Nous conseillons d'utiliser les *partitions quasi-fortes* (PQF) dérivées des *partitions floues fortes* (PFF) construites par défaut par les options *Grille régulière* ou *Grille irrégulière*. La PQF qui correspond à la PFF existante sera proposée lors de l'action sur la case à cocher *Impli*.

Le choix des règles implicatives interdit l'usage d'une sortie *Nette*.
L'option *Grille régulière* construit une partition adaptée aux règles implicatives, i.e. une partition PQF.
Lors du passage d'une sortie implicative à une sortie conjonctive, obtenu en décochant la case *impli*, la modification de la PQF existante en PFF équivalente est systématiquement proposée, et inversement.

L'option *Classification* est sans effet sur les règles implicatives.

1.3 Fenêtre Règles

La fenêtre *Règles* permet de créer et d'afficher les règles d'inférence. Les labels proposés pour chacune des entrées correspondent aux SEF qui ont été définis. La chaîne vide indique que la variable est absente de la définition de la règle, cela signifie que cette valeur n'a pas d'importance pour cette règle, elle n'est pas prise en compte pour le calcul du degré de vérité. La colonne *active* permet d'activer ou de désactiver individuellement les règles. Une règle inactive n'intervient pas dans le calcul de l'inférence.

Attention : Seules les règles actives sont conservées à l'enregistrement du fichier de configuration du SIF.

Les règles, qui sont présentées sous forme de tableau (1 colonne par entrée ou sortie) peuvent être triées par colonne, en cliquant sur l'en-tête de la colonne.

Une option du menu *Affichage* permet d'afficher ou de cacher les variables inactives.

Une autre option du menu *Affichage* permet de pondérer les règles. Ces poids doivent être positifs et sont archivés dans le fichier de configuration du SIF, avec 3 décimales, si au moins l'un d'entre eux est différent de la valeur par défaut, 1. Ce poids expert modifie le résultat de l'inférence. En effet, il est utilisé pour multiplier

le degré de vérité de la règle dans les fonctions d'agrégation de règles, somme et max. Remarque : Pour les sorties floues, les poids supérieurs à 1 peuvent conduire à un effet de saturation assez rapidement. Pour les opérateurs de défuzzification des sorties floues, à l'exception de *SugenoFuzzy*, le poids fixe le niveau de l'alpha-coupe du SEF de sortie considéré. Il est donc inférieur à 1.

1.4 Option Inférer

L'option *Inférer* du menu SIF permet d'inférer les valeurs de sortie du SIF à partir de valeurs d'entrée saisies manuellement.

Remarque : Une autre option, *inférence à partir d'un fichier*, est disponible dans le menu *Données*.

Ici, le mécanisme d'inférence est illustré graphiquement, il permet de visualiser l'impact des variations de l'une ou l'autre des valeurs d'entrée sur chacune des règles ainsi que sur la sortie défuzzifiée. Chaque ligne représente une règle et chacune des colonnes correspond à une variable d'entrée ou de sortie. Le degré de vérité de chacune des règles est affiché, ainsi que la sortie du système correspondant aux valeurs choisies pour les entrées.

Le caractère pédagogique limite l'utilisation de cette fenêtre à des systèmes de taille réduite, en termes de nombre de variables et de nombre de règles.

Cette fenêtre étant implémentée sous forme de table, les règles peuvent être triées par l'action d'un simple click dans l'entête d'une colonne.

Comme les saisies sont dynamiques, les changements sont immédiatement propagés, l'inférence est effectuée à partir de la configuration affichée.

Deux mécanismes d'inférence

Deux mécanismes d'inférence sont maintenant implémentés dans Fispro. Celui utilisé pour les règles conjonctives s'appelle FITA (First Infer Then Aggregate) et permet d'inférer une valeur pour chacune des règles avant de les agréger. Il est également utilisé avec les règles implicatives lorsque les valeurs d'entrée sont précises. Dans le cas contraire, lorsqu'au moins l'une des valeurs d'entrée est imprécise, un mécanisme différent est nécessaire : FATI (First Aggregate Then Infer).

Pour une comparaison détaillée des deux types de règles, des mécanismes d'inférence et de l'implémentation FATI de Fispro, voir [12].

Lorsque l'algorithme FATI est utilisé, la sortie correspondant à chacune des règles est laissée vide, car elle ne peut pas être calculée.

Données floues

Le menu de la fenêtre d'inférence comprend une option : *Données floues*, elle-même subdivisée en deux.

— *Gabarit* : L'objectif est de transformer la valeur précise en nombre flou centré sur cette valeur. Deux paramètres sont nécessaires : la largeur du noyau et celle du support. La contrainte imposée est que la largeur du support soit supérieure ou égale à celle du noyau. Un gabarit peut être défini pour chacune des entrées du système. Le gabarit peut être enregistré et relu à partir d'un fichier d'extension *.tpl*.
Dans l'inférence, le curseur permet de faire varier la valeur centrale, et le nombre flou est construit à partir du gabarit et de cette valeur centrale. L'inférence manuelle avec des données floues est utilisable pour tous les types de règles, elle est limitée à 2 entrées et 1 seule sortie pour des règles implicatives. Cette option n'est pas encore disponible pour l'inférence à partir d'un fichier de données.
— *Alpha-cut* : Permet de spécifier le nombre d'alpha-coupes utilisé pour approximer la valeur d'entrée imprécise, dans le mécanisme d'inférence FATI. Ce nombre est commun à l'ensemble des entrées du système.

1.5 Réponse du système

Ce sous-menu permet de représenter la variation d'une sortie du SIF en fonction des variations d'une ou de deux variables d'entrée, les autres variables d'entrée étant fixées par l'utilisateur.

Remarque : les graphiques créés dans ce sous-menu ne tiennent pas compte du fichier de données éventuellement ouvert, mais sont construits uniquement en fonction des caractéristiques du SIF : partitions et règles. Les courbes ou surfaces sont donc construites à partir d'une grille d'interpolation sur la plage de données choisie.

L'option **Visualiser** du menu **Données** permet, elle, de visualiser en 2D ou en 3D les points d'un fichier de données.

1. **section**

   Graphique en deux dimensions (X,Y) qui représente une sortie, sur l'axe Y, en fonction d'une variable d'entrée, X.

   Choix des plages de variation

   Pour la variable X, il faut choisir les limites de représentation, min et max, ainsi que le nombre de points de calcul par axe.

   Lorsque le SIF ouvert comporte plus d'une entrée active, il est nécessaire de fixer les valeurs des autres variables. La case à cocher *Breakpoints* permet de limiter les valeurs aux points caractéristiques de la partition, comme les sommets et intersections des triangles. Pour fixer la valeur de chaque variable, il faut utiliser soit les flèches d'incrémentation (ou de décrémentation) situées à droite de la variable, soit les flèches haut et bas du pavé

numérique après avoir positionné le curseur de la souris dans la zone d'édition de la valeur, soit encore entrer directement la valeur.

Graphique 2D

Ce graphique s'affiche dans la partie inférieure de la fenêtre. Il peut être exporté dans un format graphique standard (Menu *fichier*).

Le graphique réagit aux diverses commandes de la fenêtre des paramètres. Par exemple en laissant le click gauche appuyé sur une des flèches d'incrémentation/décrémentation la courbe s'actualise en temps réel, permettant d'apprécier des évolutions de façon continue.

2. **surface**

   Graphique en trois dimensions (X,Y,Z) qui représente la surface de réponse d'une sortie, sur l'axe Z, en fonction de deux variables d'entrée, X et Y.

   Choix des plages de variation

   La partie supérieure de la fenêtre permet de fixer les paramètres de variation et de résolution pour chacune des variables X et Y. Voir le paragraphe correspondant dans l'option précédente.

   Graphique 3D

   Ce graphique s'affiche dans la partie inférieure de la fenêtre. Il peut être exporté dans un format graphique standard.

   L'option *Exporter* du menu *fichier* propose une fonction d'exportation du graphique 3D, l'option *Capturer* permet un enregistrement des mouvements du graphique en fonction d'une plage de paramètres de calculs de la surface. L'enregistrement, stocké sous la forme d'une suite d'images jpeg indexées, peut être transformé en film. Voici un exemple de commande, sur (GNU)Linux, pour produire un document avi, également exploitable par windows :

   mencoder mf ://*.jpg -mf w=800 :h=600 :fps=25 :type=jpg -ovc lavc -lavcopts vcodec=mpeg4 -oac copy -o output.avi

   Signification des couleurs de la surface

   La couleur de la surface varie de façon continue entre le rouge (minimum) et le violet (maximum), suivant l'ordre de l'arc-en-ciel en fonction de la valeur du point considéré. Attention : les couleurs n'ont de sens qu'au sein d'une fenêtre. Le rouge correspond au minimum des valeurs inférées dans cette fenêtre.

   Actions de la souris

   Le graphique réagit aux diverses commandes de la fenêtre des paramètres. Par exemple en laissant le click gauche appuyé sur une des flèches d'incrémentation/décrémentation la surface s'actualise en temps réel, permettant d'apprécier des évolutions de façon continue.

Lorsque certains points de la zone n'activent aucune règle, la valeur de sortie est la valeur par défaut.

D'autre part, un click gauche dans la zone de graphique 3D indique les coordonnées (x, y, z) du point, un click droit indique les règles activées pour ce point au dessus d'un degré paramétrable par le menu *Options* (valeur par défaut 0.001) et un double click gauche ouvre, si un fichier SIF est ouvert, la fenêtre inférence présentée dans la section précédente, ce qui permet de visualiser les résultats de l'inférence pour ce point (valeurs non modifiables dans la fenêtre d'inférence).

Point de vue

Pour changer le point de vue depuis lequel on regarde le graphique, 3 actions sont possibles :

— Bouton gauche enfoncé + déplacements : rotation ;
— Bouton droit enfoncé + déplacements : translation ;
— Molette centrale + déplacements : zoom.

L'option *Réinitialiser Graphe* permet de réinitialiser le graphique 3D.

Coupe

Le menu *Section* propose une coupe de la surface de réponse suivant l'un des deux axes, x ou y. Une fenêtre popup permet de choisir la valeur de l'entrée selon laquelle on fait la coupe, et de la faire varier dynamiquement. Le plan correspondant à la coupe est affiché sur le graphique 3D, et suit les variations de la coupe.

1.6 Générer un SIF sans règles

Pour cette option, un fichier de données doit être ouvert.
Il est pris comme référence pour générer les partitions des variables.

Cette option permet de générer les parties Entrées et Sortie d'un SIF, en précisant le type de hiérarchie utilisé pour construire les partitions floues (description section 3.1.2 - Générer sommets), le nombre de sous-ensembles flous par variable et la tolérance sur les valeurs uniques ou bien le nombre de groupes pour les k-means.

Le SIF généré a autant de variables que le fichier de données a de colonnes.

Il a une seule sortie, qui correspond à la dernière colonne du fichier de données.

L'opérateur de conjonction est paramétrable. La sortie est soit nette, soit floue. Dans ce cas, les partitions sont construites de manière analogue à celles des entrées.

Le fichier de configuration du SIF sans règles peut être utilisé comme point d'entrée de l'une des procédures de génération de règles : les options *Générer*

règles et *Générer conclusions*, les arbres de décision flous, l'algorithme de Wang et Mendel ou bien l'ols.

Le SIF courant n'est pas écrasé, une nouvelle fenêtre est ouverte.

1.7 Générer règles

Cette option permet de générer l'ensemble des règles possibles correspondant à toutes les combinaisons possibles des entrées. Les règles sont archivées par ordre lexicographique. Il est nécessaire qu'au moins une entrée active soit définie, et que ses SEF soient construits.

Si, de plus, un fichier de données est déjà ouvert, alors seules les règles correspondant à un degré de vérité, cumulé pour l'ensemble des exemples, supérieur au seuil paramétré seront conservées. Dans ce cas, une option permet d'archiver les règles par ordre de poids cumulé décroissant, sinon leur ordre dépend de celui des exemples.

Les règles sont générées avec une conclusion par défaut, égale à 1 (sortie nette) ou au label du 1er SEF (sortie floue).

Le SIF courant n'est pas écrasé, une nouvelle fenêtre est ouverte.

1.8 Générer conclusions

Pour cette option, un fichier de données doit être ouvert.

Les conclusions des règles peuvent être calculées par la méthode des moindres carrés, pour minimiser la somme des carrés des erreurs entre les sorties observées et celles inférées par le modèle, ou bien générées par un algorithme inspiré de Fpa, qui est l'acronyme en anglais d'algorithme de prototypage rapide [6, 7]. Quelle que soit l'option choisie, il est ensuite possible de réduire le vocabulaire de sortie.

Fpa

La méthode Fpa consiste en un algorithme simple qui permet d'initialiser ou d'actualiser la conclusion des règles à partir d'un jeu de données d'apprentissage.

L'initialisation des conclusions se fait à partir des valeurs observées pour un ensemble d'exemples choisis, dont le choix est expliqué plus loin, E_r pour la règle *r*. 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. Il est formé d'exemples choisis en fonction de leur degré de vérité pour la règle.

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*, donné par la constante START_DEC). 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. La valeur du pas est fixée par la constante STEP_DEC, qui vaut actuellement *0.1*. 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 donnée dans le glossaire de la section V.

La seconde, dite *minimum*, consiste à considérer l'ensemble des exemples dont le degré de vérité pour la règle est supérieur à un seuil donné, *MuMin*. Cette stratégie, par l'utilisation d'un ensemble plus large, autorise l'emploi de procédures plus fines, telles qu'une descente de gradient. Dans ce cas, la valeur initiale peut être également déterminée suivant l'équation 1. La stratégie appliquée consiste à considérer l'ensemble des exemples dont le degré de vérité pour la règle est supérieur à un seuil donné, *seuil blanc*. 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.

OLS (Minimisation par les moindres carrés

L'algorithme optimise les conclusions des règles en minimisant la somme des carrés des erreurs (voir 3.2.3).

Réduction du vocabulaire de sortie

Cette option permet de réduire le nombre de conclusions distinctes des règles dans un SIF, et ainsi d'améliorer la lisibilité de la base de règles. Dans le cas d'une sortie nette de type régression, les valeurs des conclusions des règles sont distinctes les unes des autres : *a priori* il y en a autant que de règles a l'issue d'une procédure d'apprentissage.

Il faut indiquer à partir de quelles valeurs la réduction doit s'opérer. Deux choix sont possibles : soit les conclusions des règles existantes, soit les valeurs de la variable de sortie dans le fichier de données. La réduction consiste en une opération de clustering des conclusions ou des valeurs de sortie pour obtenir les valeurs finales des conclusions.

L'utilisateur peut fixer le nombre de valeurs désiré ou bien spécifier une perte de performance tolérée. En effet, la réduction de vocabulaire s'accompagne en général d'une perte de précision.

Lorsque le vocabulaire a été réduit, il est possible de "fuzzifier" la sortie, c'est à dire construire une partition floue forte à partir de ces valeurs en nombre réduit. Il suffit pour cela de transformer la sortie nette en sortie floue dans la fenêtre *Sortie*.

2 Menu Données

Dans la majorité des cas, nous souhaitons inférer des valeurs de sortie pour un ensemble de données. Le fichier de données doit comporter au moins autant de colonnes que ce qu'il y a de variables d'entrées (actives ou inactives) dans le système. Il peut également comporter, en plus, un nombre de colonnes correspondant au nombre de sorties, généralement une seule, de la configuration en cours.
Les colonnes sont séparées par un délimiteur détecté automatiquement. Les délimiteurs espace et tabulation ne sont pas autorisés.

L'ouverture du fichier de données, avec au moins une valeur de sortie observée, sera nécessaire pour activer les options du menu apprentissage.

2.1 Générer des échantillons

Ce sous-menu permet de générer des fichiers échantillons par tirage aléatoire à partir des données. Deux possibilités : génération de couples de fichier apprentis-

sage et test, ou genération de K fichiers. Notons N le nombre de lignes du fichier de données.

Dans le premier cas, chaque couple est composé d'un fichier d'apprentissage (A lignes) et de son complément par rapport au fichier de données (N-A lignes). On peut préciser le nombre de couples et la taille du fichier d'apprentissage par rapport à celle du fichier de données, soit A/N.

La génération crée autant de couples de fichiers que demandé, ils portent le nom du fichier de données, suivi de *lrn.sample.n* pour le 1er fichier du couple et de *tst.sample.n* pour le second, où n varie de 0 (1er couple) à N-1 (Nième couple).

Dans le second cas, le programme partage le fichier de données en K blocs, tous de taille égale à floor(N/K) si l'option taille constante est choisie, ou sinon de taille égale à floor(N/K) pour les K-1 premiers, et N-K*floor(N/K) pour le dernier.

Paramètres supplémentaires (valables dans les deux cas) : germe du tirage aléatoire et option classification.

le germe 0 correspond à un nouveau tirage à chaque fois, une autre valeur, 1 par exemple, permet de fixer le germe, et donc de reproduire un tirage donné.
La case à cocher *classif.* permet d'imposer que le tirage de l'échantillon respecte les proportions des classes dans une des variables du fichier de données, par défaut la dernière colonne. Dans ce cas, on peut préciser une valeur de tolérance numérique (par défaut 0.01), qui peut être utile dans le cas où les classes sont des valeurs approchées.

2.2 Visualiser

Le menu de visualisation offre 3 possibilités. Dans tous les cas, l'option *Exporter* du menu *Fichier* permet d'exporter le graphique dans différents formats (EPS, JPEG, etc.)

1- **histogramme**

Distribution de chacune des variables, conjointement avec son partitionnement flou, si un SIF est ouvert. Sous chacune des partitions sont affichés deux indices : le coefficient de partition, PC, à maximiser, et le coefficient d'entropie, PE, à minimiser.

Dans les formules suivantes, c est le nombre de SEF, n le nombre d'exemples du jeu de données et u_{ik} le degré d'appartenance de l'exemple k au SEF i.

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

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

2 - **graphe X-Y**

Il est possible de représenter le lien entre deux variables par l'intermédiaire d'un graphe X-Y, avec affichage optionnel de la droite de régression, ou de la bissectrice des axes.

Le click gauche permet d'activer ou de désactiver une donnée (lien dynamique avec la table présentée ci-après). En cas de difficulté graphique pour activer ou désactiver une donnée, il peut être plus facile de le faire à partir de la table de données (case à cocher).

Une info bulle indique le numéro d'ordre du point situé au voisinage de la souris.

3 - **graphe X-Y-Z**

Les points correspondants aux variables choisies sont affichés dans un graphe 3D.

Les mêmes fonctionnalités que dans le graphe X-Y sont disponibles en 3D, mais cette représentation offre des degrés de libertés supplémentaires par la manipulation de la souris :

— Bouton gauche enfoncé + déplacements : rotation du plot.
— Bouton droit enfoncé + déplacements : translation du plot.
— Molette centrale + déplacements : zoom.

L'option *Réinitialiser Graphe* du menu *Fichier* permet de réinitialiser le graphique 3D.

2.3 Table

Cette option permet d'afficher les données sous forme de tableau, de les trier par colonne et de désactiver certains exemples, pour ne pas les prendre en compte dans le calcul de l'inférence. Par défaut, l'état des exemples est conservé à la réouverture du fichier de données. La case à cocher *Activer tous les exemples* de la fenêtre d'ouverture de fichier permet de réactiver tous les exemples.

Un double click sur une ligne de la table ouvre la fenêtre inférence avec les valeurs de l'exemple.

2.4 Inférer

Cette option est une implémentation de la fonction *Performance*. Se reporter à la section 2 pour une description de l'ensemble des arguments et résultats. La case à cocher *Liens entre règles et données* permet de calculer les règles activées par chaque exemple, au-delà du seuil blanc choisi. Par souci de rapidité, cette case n'est cochée par défaut que si le fichier a moins de 200 lignes. Le résultat

de l'inférence, lorsqu'elle est effectuée à partir d'un fichier de données, est stocké

dans un fichier texte qui peut être lu par un tableur. Le format de ce fichier, en particulier son nombre de colonnes, dépend du type d'opérateur de défuzzification choisi pour la sortie étudiée. Il est décrit en détail dans la section 2 de la partie II de ce document.

Dans la fenêtre, il faut choisir la sortie sur laquelle on fait l'inférence. Si la sortie est présente dans le fichier de données, l'erreur est caractérisée par le nombre d'exemples mal classés dans le cas d'une sortie de type classification, et dans le cas d'une sortie continue par trois indices de performance, *PI*, *RMSE* et *MAE* définis dans la section (voir section V du glossaire). Un fichier résumé des performances est disponible, voir la section 2 pour son format.

Remarque : les procédures d'apprentissage sont toutes caractérisées par le *RMSE*.

Les fonctions d'affichage présentées dans la visualisation des données sont disponibles pour visualiser les résultats : histogrammes, graphes X-Y avec ou sans droite de régression. Dans le graphe X-Y, si l'option *Liens entre règles et données* a été choisie dans la fenêtre *Inférer*, des informations peuvent être obtenues sur chaque point en faisant glisser la souris sur le point : numéro de ligne de l'exemple dans le fichier de données, et numéros des règles activées (au-delà du seuil blanc choisi).

Pour la classification, le résultat n'est pas un graphe X-Y, mais une matrice dite *de confusion*. L'exemple de la figure 1 montre que les 50 exemples de la classe 1 sont bien classés par le système, 49 exemples de la classe 2 sont bien classés et 1 mal classé (classe inférée=classe 3), et 47 exemples de la classe 3 sont bien classés et 3 mal classés (classe inférée=classe 2).

Remarque : la classification correspond à une sortie nette, avec l'option classification cochée.

Sorties implicatives

Pour les sorties implicatives, le graphe X-Y présente la valeur inférée sous forme d'intervalle dont les bornes correspondent au minimum et maximum du noyau de la distribution de possibilité inférée.

Une option supplémentaire est disponible pour les sorties implicatives, elle permet de calculer, pour chaque exemple, le degré d'adéquation de la sortie inférée avec la sortie calculée. Cette option est inspirée de la *matrice de confusion* en classification, mais elle tient compte du fait que la sortie inférée est une distribution de possibilité, et non pas une valeur unique obtenue par défuzzification, comme c'est le cas dans les règles conjonctives.

Le degré d'adéquation varie entre 0 et 1. Il est calculé à partir de l'intersection de la distribution de possibilité inférée avec les SEF de sortie, et tient compte de

l'amplitude de cette distribution. Le résultat est cumulé et affiché sous forme de tableau, avec une ligne par SEF de sortie, une colonne *degré d'adéquation* et une colonne *exemple*.

Attention : quel que soit le type de système, implicatif ou conjonctif, l'indice de performance ne prend en compte que les exemples qui activent les règles au-delà du seuil blanc choisi (0.1 **par défaut).**

		observé		
		classe 1	classe 2	classe 3
inféré	classe 1	50	0	0
	classe 2	0	49	3
	classe 3	0	1	47
non classé		0	0	0

FIGURE 1 – Exemple de matrice de confusion en classification

2.5 Liens

Cet utilitaire permet de visualiser les liens qui existent entre les exemples et les règles d'une part, les règles entre elles d'autre part.

Il crée plusieurs fichiers de travail, dont le nom est préfixé par défaut par celui du fichier de données.

— rules.items
  Ce fichier compte autant de lignes que de règles, plus 2.
  * Première ligne : nombre de règles
  * Deuxième ligne : nombre maximal d'exemples qui activent une règle
  * Troisième ligne : description correspondant à la première règle, soit dans l'ordre : numéro de la règle (1 pour la première), poids cumulé, nombre d'exemples qui activent la règle (au-delà d'un seuil paramétré, 1e-6 par défaut) suivi de leurs numéros.

— items.rules
  Ce fichier compte autant de lignes que d'exemples.
  Pour chaque ligne : Numéro de l'exemple (1 pour le premier), puis les numéros des règles qu'il active.

— rules.links
  La notion de liens entre règles est utile pour apprécier la cohérence d'une base de règles. Si deux règles sont fortement liées, et que leurs conclusions sont différentes, alors l'incohérence provient du fait que les domaines d'entrée couverts par les règles ne sont pas suffisamment spécifiques. Cette

situation peut aussi correspondre à la présence d'une exception dans la base d'apprentissage. L'expression mathématique du niveau de lien de la règle *i* avec la règle *j* est :

$$L_{i,j} = \frac{N_{i,j}}{N_i}$$

N_i représente le cardinal de l'ensemble E_i des exemples qui activent la règle *i*, $N_{i,j}$ est le cardinal de $E_i \cap E_j$.

Ce fichier se présente sous la forme d'une matrice carrée de la taille du nombre de règles. Chaque cellule indique le niveau de lien correspondant. Il convient de noter que cette matrice n'est pas symétrique.

— rules.sorted
Si l'option correspondante est activée, les règles sont triées suivant leur poids cumulé, c'est-à-dire leur influence dans la base d'exemples.
Remarque : les liens ne dépendent pas des sorties du système.

2.6 Distance

Les partitions floues peuvent être utilisées pour générer une fonction de distance entre individus,appelée FP-distance. Cela permet d'introduire de la connaissance experte dans le calcul des distances. Le programme génère la matrice (n x n) des distances entre les (n) exemples du fichier de données ouvert. Par défaut, la distance est normalisée dans chaque dimension entre 0 et 1. Il est possible de combiner FP-distance et distance euclidienne dans le cas multi-varié. L'agrégation des distances partielles est réalisé suivant la formule de Minkowski, la valeur de l'exposant p est un paramètre. Cette matrice peut être ensuite exportée pour être manipulée par des algorithmes de clustering, par exemple dans R.

La distorsion en une dimension de la distance euclidienne par la distance paramétrée est visualisée sous forme d'image en couleur de leurs différences.

La distance est présentée en détail dans :

Serge Guillaume, Brigitte Charnomordic, et Patrice Loisel.

Fuzzy partitions : a way to integrate expert knowledge into distance calculations. International Journal of Information Sciences, page In Press, 2012.

3 Menu Apprentissage

Ce menu propose des procédures d'apprentissage supervisé, de simplification et d'optimisation.

Attention : toutes les procédures d'apprentissage sont compatibles avec les systèmes de règles implicatives, mais, dans ce cas, les résultats doivent être interprétés avec la sémantique de ces règles.

L'apprentissage permet de prendre en compte des données, sous la forme d'un fichier appelé ensemble d'apprentissage, pour concevoir ou optimiser certaines parties, voire la totalité, du système. Les principales méthodes d'apprentissage sont présentées dans [8]. Pour garantir l'interprétabilité, nous avons choisi de séparer l'étape de conception des partitions de celle de la génération des règles, même lorsque les deux sont confondues dans les publications originales.

3.1 Partitions

3.1.1 Générer un SIF sans règle

Une option est disponible pour *Générer un SIF sans règle*, SIF qui pourra être utilisé en entrée des méthodes d'induction. Cette procédure est decrite dans le menu SIF (voir 1.6).

Un menu, appelé *HFPSEF*, permet de générer une famille de partitions dans chaque dimension. Cet algorithme ne travaille qu'à partir des données, et si un SIF est ouvert, il n'en tient pas compte.

3.1.2 Menu HFP SEF et HFP SIF

La méthode Hfp, qui signifie Hierarchical Fuzzy Partitionning, est décrite en détail dans [9, 10]. Son objectif est de créer une famille de partitions floues par entrée, puis de sélectionner la meilleure combinaison de ces partitions pour créer le SIF correspondant.

Nous présentons simplement ici différentes options proposées dans le menu. Seul un fichier de données doit être ouvert. Si un SIF est ouvert, le programme n'en tient pas compte. **La sortie est la dernière colonne du fichier de données.**

Pour utiliser cette méthode, les opérations sont à faire dans un certain ordre, car elles créent des fichiers intermédiaires :

Générer un fichier de configuration pour HFP
Générer les sommets
Sélectionner une partition
Examiner le fichier créé par l'étape précédente (result.min), pour déterminer le meilleur nombre de SEF pour chaque entrée
Générer un SIF (Apprentissage-Induction de règles-HFP SIF)

Les options sont détaillées ci-dessous :

1. Générer un fichier de configuration pour HFP

   Pour chaque variable (colonne du fichier de données), l'utilisateur peut modifier la taille maximum de la partition (valeur par défaut=7). La dernière colonne est par défaut laissée de côté (considérée comme une sortie) pour la construction des partitions. La conjonction des prémisses est paramétrable.

   Le paramètre le plus important de cette étape est le type de hiérarchie à générer. 3 choix sont possibles :

   (a) grille régulière

   (b) k-means

   La seconde option proposée est d'utiliser une méthode de groupage éprouvée, les k-means. Il n'y a, a priori, aucune relation entre les centres des différentes partitions d'une même entrée.

   (c) hfp

   Cette dernière possibilité est une procédure ascendante. A chaque étape, au sein d'une entrée, deux sous-ensembles flous sont fusionnés. Le temps de calcul peut être important. Il dépend principalement, du nombre de sous-ensembles flous dans la partition initiale. Celle-ci peut être constituée de deux manières. La première consiste à regrouper les valeurs de l'ensemble d'apprentissage en considérant que deux d'entre elles sont identiques si leur différence est plus petite que le seuil fixé, en pourcentage du domaine de variation. Le nombre de groupes retenu, chacun correspondant à un sous-ensemble flou, dépend directement de ce seuil. La seconde possibilité, est de fixer le nombre de groupes dont les centres seront calculés par les k-means.

2. Générer les sommets

   Cette option crée un fichier qui comprend l'ensemble des coordonnées des centres pour toutes les partitions, de deux au nombre maximal fixé de sous-ensembles flous (7 par défaut), et ce, pour chacune des variables choisies lors de l'étape précédente, conformément au fichier généré lors de l'étape précédente.

3. Visualiser les sommets

   Cette option permet de visualiser la position des sommets, en regard de l'histogramme des données. Sous chacune des partitions sont affichés deux indices : le coefficient de partition, PC, à maximiser, et le coefficient d'entropie, PE, à minimiser.

   Dans les formules suivantes, c est le nombre de SEF, n le nombre d'exemples du jeu de données et u_{ik} le degré d'appartenance de l'exemple k au SEF i.

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

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

4. Sélectionner une partition

   Les hiérarchies mono dimensionnelles calculées à l'étape précédente sont utilisées pour construire des systèmes d'inférence floue qui sont à la fois compacts et performants.

   Parmi les paramètres figure le nombre initial de sous-ensembles flous, 1 ou 2. Choisir 1, revient, de fait, à introduire progressivement les variables dans le système. Pour chacune des combinaisons, c'est-à-dire un nombre de SEF par entrée, le système, formé de l'ensemble des règles possibles, est caractérisé par des indicateurs de performance et de couverture. A chaque itération, un nouveau SEF est ajouté sur une et une seule entrée suivant le critère d'amélioration de le performance.

   L'autre paramètre important est celui de la méthode d'induction de règles. Deux sont disponibles : l'algorithme de Wang et Mendel (pas de paramètres supplémentaires) et la méthode FPA. Dans ce cas, les autres paramètres sont ceux de *Générer conclusions* (voir section 1.8) pour initialiser la conclusion des règles. Enfin, il faut spécifier le nombre maximum d'itérations et le nom du fichier sommets à utiliser.

   La performance peut être calculée sur un fichier de validation différent du fichier de données (qui a servi pour initialiser la conclusion des règles) : données actives ou inactives gérées par l'option table du menu Données, ou autre fichier.

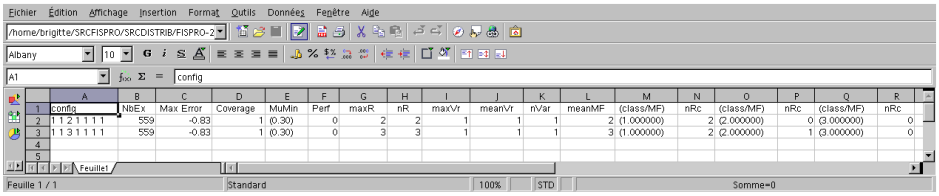
   Fichiers résultats :

   Le résultat de chaque itération est stocké, sous la forme d'une ligne dans les fichiers "result" et "result.min". Le fichier "result" contient l'ensemble des tentatives (ajout d'un sef sur chacune des variables d'entrée), le fichier "result.min" indique la configuration retenue après chacune des itérations.

   Format : les colonnes sont séparées par le délimiteur &, qui permet de relire le fichier dans un tableur, ou d'introduire directement le tableau dans un document LaTeX.

   1ère colonne : p (nombre de variables d'entrée) entiers, séparés par un blanc, qui indiquent le nombre de SEF pour chaque variable.

   2ème colonne : nombre d'exemples dans le fichier de validation



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	config	NbEx	Max Error	Coverage	MuMin	Perf	maxR	nR	maxVr	meanVr	nVar	meanMF	(class/MF)	nRc	(class/MF)	nRc	(class/MF)	nRc
2	1 1 2 1 1 1 1	559	-0.83	1	(0.30)	0	2	2	1	1	1	2	(1.000000)	2	(2.000000)	0	(3.000000)	0
3	1 1 3 1 1 1 1	559	-0.83	1	(0.30)	0	3	3	1	1	1	3	(1.000000)	2	(2.000000)	1	(3.000000)	0
4																		
5																		

FIGURE 2 – Exemple de fichier résultat de hfp

3ème colonne : erreur max

4ème colonne : indice de couverture

5ème colonne : seuil utilisé pour le calcul de l'indice de couverture

6ème colonne : indice de performance

colonnes suivantes : les caractéristiques de la base de règles

Les indices de performance et de couverture sont décrits dans la section 2, les caractéristiques de la base de règles dans la section 3 de la partie II de ce document.

L'analyse de ce fichier permet de choisir le ou les SIF que l'on souhaite générer. Les éléments à prendre en compte sont la performance (colonne 6) associée au niveau de couverture (colonne 4), mais aussi la complexité de chacun des systèmes mesurée par le nombre de règles et les caractéristiques de la base de règles données par la structure *InfoRB* (voir partie II, section 3) .

La performance, qui est une mesure d'erreur doit être la plus petite possible (minimum 0) avec un indice de couverture le plus grand possible (maximum 1) tandis que le nombre de règles ainsi que le nombre de SEF doivent être raisonnablement petits. Le compromis entre la complexité et la performance est laissé à la responsabilité de l'utilisateur.

Remarque : Cette option ne génère pas directement de SIF.

5. Générer un SIF

   Cette option, disponible dans le sous-menu Induction de règles-Option HFP SIF, permet de créer un fichier de configuration du SIF correspondant à une combinaison donnée. Le choix de la combinaison est fait par l'utilisateur, après examen des résultats de l'étape de sélection.

   Il faut préciser le fichier de sommets à utiliser, et le nombre de SEF souhaité pour chacune des entrées et la méthode de génération de règles : Wang et Mendel ou bien Fpa (et, dans ce cas, les paramètres qui lui sont nécessaires). Le SIF ainsi créé pourra ensuite être utilisé tel quel, ou bien en entrée du menu simplification.

3.2 Génération des règles

Plusieurs méthodes de génération de règles sont disponibles. La plupart travaillent à partir d'un SIF, l'algorithme "FPA", l'algorithme de "Wang & Mendel" et les "Arbres de décision flous". Dans ce cas les éventuelles règles existantes sont ignorées, et les SEF des partitions ne sont pas modifiés. L'OLS peut travailler à partir d'un SIF, la structure des partitions des entrées est conservée, ou bien seulement à partir du fichier de données, dans ce cas une étape supplémentaire permet la génération de partitions.

La simplification peut s'appliquer à tout SIF, indépendamment de la méthode de génération utilisée pour le construire. Elle ne modifie pas les SEF des partitions, mais seulement les règles.

L'optimisation peut s'appliquer à tout SIF, indépendamment de la méthode de génération utilisée pour le construire. Elle peut modifier tous ses éléments, y compris les SEF des partitions.

Nous indiquerons pour chacune des rubriques quels fichiers, celui de configuration d'un système (SIF) ou/et celui d'un jeu de données, doivent être ouverts. Les options de ce menu sont grisées si les fichiers indispensables ne sont pas ouverts.

Remarque : la plupart des méthodes d'apprentissage fonctionnent avec un seuil de tolérance EPSILON=10^{-6}. Avant d'utiliser ces méthodes, il est préférable de normaliser les données entre 0 et 1, si leur étendue est très grande ou au contraire très faible.

Remarque : l'indice de performance pour les sorties continues est le *RMSE* pour toutes les procédures d'apprentissage.

3.2.1 Menu FPA

Pour cette méthode, un SIF et un fichier de données doivent être ouverts. L'option *Générer un SIF sans règles* permet de construire automatiquement les partitions à partir des données (voir 1.6).

L'option *FPA* permet d'effectuer en une seule passe la génération des règles et de leurs conclusions (voir 1.7 et 1.8).

3.2.2 Menu Wang & Mendel

Pour cette méthode, un SIF et un fichier de données doivent être ouverts. Contrairement à l'algorithme original de Wang & Mendel [14], les partitions

doivent être définies avant l'application de la procédure. Elles peuvent être construites automatiquement par exemple par l'option *Générer un SIF sans règles*.

Cette procédure gère l'ensemble des sorties du SIF, et ignore les règles existantes.

Elle commence par initialiser une règle à partir de chacun des exemples de l'ensemble des données.

Celle qui correspond à l'exemple i s'écrit :

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

Les sous-ensembles flous A_j^i sont ceux pour lesquels le degré d'appartenance de x_j^i est maximum pour chacune des variables d'entrée j pour l'exemple i. Le sous-ensemble flou C_i est celui pour lequel le degré d'appartenance de la sortie observée, y_i, est maximum.

Si la sortie est nette, une conversion interne à la procédure crée des SEF centrés sur les valeurs de sortie, et retransforme ensuite les conclusions des règles en valeurs égales aux centres des SEF.

Un degré est assigné à chacune des règles. Il est calculé comme le poids de la règle pour l'exemple. Lorsque deux règles sont générées avec les mêmes prémisses, seule celle dont le degré est le plus fort est conservée.

Cette option est très simple d'emploi, elle ne requiert aucun paramètre.

Limite : Dans le cas de sorties nettes le nombre de valeurs distinctes est limité à MAX_CLASS, soit 15 actuellement. Un paramètre, *Pas de limite*, permet de repousser celle-ci à MAX_MF, soit 999. Si ce paramètre est activé et que le nombre de valeurs distinctes pour la sortie dans l'ensemble d'apprentissage est supérieur, alors une procédure de groupage, les kmeans, est appliquée pour trouver MAX_MF centres.

3.2.3 Menu Ols

Cette méthode peut travailler à partir du seul fichier de données, mais elle offre des options différentes si un SIF est également ouvert.

L'algorithme OLS [2, 3, 11] transforme chaque exemple d'un jeu de données en une règle floue et sélectionne ensuite les plus importantes d'entre elles au sens des moindres carrés, par régression linéaire et orthogonalisation de Gram-Schmidt. Une fois la sélection faite, un deuxième passage de l'algorithme est réalisé au cours duquel les conclusions des règles sont optimisées au sens des moindres carrés.

Notre implémentation est motivée et décrite en détail dans [5].

Comme pour toutes les méthodes d'apprentissage de Fispro, la démarche est découpée en deux étapes distinctes : la génération des partitions si nécessaire, puis celle des règles.

Génération des partitions

La première étape consiste en la génération des partitions si aucun SIF n'est ouvert. Dans ce cas, les partitions sont générées à partir des données. L'interface propose la génération de partitions floues fortes (comme dans l'option du menu **SIF**"Générer un SIF sans règle", section 1.6)).

Elle propose également l'algorithme OLS original. Il consiste à générer une fonction d'appartenance gaussienne pour chaque valeur de la distribution, ce nombre est ensuite limité à MAX_MF (999 actuellement) par regroupement. Dans ce cas, les partitions générées ne sont pas des partitions floues fortes.

Si un fichier SIF est ouvert, les partitions des entrées sont conservées.

Génération des règles

Le seconde étape est celle de la sélection des règles (phase 1), puis de l'optimisation de leurs conclusions (phase 2). L'algorithme travaille en fonction d'une sortie (Défaut : la première sortie, ou la dernière colonne du fichier de données). Le nombre de règles sélectionnées au cours de la phase 1 est paramétré par deux critères concomitants :

— Part de l'inertie de la sortie non expliquée par l'ensemble des règles (Défaut : 0.1).

— Nombre de règles. Défaut : 100.

L'algorithme s'arrête dès que l'un de ces deux critères est satisfait. Avec les valeurs par défaut, c'est en général le premier qui est d'abord satisfait.

La phase 2 de l'algorithme OLS optimise suivant le critère des moindres carrés les conclusions des règles sélectionnées. Dans le cas où un SIF est ouvert, une option permet de conserver les règles existantes. Dans ce cas, seule la phase d'optimisation est appliquée.

L'optimisation, qui n'affecte ni les partitions ni les prémisses des règles, mais seulement les conclusions de celles-ci, est également disponible depuis le menu *Générer conclusions*, section 1.8.

Réduction du vocabulaire de sortie

Enfin, comme les valeurs des conclusions des règles sont distinctes les unes des autres (il y en a autant que de règles !), cette option permet de réduire leur nombre. Il faut tout d'abord indiquer à partir de quelles valeurs la réduction doit s'opérer. Deux choix sont possibles : soit les conclusions des règles existantes, soit les valeurs de la variable de sortie dans le fichier de données. L'utilisateur peut fixer le nombre de valeurs désiré ou bien spécifier une perte de performance tolérée. En effet, cette réduction s'accompagne en général d'une perte de précision. Lorsque le vocabulaire a été réduit, il est possible de "fuzzifier" la sortie, c'est à dire construire une partition floue forte à partir des valeurs en nombre réduit.

L'option "Réduction du vocabulaire de sortie" est également disponible depuis le menu "Générer Conclusions".

3.2.4 Menu Arbre

Les arbres de décision flous sont une extension des arbres de décision classiques [1, 13]. Ils sont composés d'une racine, qui est le sommet ou point de départ de l'arbre, et de nœuds. Les nœuds terminaux sont appelés feuilles de l'arbre. Chaque nœud correspond à un sous-ensemble de valeurs d'une variable d'entrée (variable explicative) du problème traité. Ces éléments sont déterminés de façon à avoir une homogénéité maximale des exemples appartenant au nœud, par rapport à la variable à expliquer (variable de sortie). Cette homogénéité se traduit par une maximisation de l'entropie. Les chemins allant de la racine aux feuilles peuvent être interprétés de façon naturelle comme des règles de décision, strictes ou floues selon la nature de l'arbre.

Un élagage de l'arbre peut être effectué, en transformant un nœud en feuille, si la perte de performance qui en découle est faible. Cette procédure facilite l'interprétation. L'élagage est basé sur la performance, à l'inférence, du SIF équivalent à l'arbre.

Remarque : il est possible que l'arbre élagué ait une meilleure performance que l'arbre complet. En effet, la construction de l'arbre complet est basée sur la réduction de l'entropie, et non pas directement sur la performance.

Les arbres de décision flous proposés dans FisPro sont basés sur une implémentation floue de l'algorithme ID3 [15].

Génération
Pour générer un arbre de décision flou avec FisPro, un SIF et un fichier de données doivent être ouverts. La construction de l'arbre se fait par apprentissage sur une seule sortie, même si le SIF en compte plusieurs. Cette sortie est choisie par l'utilisateur.

Pour construire automatiquement le SIF, on peut utiliser l'option *Générer un SIF sans règles* du menu *Arbre* (voir 1.6).

Type de la sortie
Quatre cas sont possibles :
— sortie floue, avec l'option classification.
— sortie floue, sans l'option classification
— sortie nette, avec l'option classification

— sortie nette, sans l'option classification (arbre de régression).

Les 3 premiers cas utilisent le critère d'entropie floue pour construire l'arbre, le dernier utilise le critère de déviance, basé sur la dispersion des valeurs de sortie dans chaque nœud, mieux adapté à un problème de régression.

Quand la sortie est nette, avec l'option classification, les classes sont construites à partir des données, et des SEF discrets correspondant aux classes sont affectés automatiquement à la sortie. Le nombre maximal de classes ou de SEF en sortie est de 100.

Avec l'option classification, la classe majoritaire est associée à chaque nœud, sinon, la moyenne des observations arrivant au nœud lui est affectée.

Dans le cas d'une sortie floue, le résumé de l'arbre indique les proportions floues des observations arrivant au nœud pour chaque SEF.

Règles

Si le SIF possède des règles, elles sont ignorées.

Options

La fenêtre *Génère arbre* permet de choisir :

— le nom du fichier qui contiendra l'arbre créé, sous une forme adaptée à la visualisation (extension *.tree*).

— la profondeur de l'arbre (nombre maximum de niveaux), par défaut égale au nombre de variables.

— le seuil d'appartenance minimum pour qu'un exemple soit considéré comme appartenant au nœud

— le nombre d'exemples minimum, avec un niveau d'appartenance supérieur au seuil, pour envisager de développer le nœud

— la tolérance sur le degré d'appartenance à la classe majoritaire (classification seulement)

— le gain minimum (valeur relative) en entropie/déviance requis pour construire une branche

— en classification seulement : le choix du critère de construction de l'arbre : gain absolu d'entropie (valeur par défaut), ou gain relatif d'entropie.

— en régression seulement : la possibilité d'optimiser les conclusions des règles correspondant aux feuilles, dans une seconde passe, avec la technique déjà décrite d'optimisation par les moindres carrés (voir option OLS). L'option *gain relatif d'entropie* favorise les variables, pour lesquelles la répartition des exemples (et non des classes) est inégale entre les différents SEF. Le gain relatif favorise aussi les variables avec un petit nombre de SEF.

— l'option élagage :

L'élagage consiste en la suppression récursive des nœuds, depuis le bas de l'arbre en remontant vers la racine, si cette suppression ne diminue pas, ou diminue peu, la performance du SIF équivalent à l'arbre.
La perte relative de performance autorisée (par rapport à la performance de l'arbre avant élagage) est par défaut de 0.1. Elle est modifiable par l'utilisateur. La performance peut être calculée sur un fichier de validation différent du fichier de données : données actives ou inactives gérées par l'option table du menu Données, ou autre fichier.
L'élagage peut se faire par branches entières, en supprimant tous les nœuds fils d'un certain nœud (valeur par défaut), ou feuille par feuille.
Pour élaguer, on peut choisir un fichier différent de celui utilisé pour construire l'arbre.

— l'affichage des résultats intermédiaires

Résultats

L'application de la procédure déclenche la création d'un arbre, ou de deux arbres, si l'option élagage a été choisie, ainsi que :

— l'ouverture d'une fenêtre de visualisation graphique de chaque arbre créé. L'utilisateur peut choisir d'afficher différentes informations pour tous les nœuds, ou seulement pour les feuilles de l'arbre : nombre d'exemples attirés par le nœud, entropie/déviance du nœud, classe majoritaire et répartition d'effectif par classe (en classification), ou bien moyenne et écart type (en régression). La partie de l'arbre visible dans la fenêtre peut être exportée ou imprimée.
Le placement initial de l'arbre dans la fenêtre est automatique, mais l'utilisateur peut aussi sélectionner des branches de l'arbre et les déplacer manuellement pour améliorer leur placement. L'échelle et la police de caractères utilisées sont modifiables.

— l'ouverture d'une fenêtre SIF correspondant au système d'inférence floue équivalent à chaque arbre créé. Ce SIF peut ensuite être manipulé et sauvegardé.

— la création d'un fichier résumant la procédure, nommé result.fistree (figure 3) :

Chacune des lignes décrit un arbre flou (complet ou élagué) : la première colonne indique le nom du fichier de configuration correspondant à l'arbre. On a ensuite les indices décrits dans la section 2 :

2ème colonne : indice de performance

3ème colonne : indice de couverture

4ème colonne : erreur max

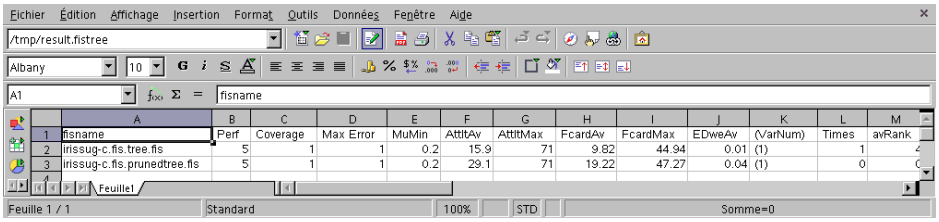


FIGURE 3 – Fichier result.fistree

5ème colonne : seuil d'appartenance utilisé pour le calcul de l'indice de couverture

Suivent les indices spécifiques a l'arbre flou :

6ème colonne : nombre moyen d'exemples attirés par une feuille

7ème colonne : nombre maximum d'exemples attirés par une feuille

8ème colonne : cardinalité floue moyenne des feuilles

9ème colonne : cardinalité floue maximum des feuilles

10ème colonne : entropie ou déviance moyenne des feuilles, pondérée par leur cardinalité floue

groupes de 3 colonnes (en nombre égal au nombre de variables d'entrée actives du SIF)

- numéro de la variable
- nombre de fois où elle est choisie comme variable discriminante
- rang moyen d'apparition

colonnes suivantes : les caractéristiques de la base de règles (voir la section 3).

Visualisation 2 options sont possibles pour visualiser un arbre créé par Fispro.

Graphe : comme décrit plus haut

Table : visualisation non graphique de l'arbre, avec les mêmes informations sous forme de tableau.

3.2.5 HFP SIF

Cette option est décrite dans la section 3.1.2.

3.3 Simplification

Deux fichiers doivent être ouverts : celui de configuration du SIF et celui des données. La simplification d'un système d'inférence floue, présentée dans [9], vise à éliminer des règles les variables les moins utiles.

Cette procédure est applicable à n'importe quel système d'inférence floue, spécialement ceux construits par apprentissage.

Le principe consiste à essayer de supprimer des variables dans la définition des règles ainsi que des règles entières. Ces modifications sont acceptées si les indices de performance et de couverture, présentés dans la section 2, restent dans des limites acceptables.

La procédure est réalisée à plusieurs niveaux. Le plus important, effectué dans tous les cas, est celui de la fusion d'un groupe de règles proches au sens de leurs prémisses. Deux autres niveaux sont optionnels : la suppression des règles, et la suppression de variables au sein de chacune des règles.

Paramètres principaux

La simplification s'accompagne généralement d'une perte de performance, celle-ci peut être relative, exprimée en pourcentage de la valeur initiale, ou bien absolue, dans ce cas l'utilisateur indique la limite.

Au cours de l'étape de fusion d'un groupe de règles, la couverture ne peut qu'augmenter. Il n'en n'est pas de même lors de l'étape optionnelle d'élimination des règles. La perte de couverture maximale autorisée est fixée par défaut à 10%, et peut être modifiée.

Paramètres secondaires

Comme la suppression de variables de la définition de la règle correspond à un élargissement de l'espace d'entrée, il est nécessaire de vérifier que cet élargissement ne conduit pas à des simplifications abusives. Le seuil d'homogénéité fixe la valeur maximale de la dispersion des sorties observées des exemples attirés par la règle. 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. Il est surtout significatif pour les problèmes de classification. Par défaut, nous le rendons inactif pour les problèmes de regression (valeur *1000*) et nous proposons *0.8* pour une sortie de type classification.

L'option *Garder la dernière règle*, activée par défaut, indique que la procédure ne doit pas supprimer la dernière règle dont la conclusion correspond à une classe

donnée, dans le cas d'une sortie nette et de l'option classification, ou bien à un sous-ensemble flou donné pour tout type de sortie floue.

Il est possible de spécifier une fichier de validation. Dans ce cas il est utilisé à chaque étape pour retenir ou non la simplification calculée à partir du fichier d'appprentissage.

Résultat

Le résultat de la procédure est résumé dans un fichier, *result.simple*. Ce fichier ne contient qu'une ligne qui décrit le système simplifié : la première colonne indique le nom du système, celui mentionné dans le fichier de configuration, puis les indices décrits dans la section 2 et enfin les caractéristiques de la base de règles, voir la section 3 :

1ère colonne : nom de fichier SIF initial

2ème colonne : indice de performance

3ème colonne : indice de couverture

4ème colonne : erreur max

colonnes suivantes : les caractéristiques de la base de règles

Le système le plus simple résultant de la procédure est édité dans une nouvelle fenêtre.

Une option permet de supprimer l'ensemble des fichiers intermédiaires créés par cette procédure, y compris les fichiers de type SIF.

3.4 Optimisation

Deux options sont proposées dans ce menu. La première, *OLS (Moindres carrés)* permet d'optimiser les conclusions des règles. Elle est présentée dans le menu ols, section 3.2.3. La seconde, *Solis et Wets*, permet d'optimiser les différentes parties du SIF : placement des SEF du partitionnement flou des entrées et des sorties, valeurs de conclusion des règles. Cette méthode consiste en une stratégie d'évolution mono-agent. Elle est décrite en détail dans [7].

Elle est elle même présentée sous a forme de deux options :

— *Solis Wets* C'est la version standard, prête à l'emploi. Elle optimise les entrées, dans l'ordre spécifié, puis la sortie et enfin les règles, lorsque ces composants sont sélectionnés. Le nombre de boucles de cette procédure est un paramètre, dont la valeur par défaut est 1.
Une contrainte de distance minimale entre les noyaux de 2 SEF adjacents est proposée, elle est par défaut à 1% de l'étendue de chaque variable.

Une option validation croisée est disponible. Des paires d'échantillons apprentissage test sont générées, 10 par défaut. Pour chaque paire, un système optimisé est conçu à partir du fichier d'apprentissage et sa performance est mesurée sur le fichier test correspondant. Les systèmes optimisés sont ensuite agrégés en un système final. Chacun des paramètres est remplacé par la médiane du même paramètre dans les systèmes optimisés. Le système final est, en général, plus performant en moyenne sur les échantillons de test que chacun des systèmes optimisés.
Présentation détaillée du module d'optimisation dans :
Serge Guillaume et Brigitte Charnomordic. Parameter optimization of a fuzzy inference system using the fispro open source software. In IEEE Catalog Number : CFP12FUZ-USB, editor, IEEE International Conference on Fuzzy Systems, pages 402-409, Brisbane, Australia, June 2012. IEEE.

— *Solis Wets "sur mesure"* Permet un accès à l'ensemble des paramètres, mais d'un usage plus délicat. La procédure d'optimisation ne trouve pas la meilleure solution dans l'absolu, mais une solution répondant aux critères demandés.
Il est conseillé de procéder par étapes successives, plutôt que de tout optimiser à la fois. On reprendra alors l'optimisation à partir du SIF créé par la procédure d'optimisation précédente.

Quelle que soit la partie du SIF optimisée, cette optimisation est basée sur une amélioration en terme de performance du SIF.

L'algorithme peut obéir à des contraintes, qui seront choisies par l'utilisateur. Les solutions trouvées par l'algorithme ne sont retenues que si les contraintes sont respectées.

Paramètres :

— *choix de la sortie*, si le fichier de données a plusieurs sorties. Le calcul de la performance ne prend en compte qu'une seule sortie.
— *nombre max d'itérations* : valeur par défaut=100.
A augmenter pour laisser plus de chances à l'algorithme de trouver une solution.
— *seuil blanc* : paramètre de l'inférence floue, pour le calcul de la performance (cf. paragraphe correspondant dans la section 2).
— *Paramètres avancés :*
Donne accès à une fenêtre popup, qui permet de règler des paramètres fins de l'algorithme
  — *valeur du germe*
  Initialisation du générateur aléatoire.
  — Constantes de Solis & Wets (Non éditables)

— *écart-type du bruit gaussien* : paramètre interne à l'algorithme, il doit être compris entre 0.0 et 1.0. S'il est trop grand, la convergence est difficile. Valeur par défaut : 0.005.
— *nombre max de contraintes* : valeur par défaut=1000. Nombre de tirages aléatoirepour considérer une configuration comme candidate et incrémenter le nombre d'itérations. A augmenter pour laisser plus de chances à l'algorithme de trouver une solution, dans le cas où l'on a imposé beaucoup de contraintes.

Parties du SIF à optimiser : Dans la fenêtre, les cases cochées indiquent les parties du SIF à optimiser.

— *Sélectionner toutes les entrées* :
Cette case permet de sélectionner/déselectionner d'un coup tous les SEF de toutes les entrées, dont les paramètres seront modifiés.
— *Sélectionner toutes les règles* :
Cette case permet de sélectionner/déselectionner d'un coup toutes les règles dont les conclusions seront modifiés
— *PFF*
Si cette case est cochée, les partitions sont sous la forme de partitions floues fortes, afin de garantir le respect de la sémantique [4] et donc l'interprétabilité des règles.
— *Optimiser les sous-ensembles flous*
Pour chacune des entrées, liste de ses SEF et case à cocher pour optimiser chacun d'entre eux.

Pour la sortie :
Si la sortie est floue, choix PFF/non PFF, et des SEF à optimiser, comme pour les entrées.

— *Règle r*
Pour chacune des règles, case à cocher pour optimiser sa conclusion. Si la sortie est nette, l'option *Vocabulaire de sortie limité* permet de limiter les valeurs des conclusions des règles optimisées à une permutation des valeurs présentes dans la configuration du SIF. Sinon, les conclusions peuvent être quelconques.
— *Afficher la clé*
Permet d'afficher la clé pour la donner en argument du programme *fisopt*.

Résultat

Si une solution meilleure que le SIF initial est trouvée par la procédure, un nouveau SIF correspondant au SIF optimisé est crée et s'ouvre dans une nouvelle fenêtre. Sinon, un message d'avertissement apparaît.

Un fichier perf.res est créé, contenant les performances du SIF initial, de chaque SIF optimisé, du SIF médian, calculées sur le jeu de données initial, sur chaque échantillon de test et sur chaque échantillon d'apprentissage.

4 Menu Options

FisPro est un logiciel localisable : les menus, boutons et messages d'erreur sont disponibles en plusieurs langues (Français, Espagnol, Anglais). Outre le choix de la langue, ce menu permet de modifier l'aspect des fenêtres. L'interface peut facilement prendre en compte d'autres langues, en changeant la variable d'environnement correspondant à la langue, et en traduisant les fichiers de messages du sous-répertoire *resources* du répertoire de classes java.

Deuxième partie
Bibliothèque de fonctions C++

La hiérarchie des classes est présentée dans la figure 4.

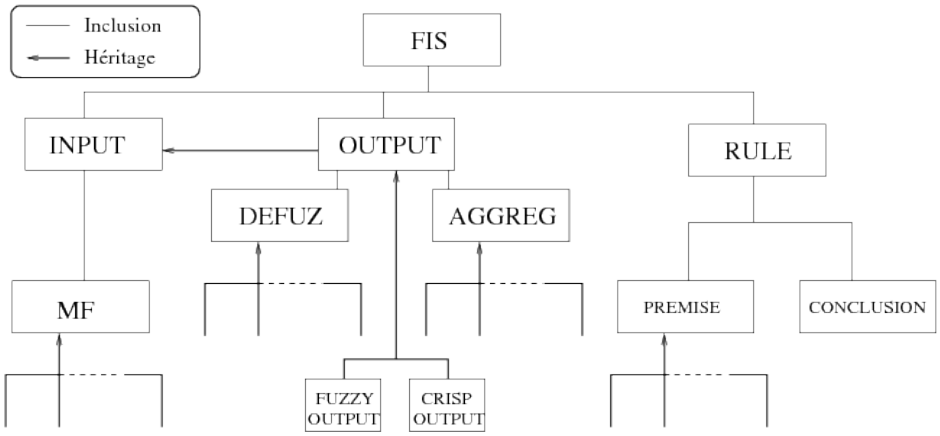


FIGURE 4 – La hiérarchie des classes de base

En règle générale, à chaque classe correspondent deux fichiers, l'un de déclaration portant l'extension .h, l'autre de définition avec l'extension .cpp.

1 Fonctionnement du SIF

Les SIF gérés par cette bibliothèque sont de type *MIMO*, Multiple Input Multiple Output. Le SIF est construit à partir des informations contenues dans le fichier de configuration. Le module de base comporte deux fonctions principales, toutes deux définies au sein de la classe *FIS* : *Infer* et *Performance*. partbiblifonc

La fonction *Infer*, à partir d'un vecteur d'entrée, infère une valeur pour chacune des sorties actives. Les étapes de l'inférence floue sont les suivantes :

1. Fuzzification : cette étape est réalisée au sein de la classe *INPUT*. Ainsi, pour une valeur donnée, la fonction *GetDegs* remplit le tableau *Mfdeg* qui contient les coefficients d'appartenance de la valeur à chacun des sous-ensembles flous de l'entrée. Ce tableau est public.
2. Inférence : la conséquence de chacune des règles est pondérée par le degré de vérité de la règle pour l'exemple. Les règles, pour un système de deux entrées et une sortie, peuvent être de la forme :

   SI Entrée 1 est Sef 2 ET Entrée 2 est Sef 1 ALORS Sortie 1 est Valeur 1
   SI Entrée 1 est Sef 1 ET Entrée 2 est Sef 1 ALORS Sortie 1 est Sef 2

   La fonction *ExecRule* de la classe *RULE* calcule le degré de vérité de la règle pour l'exemple, au sein de la classe *PREMISE*. Il s'agit d'une opération de conjonction des degrés d'appartenance des valeurs des différentes variables d'entrée (*Entree 1* et *Entree 2*) pour les sous-ensembles flous qui décrivent la prémisse de la règle (SEF numéro *2* et SEF numéro *1* pour la première règle ; *1* et *1* pour la seconde). La classe *PREMISE* est construite avec un pointeur sur le tableau des entrées qui lui permet d'accéder au champ *Mfdeg* des différentes entrées. Ce degré de vérité est stocké au sein de la classe *RULE* dans une variable publique *Weight*. Lorsque les règles sont pondérées (voir la section *Menu SIF - Fenêtre Règles*), cette valeur est multipliée par le poids expert.
3. Calcul de la valeur de sortie : Ce calcul est réalisé au sein de la classe *FISOUT* en deux étapes principales.
   — **L'agrégation des règles** : Deux opérateurs d'agrégation sont possibles, qui peuvent s'utiliser aussi bien pour une sortie nette que pour une sortie floue : *max* ou *sum*.
   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.
   Pour chacune de ces valeurs, les degrés de vérité des règles sont cumulés pour une agrégation de type *sum* ou bien seul le maximum est

conservé, cas d'une agrégation de type *max*. Le degré de vérité résultant de l'agrégation est stocké dans le tableau *MuInfer*, le tableau *RuleInfer* contient le numéro de la règle correspondant au max, agrégation *max*, ou bien celui de la dernière règle dont le degré de vérité a été ajouté, agrégation *sum*.
r est le nombre de règles, $w^r(x)$ est le degré de vérité de la règle r pour la donnée multidimensionnelle x.
m est 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).
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 $\forall j = 1, \ldots, m$
$W^j = \left\{ \sum_r \left(w^r(x) \right) \mid C^r = j \right\}$
— sortie floue $\forall j = 1, \ldots, m$
$W^j = min \left(1, \left\{ \sum_r \left(w^r(x) \right) \mid C^r = j \right\} \right)$

— **La défuzzification** : Cette opération utilise les résultats de l'agrégation.

$\widehat{y}_i$, la valeur inférée pour l'exemple i, dépend des opérateurs de défuzzification sont différents et du type de sortie, nette ou floue.

— sortie nette

(a) opérateur de **sugeno**

$$\widehat{y}_i = \frac{\sum_{j=1}^{m} W^j C^j}{\sum_{j=1}^{m} W^j} \tag{2}$$

(b) opérateur **MaxCrisp**

$$\widehat{y}_i = \left\{ j = argmax \left(W^j \right) \mid j = 1 \ldots m \right\}$$

— sortie floue
La figure 5 illustre le processus de défuzzification pour une sortie floue.

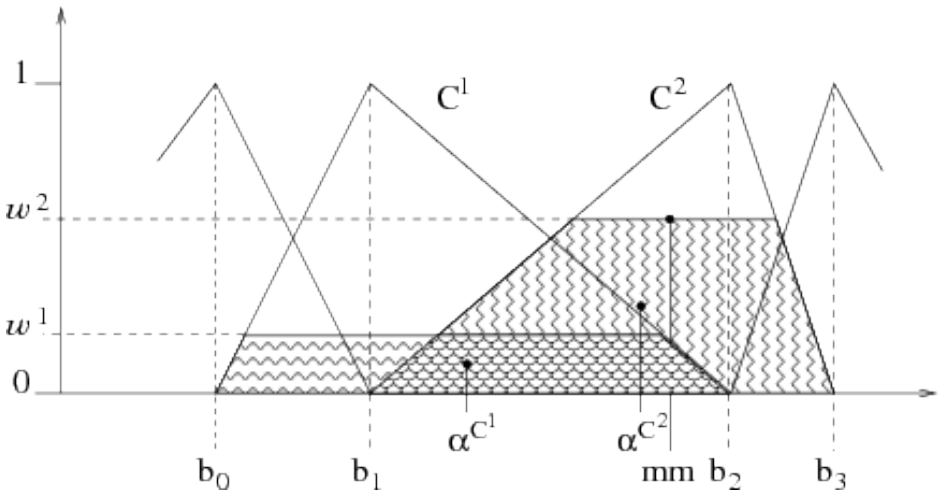


FIGURE 5 – Un exemple de défuzzification

(a) **pondération par les aires** Cet opérateur favorise l'interpolation entre termes linguistiques. La sortie est calculée suivant l'équation 3.

$$\widehat{y}_i = \frac{\sum_{j=1}^{m} \alpha^{C^j}\ area(C_\alpha^j)}{\sum_{j=1}^{m} area(C_\alpha^j)} \tag{3}$$

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 :

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

(b) **moyenne des maxima** La sortie vaut $\widehat{y}_i = mm$ (figure 5). Cet opérateur considère seulement le segment correspondant au niveau d'activation maximum. Aussi, il travaille principalement au sein d'un terme linguistique. D'autres valeurs que la moyenne sont possibles, comme le minimum ou le maximum des maxima.

(c) **sugeno** Cet opérateur utilise la même formule que pour une sortie nette (équation 2), mais C^j représente cette fois le milieu du noyau du SEF j.

‘

Opérateurs d'implication

Pour les règles implicatives, trois opérateurs d'implication sont disponibles :

— Resher-Gaines : $a \rightarrow b = \begin{cases} 1 & si\ a \leq b \\ 0 & sinon \end{cases}$

— Goguen : $a \rightarrow b = \begin{cases} 1 & si\ a = 0 \\ min(1, \frac{b}{a}) & sinon \end{cases}$

— Gödel : $a \rightarrow b = \begin{cases} 1 & si\ a \leq b \\ b & sinon \end{cases}$

Le résultat de l'inférence est une distribution de possibilité, qui peut être défuzzifiée. L'implémentation actuelle ne permet pas de paramétrer l'opérateur de défuzzification. Celui qui est appliqué par défaut est la *Moyenne des maxima*, qui correspond au milieu du noyau de la distribution.

2 La fonction Performance

Elle s'utilise lorsque l'on veut calculer la sortie inférée par le système, pour un ensemble d'exemples. Elle est appelée avec deux arguments principaux : le premier est le numéro de la sortie à tester, le second est l'ensemble des données d'entrée (et de sortie éventuellement).

Cette fonction crée un fichier contenant un tableau, compatible avec les tableurs usuels, dont chaque ligne correspond à un exemple et dont le nombre de colonnes dépend du fichier de données et de la configuration de la sortie, plus précisément du type d'opérateur de défuzzification et de la valeur du paramètre classification.

Si la sortie est présente dans les données, la performance du SIF pour ce fichier de données est estimée par trois indices complémentaires : l'erreur maximum, l'indice de performance proprement dit et l'indice de couverture qui dépend du paramètre seuil.

— Erreur max : Maximum en valeur absolue, sur l'ensemble des exemples, de la différence entre la valeur observée, y_i pour l'exemple i, et celle inférée par le système, $\widehat{y_i}$.

— Indice de couverture : Il possible qu'il existe des exemples qui n'activent que faiblement les règles. Un exemple sera déclaré inactif si le maximum de son degré de vérité, calculé sur l'ensemble des règles, est inférieur à un seuil paramétrable (fixé à *0.1* par défaut). Un exemple inactif n'est pas géré par le système. Le nombre d'exemples inactifs est utilisé pour définir un

indice de couverture. Sa formule est la suivante : $CI = 1 - \frac{I}{N}$. *I* désigne le nombre d'exemples inactifs et *N* le nombre total d'exemples du fichier. Remarque : le degré de vérité d'une règle est lié à l'opérateur de conjonction des prémisses. Avec l'opérateur *prod*, le degré de vérité diminue plus vite qu'avec l'opérateur *min*. L'indice de couverture peut décroître très rapidement si le nombre de variables dans la prémisse des règles est grand.

— Indice de performance : Il dépend du type de sortie. Pour une sortie nette avec l'option classification, il correspond au nombre d'exemples mal classés. Dans tous les autres cas il est paramétré par la fonction SetErrorIndex, et peut être *PI*, *RMSE* ou *MAE* (voir glossaire V. L'indice de performance est renvoyé par la fonction.

Exemple :

La figure 6 montre le fichier résultat qui correspond à un SIF avec des règles conjonctives, une sortie floue avec 3 SEF (MF1, MF2, MF3) et l'option classification.
Dans ce cas, le fichier créé a 9 colonnes : valeur observée, valeur inférée après défuzzification, alarme (0 si tout va bien), 3 colonnes MF1, MF2, Mf3 avec le degré d'appartenance de l'exemple à chaque SEF, l'erreur pour cet exemple, l'indicateur Bl (0 si l'exemple est actif, 1 sinon), et l'erreur cumulée.

Format général :

La première ligne de ce fichier indique le label des colonnes, les étiquettes possibles sont définies dans le fichier fis.h :

— "OBS" : Valeur de la sortie observée, décrite dans le fichier de données
— "INF" : Valeur de la sortie inférée par le SIF
— "Al" : Alarme déclenchée lors de l'inférence (voir ci dessous)
— "ClINF" : Classe inférée pour les sorties nettes avec option classification
— "CLAl" : Alarme déclenchée lors de l'inférence de la classe (voir ci dessous)
— "Err" : La différence entre la sortie inférée et la sortie observée
— "Bl" : Vaut 1 si l'exemple est inactif, 0 sinon
— "CErr2" : Le carré de l'erreur cumulée sur l'ensemble des exemples précédents

ATTENTION : L'erreur cumulée ne tient pas compte des exemples inactifs.

La première colonne est la sortie observée, si elle est disponible dans le fichier de données, car l'inférence peut aussi être faite en l'absence de sortie observée. Suivent des colonnes en nombre variable, ce nombre dépendant du type de sortie :

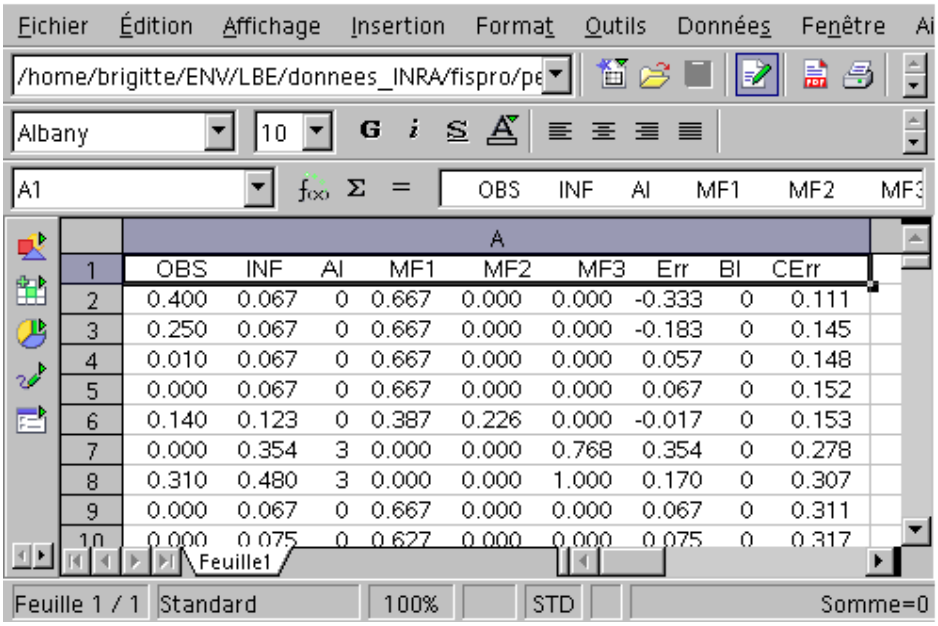


FIGURE 6 – Exemple de fichier résultat de l'inférence-sortie floue avec 3 SEF, option classification

Sortie	Classif	Défuzzification	Nb. champs	Champs présents			
nette	oui	sugeno	4	V. inférée	Alarme	Classe inférée	Alarme
nette	non	sugeno	2	V. inférée	Alarme		
nette	oui	MaxCrisp	2	V. inférée	Alarme	$\mu_1, \mu_2 \ldots \mu_n$	
nette	non	MaxCrisp	2	V. inférée	Alarme		
floue	oui	sugeno ou area	n+2	V. inférée	Alarme	$\mu_1, \mu_2 \ldots \mu_n$	
floue	non	sugeno ou area	2	V. inférée	Alarme		
floue	oui	MeanMax	n+2	V. inférée	Alarme	$\mu_1, \mu_2 \ldots \mu_n$	
floue	non	MeanMax	2	V. inférée	Alarme		

La colonne suivante indique, lorsque la valeur observée est disponible, l'écart entre celle-ci et la valeur inférée.

Remarque : $\mu_1, \mu_2 \ldots \mu_n$ ne sont présents que si l'exemple active au moins une règle. Pour une sortie floue avec opérateurs *sugeno* ou *area*, ils sont les degrés d'appartenance de la valeur inférée aux SEF de sortie numéro $1, 2 \ldots n$. Dans le cas d'une sortie floue avec opérateur *MeanMax* ou d'une sortie nette avec *MaxCrisp* ils représentent le niveau d'activation de chacune des sorties possibles, SEF pour la sortie floue ou bien valeurs réelles pour la sortie nette.

La colonne *Bl* indique si l'exemple est inactif (valeur 1 dans ce cas, sinon 0). Enfin, la dernière colonne contient la valeur courante de l'indice de performance.

Valeurs possibles pour l'alarme :
valeurs entières qui dépendent du type d'opérateur de défuzzification.

— NOTHING (Valeur 0) : Tous types. Tout va bien, rien à signaler.
— NO_ACTIVE_RULE (Valeur 1) : Tous types. L'exemple d'entrée n'a activé aucune des règles de la base. L'activation est ici prise au sens strict ($\mu > 0$), et ne tient pas compte du seuil d'activation pour les exemples inactifs (colonne *Bl*).
— AMBIGUITY (Valeur 2) : SugenoClassif (Sortie nette) et MaxCrisp. La différence entre les deux classes les plus possibles est inférieure au seuil (valeur par défaut AMBIGU_THRES = 0.1).
— NON_CONNEX_AREA (Valeur 3) : WeArea - Est declenchée lorsque l'aire définie dans l'espace de sortie n'est pas connexe. Seuls sont considérés les SEF dont le niveau d'activation est supérieur au seuil (valeur par défaut MIN_THRES = 0.1).
— NON_CONNEX_SEGMENT (Valeur 4) : MeanMax - Se produit lorsque le segment des maxima n'est pas connexe. Deux niveaux sont considérés comme égaux, si leur différence est inférieure au seuil (valeur par défaut EQUALITY_THRES = 0.1).

Cette fonction renseigne l'indice de couverture défini plus haut, et calculé en fonction du seuil passé en argument.

Format général du fichier pour les règles implicatives

Si la sortie sur laquelle est calculée la performance est implicative, de nouvelles colonnes sont introduites dans le fichier, dans l'ordre suivant.

— k colonnes $MF_1, \ldots MF_k$ juste après la colonne *OBS*. k est le nombre de SEF de sortie. La valeur écrite dans chaque colonne est le degré d'appartenance de la sortie observée au SEF correspondant.
— k colonnes $MF_1, \ldots MF_k$ juste après la colonne *Al*. k est le nombre de SEF de sortie. La valeur écrite dans chaque colonne est le degré d'appartenance de la sortie inférée au SEF correspondant.
— *MINK*, qui contient le minimum du noyau de la distribution de possibilité inférée.
— *MAXK*, qui contient le maximum du noyau de la distribution de possibilité inférée.
— *MINS*, qui contient le minimum du support de la distribution de possibilité inférée.
— *MAXS*, qui contient le maximum du support de la distribution de possibilité inférée.

— *MATCH*, qui donne le degré d'adéquation de la distribution de possibilité inférée avec la valeur observée.

3 Les caractéristiques d'une base de règles

La structure *InfoRB* résume les caractéristiques de la base de règles. Elle comprend les attributs suivants :

— maxR : nombre maximal de règles possibles en fonction des partitions des entrées
— nR : nombre de règles
— maxVr : nombre maximum de variables dans une règle
— meanVr : nombre moyen de variables par règle
— nVar : nombre de variables distinctes utilisées par les règles
— meanMF : nombre moyen de MF par variable utilisée
— nClass : nombre de classes ou de MF en sortie
— nRc : nombre de règles par classe ou par MF

Cette structure possède également deux méthodes permettant de les manipuler : *Print* et *WriteHeader* pour écrire la description de chacune des colonnes. Chacun des champs, valeurs des caractéristiques ou label des colonnes, est séparé du suivant par le caractère '&' (Les utilisateurs latex apprécieront !)

La fonction *AnalyzeRB* de la classe FIS permet de renseigner une structure de type *InfoRB* pour une sortie d'un système donné.

Troisième partie
Structure détaillée des classes

Une documentation en html des classes C++ et java à l'usage des programmeurs est disponible sur le site de FisPro.

Quatrième partie
Problèmes connus

— La plupart des méthodes d'apprentissage fonctionnent avec un seuil de tolérance EPSILON=10^{-6}. Avant d'utiliser ces méthodes, il est préférable de normaliser les données entre 0 et 1, si leur étendue est très grande ou au contraire très faible.

— La procédure d'apprentissage HFP, qui travaille seulement à partir d'un fichier de données, considère que la sortie est unique, et correspond à la dernière colonne de ce fichier.
— Certaines machines virtuelles Java (jvm) ne rendent pas la mémoire, une fois qu'elle est allouée, même si elle n'est plus utilisée. De grands tableaux peuvent être alloués lors de la lecture de fichiers de données volumineux. La seule solution pour libérer la mémoire est dans ce cas de sauvegarder le travail en cours, de sortir de FisPro et de le relancer.

Cinquième partie
Petit glossaire de la logique floue

1 Variable linguistique et système d'inférence floue

— 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 7 montre un ensemble de forme triangulaire.

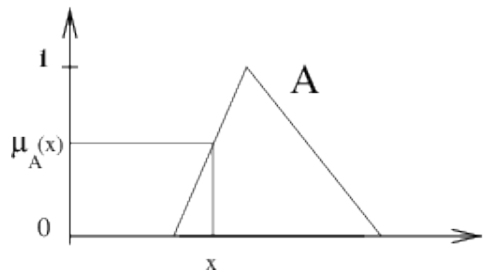


FIGURE 7 – Un ensemble flou de forme triangulaire

— Support d'un ensemble flou : l'ensemble des points pour lesquels le degré d'appartenance est non nul, $S_A = \{x|\mu_A(x) > 0\}$

— Noyau d'un ensemble flou : l'ensemble des points pour lesquels le degré d'appartenance vaut 1, $K_A = \{x|\mu_A(x) = 1\}$

— Prototype d'un ensemble : un point est un prototype d'un ensemble flou s'il appartient au noyau.

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

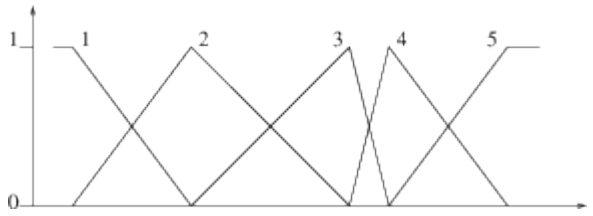


FIGURE 8 – Exemple de partition floue forte

— Variable linguistique : Dans une partition floue forte, chaque ensemble correspond à un concept linguistique, par exemple *Très faible*, *Faible*, *Moyen*, *Elevé*, *Très élevé*. Pour le raisonnement les variables sont manipulées par les termes linguistiques ainsi définis, les ensembles flous assurent la correspondance avec l'univers numérique.

— 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.

— Opérateurs :
  — est : la relation $x\ est\ A$ est quantifiée par le degré d'appartenance de la valeur x au sous-ensemble flou A. Elle mesure le niveau de correspondance entre la valeur numérique x et le concept linguistique représenté par l'ensemble A.
  — ET : opérateur de conjonction, noté $\wedge$. Il généralise l'ntersection et permet, notamment, d'agréger les degrés d'appartenance au sein d'une prémisse multidimensionnelle. Les plus employés sont le minimum et le produit.
  — OU : opérateur de disjonction, généralise l'union. Les plus employés sont le maximum et la somme (limitée à 1).

— Règle incomplète : Une règle floue sera dite incomplète si sa prémisse est définie par un sous-ensemble des variables d'entrée seulement. La règle, $SI\ x_2\ est\ A_2^1\ ALORS\ y\ est\ C_2$, est incomplète car la variable x_1 n'intervient pas dans sa définition. Les règles formulées par les experts sont principalement des règles incomplètes. Formellement, une règle incomplète est définie par une combinaison implicite de connecteurs logiques ET et OU opérant sur l'ensemble des variables. Si l'univers de la variable x_1 est découpée en 3 sous-ensembles flous, la règle incomplète ci-dessus peut aussi s'écrire de la façon suivante :

$$SI\ (x_1\ est\ A_1^1\ OU\ x_1\ est\ A_1^2\ OU\ x_1\ est\ A_1^3)\ ET\ x_2\ est\ A_2^1\ ALORS\ y\ est\ C_2.$$

— 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.

— 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.

— Activité : Un exemple active une règle, ou bien une règle active un exemple, si le degré de vérité de la règle pour l'exemple est non nul.

— Seuil blanc : Un exemple est dit blanc ou inactif si le maximum de son degré de vérité, calculé sur l'ensemble des règles, est inférieur à un seuil, appelé seuil blanc.

— 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 SEF de la prémisse.

— Système d'inférence floue (SIF) : Un système d'inférence floue est formé de trois blocs comme indiqué sur la figure 9. 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 : Il s'agit d'une distribution de possibilité dont l'interprétation varie avec le type de règles. Elle dépend bien entendu

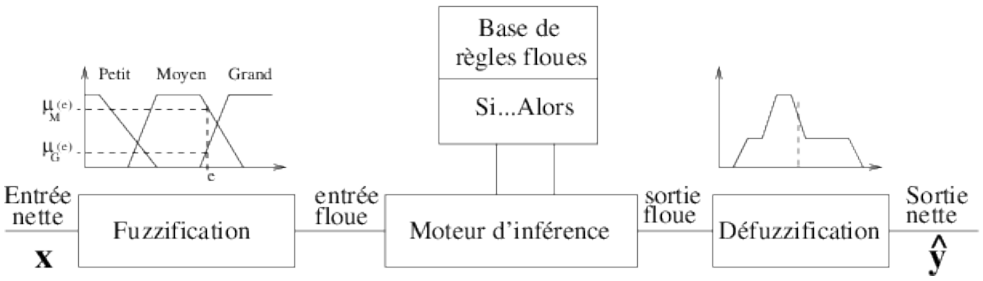


FIGURE 9 – Un système d'inférence floue

de la base de règles et des différents opérateurs. Elle peut être réduite à une valeur précise que l'on note $\widehat{y}_i$ pour l'exemple i.

— Les règles, de la forme *Si x est A alors y est C* sont de deux types :

1. Règles conjonctives : Dans ce cas la règle représente une connaissance positive, les entrées (*A*) et sortie (*C*) sont des paires de valeurs conjointement possibles. La sémantique de la règle est : Si l'entrée est de type *A* alors une valeur possible pour la sortie est *C*. La relation entre l'entrée et la sortie de la règle est modélisée par une conjonction (t-norme). Ces règles s'inspirent de la philosophie des bases de données, *c'est possible car je l'ai observé*, et implémentent le raisonnement par similarité. La sortie est une distribution de possibilité garantie.
2. Règles implicatives : La règle représente une connaissance négative, et la sémantique devient "Si l'entrée est de type *A* alors la sortie est obligatoirement dans *C*". Cette deuxième forme d'expression, plus formelle, est issue de la branche logique et intelligence artificielle de l'informatique. Au lieu de procéder par accumulation, elle procède par élimination : sont écartées les valeurs qui ne respectent pas les contraintes. La relation entre l'entrée et la sortie de la règle est modélisée par une implication, éventuellement floue. La sortie est une distribution de possibilité potentielle (usuelle). Le passage, pour un expert, de la connaissance positive à la connaissance négative suppose une étape de modélisation.

Pour une comparaison plus approfondie entre les deux types de règles, se reporter à [12].

— Mécanisme d'inférence : deux sont nécessaires

1. FITA : First Infer Then Aggregate. Il permet d'inférer une valeur pour chacune des règles avant de les agréger. Il est utilisé pour les règles

conjonctives ainsi qu'avec les règles implicatives lorsque les valeurs d'entrée sont précises.

2. FATI : First Aggregate Then Infer. Dans ce cas, toutes les contraintes (règles) sont agrégées pour inférer la distribution de possibilité. Plus lourde à mettre en œuvre, cette méthode est indispensable pour des règles implicatives dès lors qu'au moins une des données est imprécise.

2 Règles conjonctives

Les règles conjonctives sont rassemblées en deux familles :

1. Mamdani : La conclusion est un ensemble flou, la règle 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 limitée à une constante au lieu de cette combinaison linéaire des entrées.

— **L'agrégation** : Elle est disjonctive pour les règles conjonctives, 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.
Pour chacune de ces valeurs, les degrés de vérité des règles sont cumulés pour une agrégation de type *sum* ou bien seul le maximum est conservé, cas d'une agrégation de type *max*. Le degré de vérité résultant de l'agrégation est stocké dans le tableau *MuInfer*, le tableau *RuleInfer* contient le numéro de la règle correspondant au max, agrégation *max*, ou bien celui de la dernière règle dont le degré de vérité a été ajouté, agrégation *sum*.
Notons :
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).
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 $\forall j = 1, \ldots, m$
$W^j = \left\{ \sum_r \left(w^r(x) \right) \mid C^r = j \right\}$
— sortie floue $\forall j = 1, \ldots, m$
$W^j = min \left(1, \left\{ \sum_r \left(w^r(x) \right) \mid C^r = j \right\} \right)$

— **La défuzzification** : Cette opération, indispensable pour les règles conjonctives, 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**

$$\widehat{y}_i = \frac{\sum_{j=1}^{m} W^j C^j}{\sum_{j=1}^{m} W^j} \tag{4}$$

2. opérateur **MaxCrisp**

$$\widehat{y}_i = \left\{ j = argmax \left(W^j \right) \mid j = 1 \ldots m \right\}$$

— sortie floue
La figure 10 illustre le processus de défuzzification pour une sortie floue.

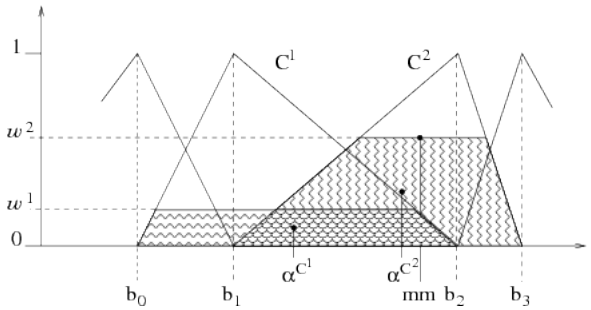


FIGURE 10 – Un exemple de défuzzification

1. **pondération par les aires** Cet opérateur favorise l'interpolation entre termes linguistiques. La sortie est calculée suivant l'équation 5.

$$\widehat{y}_i = \frac{\sum_{j=1}^{m} \alpha^{C^j} \; area(C_\alpha^j)}{\sum_{j=1}^{m} area(C_\alpha^j)} \tag{5}$$

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 10). Cet opérateur considère seulement le segment correspondant au niveau d'activation maximum. Aussi, il travaille principalement au sein d'un terme linguistique.
3. **sugeno** Cet opérateur utilise la même formule que pour une sortie nette (équation 4), mais C^j représente cette fois le milieu du noyau du SEF j.

3 Règles implicatives

L'agrégation des règles implicatives est conjonctive, signifiant que chaque règle impose une contrainte sur la sortie. La prise en compte de l'ensemble des contraintes se fait donc au moyen d'un opérateur d'intersection (t-norme). Si aucune règle ne s'applique, la sortie est l'ensemble flou universel, lorsqu'on ne sait rien toutes les valeurs de sorties sont également possibles.

Trois opérateurs d'implication sont disponibles :

— Resher-Gaines : $a \to b = \begin{cases} 1 & si \; a \leq b \\ 0 & sinon \end{cases}$

— Goguen : $a \to b = \begin{cases} 1 & si \; a = 0 \\ min(1, \frac{b}{a}) & sinon \end{cases}$

— Gödel : $a \to b = \begin{cases} 1 & si \; a \leq b \\ b & sinon \end{cases}$

L'opérateur *Resher-Gaines* n'est pas flou : la distribution de possibilité produite correspond au noyau des deux autres, *Goguen* et *Gödel*.

Les partitions des variables de sortie doivent être adaptées pour tenir compte du mode d'agrégation particulier aux règles implicatives. Nous proposons le concept de partition quasi forte (PQF) comme compromis entre interprétabilité et cohérence. En effet la PQF est dérivée d'une PFF, dont nous connaissons les avantages en terme d'interprétabilité, et les ensembles flous supplémentaires permettent d'assurer une intersection non vide lorsque plusieurs règles sont activées du fait de la multi-appartenance en entrée. Le figure 11 montre une PFF et la PQF équivalente.

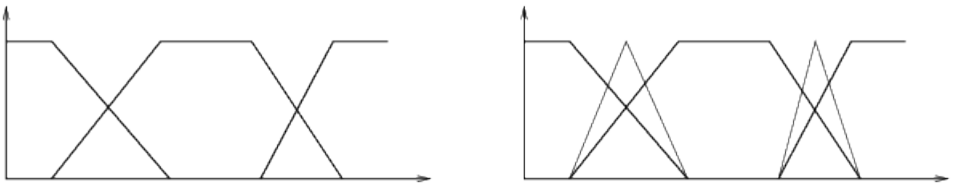


FIGURE 11 – Partitions floue forte et quasi forte équivalentes

Les règles implicatives offrent plusieurs avantages par rapport aux règles conjonctives :

- Modus Ponens : La version classique, $A \wedge (A \rightarrow C) \models C$, est généralisée par l'usage des ensembles flous et d'une implication floue : $A' \wedge (A \rightarrow C) \models C'$.
- Indépendance inférentielle : Dans la base de règles $\{A_j \rightarrow C_j, j = 1, \ldots, n\}$, la sortie inférée pour la donnée A_i est égale à C_i, s'il existe la règle $A_i \rightarrow C_i$, quelles que soient les autres règles présentes dans le système.
- Respect de l'imprécision : Comme l'accumulation des règles se traduit par une élimination des valeurs possibles pour la sortie, l'imprécision résultante a du sens et peut être analysée.
  Il est également possible d'en extraire une valeur précise par défuzzification. L'implémentation actuelle ne permet pas de paramétrer l'opérateur de défuzzification. Celui qui est appliqué par défaut est la *Moyenne des maxima*, qui correspond au milieu du noyau de la distribution.
- Détection des conflits : Une distribution de possibilité vide en sortie traduit une incohérence dans la base de règles, les contraintes ne sont pas compatibles.

4 Apprentissage

— 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.

— Indice de couverture : Il possible qu'il existe des exemples qui n'activent que faiblement les règles. Un exemple sera déclaré inactif si le maximum de son degré de vérité, calculé sur l'ensemble des règles, est inférieur à un seuil paramétrable (fixé à *0.1* par défaut). Un exemple inactif n'est pas géré par le système. Le nombre d'exemples inactifs est utilisé pour définir un indice de couverture. Sa formule est la suivante : $CI = \frac{A}{n}$. *A* désigne le nombre d'exemples actifs et *n* le nombre total d'exemples du fichier.

— Indice d'error ou de performance. Ils sont calculés à partir des seuls exemples actifs. Pour une sortie de type classification : $PI = \sum_{i=1}^{A} mc(i)$ où $mc(i)$ vaut 0 si l'exemple est bien classé, 1 sinon.
Trois sont disponibles pour une sortie continue :
— $PI = \frac{1}{A}\sqrt{\sum_{i=1}^{A} \|\widehat{y_i} - y_i\|^2}$
— $RMSE = \sqrt{\frac{1}{A}\sum_{i=1}^{A} \|\widehat{y_i} - y_i\|^2}$
— $MAE = \frac{1}{A}\sum_{i=1}^{A} |\widehat{y_i} - y_i|$
PI est l'index de performance historique de *FisPro*, *RMSE* signifie Root Mean Squared Error, et *MAE* est Mean Absolute Error.

5 Distribution de possibilité

Soit U un ensemble d'événements élémentaires, u. On appelle mesure de possibilité et on note Π une fonction définie sur l'ensemble des parties $P(U)$ de U, à valeurs dans $[0, 1]$ telle que :
— $\Pi(\emptyset) = 0$
— $\Pi(U) = 1$

— $\forall i, A_i \in P(U), \Pi(\bigcup A_i) = sup\Pi(A_i)$

Un degré de possibilité $\Pi(A) = 1$ indique que l'événement A est complètement possible, inversement $\Pi(A) = 0$ signifie que A est impossible.

Une distribution de possibilité assigne à chaque élément u de U une possibilité $\pi(u) \in [0, 1]$. La distribution est normalisée : $\sup_{u \in U} \pi(u) = 1$.

Possibilité garantie

Une mesure de possibilité garantie Δ est une fonction définie sur l'ensemble des parties de U, à valeurs dans $[0, 1]$ telle que :

— $\Delta(\emptyset) = 1$

— $\Delta(A_1 \cup A_2) = min(\Delta(A_1), \Delta(A_2))$

Relation entre le degré de possibilité et degré de possibilité garanti :

$$\forall A \subseteq U, \Delta(A) = inf_{u \in A} \pi(u)$$

.

Donc, lorsque $\Delta(A) = \alpha$, tous les événements élémentaires $u \in A$ sont garantis possibles au niveau α.

Les distributions de possibilité garantie sont notées δ.

Interprétation des degrés de possibilité et possibilité garantie

— $\pi_X(u) = 1$: rien n'empêche x d'être égal à u, u est une valeur complètement possible.

— $\pi_X(u) = 0$: u est une valeur impossible pour x.

— $\delta_X(u) = 1$: u est une valeur possible pour x, par exemple parce qu'elle a été observée.

— $\delta_X(u) = 0$: cela ne signifie pas que u est une valeur impossible pour x, mais indique seulement que rien ne la garantit.

Références

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

[2] 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.

[3] 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.

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

[5] Sébastien Destercke, Serge Guillaume, and Brigitte Charnomordic. Building an interpretable fuzzy rule base from data using orthogonal least squares-application to a depollution problem. *Fuzzy Sets and Systems*, 158 :2078–2094, 2007.

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

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

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

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

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

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

[12] Hazael Jones, Brigitte Charnomordic, Didier Dubois, and Serge Guillaume. Practical inference with systems of gradual implicative rules. *IEEE Transactions on Fuzzy Systems*, 17 (1) :61–78, 2009.

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

[14] 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.

[15] R. Weber. Fuzzy-id3 : A class of methods for automatic knowledge acquisition. In *International conference on fuzzy logic and neural networks*, pages 265–268, 1992.

[16] L. A. Zadeh. Is there a need for fuzzy logic ? *Information Sciences*, 178 (13) :2751–2856, 2008.