Débuter avec FisPro

fispro@supagro.inra.fr

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 oenologie. 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 ([**?**]). 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 mécanisme 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.

Auteurs

- Conception et implémentation C++ : Serge GUILLAUME du Cemagref (https ://itap.cemagref.fr/pages-personnelles/serge-guillaume), Brigitte CHARNOMORDIC de l'INRA (http ://www.inra.fr)
- Interface java : Jean-Luc LABLEE, du Cemagref

- Contributions :
  - Pierre-Marie BOYER, interface JNI (Java-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 ;
  - Mathieu GRELIER, visualisation des données (Table, 2D et 3D) ;
  - Sébastien DESTERCKE, induction de règles par moindres carrés (ols)
  - Jean-Michel FATOU, icônes de FisPro et conception graphique du site WEB
  - Vincent THERRY, avec l'appui de la société Envilys, chaînage de systèmes (superfis)
  - Hazaël JONES, conception de la partie *règles implicatives*
  - Lydie DESPERBEN, implémentation C++ de la partie *règles implicatives*
  - Anne TIREAU, interface java de la partie *règles implicatives*.
  - Russel STANDISH, version optimisée et parallélisable (standard OPENMP),

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 (http ://www.transferts-lr.org) et dont le partenaire industriel était la cave coopérative "La Malepère", Arzens, Aude.

FisPro est un logiciel *open source*, disponible sur le site

http ://www.inra.fr/internet/Departements/MIA/M/fispro/

Notions élémentaires

Un système est aussi appelé SIF, pour système d'inférence floue.

On utilisera le terme SEF pour sous-ensemble flou (voir glossaire de logique floue, section 3).

Au démarrage de FisPro, aucun système n'est présent.

Vous pouvez ouvrir un système existant, ou en créer un nouveau, soit à partir de données, soit de toutes pièces.

3

Ce petit guide donne la démarche à suivre dans ce dernier cas, adapté à l'entrée de règles expertes.

Remarques :

- Pour la saisie des nombres, le séparateur décimal est la virgule (,).
- Les modifications, ajout d'entrée, de sortie ou de SEF, sont prises en compte immédiatement dans le SIF. Les fenêtres intermédiaires peuvent être fermées sans perte de données.
- Les options non disponibles en fonction du contexte sont grisées dans les menus.
- La définition experte ne fait intervenir que le menu *SIF*. Le menu *Apprentissage* est pour d'autre usages (induction automatique).
- Le menu *Données* permet d'ouvrir un fichier de données, de les visualiser, et d'inférer sur l'ensemble du fichier.

Rappel :

L'option *Langue* du menu Options permet l'affichage des menus et des messages dans la langue de votre choix.

Table des matières

1 Créer un système simple

L'exemple choisi crée un système très simple : 1 entrée, 1 sortie et 3 règles.

L'entrée est le degré du vin, la sortie est son prix. Les règles font évoluer le prix en fonction du degré.

- Commencer par choisir l'option *Nouveau* du menu *Sif*
- Le nom par défaut *Nouveau SIF* s'affiche dans le champ *Nom*. Il est modifiable par simple saisie. Donnes-lui le nom *coop*.
- La conjonction est l'opérateur de combinaison des prémisses de la règle. C'est par défaut le produit.

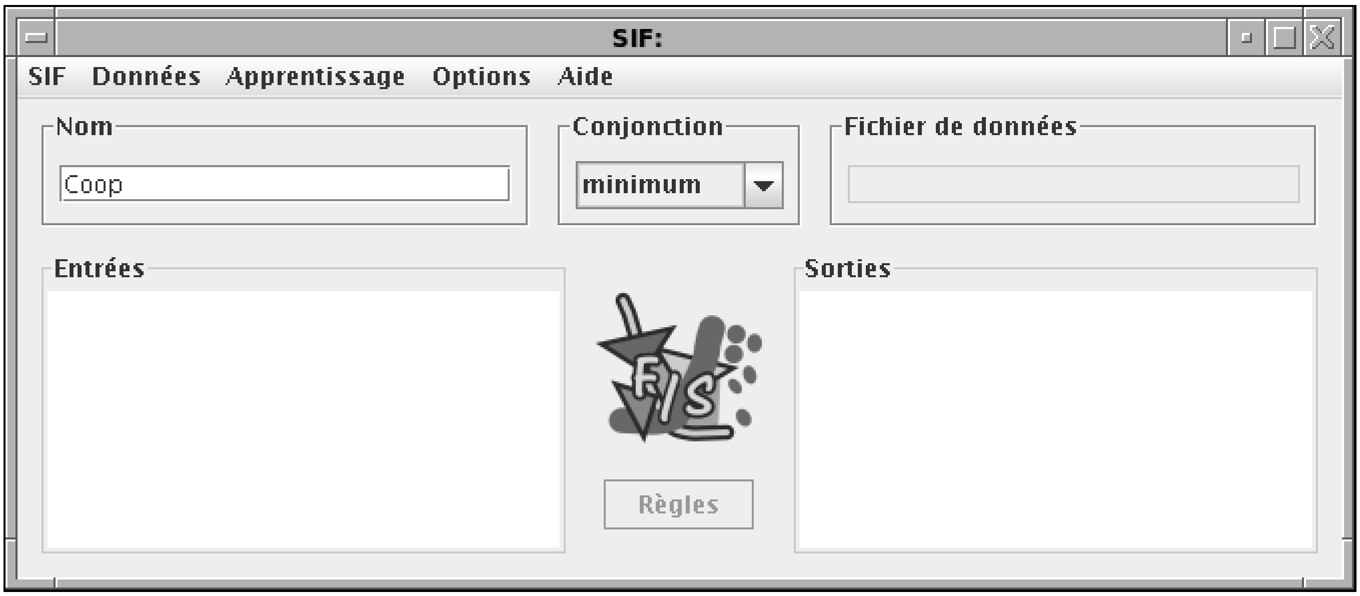


FIG. 1 – Fenêtre principale de FisPro

1.1 Définir une entrée

Pour ajouter une entrée, utiliser soit l'option *Nouvelle Entrée* du menu SIF, soit un clic droit sur la zone *Entrées* de la fenêtre principale. La fenêtre *Entrée* apparaît.

Une entrée est caractérisée par son domaine, et sa partition floue, c'est-à-dire les sous-ensembles flous (SEF) qui la composent.

Elle peut être active (par défaut) ou inactive.

Donnez-lui le nom *Degré*.

- Le domaine de variation d'une entrée est par défaut [0,1].
  Pour le changer, choisir le menu Domaine de la fenêtre *Entrée*, et entrer les nouvelles valeurs du domaine de variation : 9 et 14.
- La méthode la plus rapide, dans ce cas, pour définir la partition est l'option *Grille irrégulière* du menu SEF, avec le nombre de SEF correspondant à la finesse voulue pour les labels linguistiques (3 par défaut). Cette option permet de donner la position des sommets de chaque SEF :
- Les SEF s'affichent dans la partie inférieure de la fenêtre : demi-trapèzes aux extrémités du domaine, et triangles ailleurs.
- Pour la clarté du SIF, les noms des SEF sont importants, car ils apparaissent dans les règles.
  - SEF 1 : nom *Faible*, sommets 9, 11,5 et 12
  - SEF 2 : nom *Moyen*, sommets 11,5, 12 et 12,5
  - SEF 3 : nom *Elevé*, sommets 12, 12,5 et 14

  Vous obtenez la partition ci-dessous.

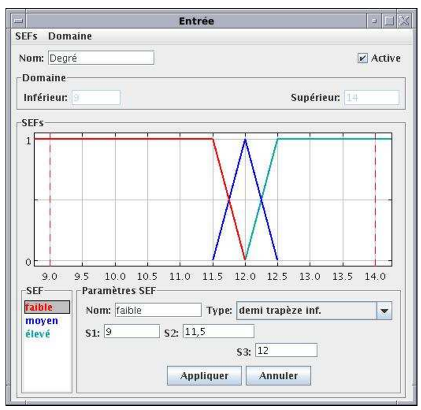


FIG. 2 – Définition d'une entrée dans FisPro

1.2 Modifier une entrée ou une sortie

Pour modifier une entrée ou une sortie, double cliquer sur son nom dans la fenêtre principale.

1.3 Définir une sortie

Pour ajouter une sortie, utiliser soit l'option *Nouvelle Sortie* du menu SIF, soit un clic droit sur la zone *Sortie* de la fenêtre principale. La fenêtre *Sortie* apparaît.

Donnez lui le nom *Prix*. Une sortie est caractérisée avant tout par son domaine, et sa nature : sortie nette ou floue.
La nature est liée au raisonnement mis en oeuvre dans le SIF.

- Avec une sortie nette, la conclusion des règles peut être une valeur numérique quelconque.
- Avec une sortie floue, la conclusion des règles est le label linguistique d'un SEF, par exemple *Petit*, *Moyen* ou *Grand*.

Indépendamment du type de sortie, le résultat de l'inférence est une valeur numérique.

Autres paramètres :

- Valeur par défaut : c'est la valeur que prendra le résultat de l'inférence, pour cette sortie, au cas où aucune règle n'est activée.
- Défuzzification et disjonction : choix liés à la façon d'agréger les conclusions des règles (voir 3).
- classif : cocher cette case pour arrondir le résultat de l'inférence à la classe (valeur discrète) la plus proche.
  Les classes possibles sont limitées aux valeurs des conclusions des règles.
  Si la sortie est floue, il faut en plus définir sa partition floue, comme pour une entrée floue.

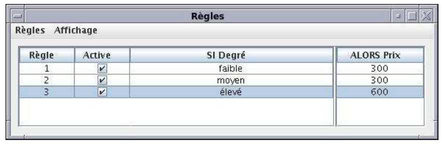
FIG. 3 – Définition d'une sortie dans FisPro

1.4 Définir une règle

Pour créer les règles, cliquer sur la case *Règles* de la fenêtre principale. La fenêtre *Règles* s'affiche.

Dans cette fenêtre, pour ajouter une règle, utiliser soit l'option *Nouvelle Règle* du menu SIF, soit un clic droit sur la colonne *Règle*.

Cliquer successivement sur les colonnes représentant les variables, pour choisir les labels qui interviendront dans la règle, ou pour entrer la valeur numérique (sortie nette).



Règle	Active	SI Degré	ALORS Prix
1	✓	faible	300
2	✓	moyen	300
3	✓	élevé	600

FIG. 4 – Définition de règles dans FisPro

1.5 Inférer

L'option *Inférer* du menu SIF permet d'explorer le fonctionnement du système.
Le raisonnement à base de règles floues est représenté graphiquement.

Les valeurs des variables d'entrée sont saisies manuellement (déplacement du curseur ou saisie de valeur).
Le système affiche la valeur inférée pour la sortie, avec plusieurs informations intermédiaires permettant de comprendre les étapes du raisonnement flou :

Pour chaque règle :

- degré d'appartenance de la valeur à chaque SEF présent dans les prémisses de la règle, visualisé comme un taux de remplissage du SEF
- degré de vérité de la règle.
  Ce degré de vérité est, dans notre exemple, égal au degré d'appartenance précédent, car le système ne comporte qu'une seule entrée. Dans les cas plus complexes, ce degré est obtenu par la combinaison des SEF des prémisses.
  Il est visualisé sous forme de valeur numérique si la sortie est nette, ou comme le taux de remplissage du SEF de conclusion, si la sortie est floue.

Pour chaque sortie :

- La valeur inférée est affichée en haut à droite de la fenêtre, *480* ici.

La sortie *Prix* étant nette, avec une défuzzification de type Sugeno et une agrégation de type somme, elle s'obtient par une simple moyenne des conclusions des règles, pondérées par les degrés de vérité des règles, affichés à droite. Les conclusions des règles sont indiquées entre parenthêses.

Pour l'ensemble des degrés compris dans le domaine de l'entrée *Degré*, donc entre 9 et 14, on peut obtenir le prix correspondant. La progression du prix est continue, grâce aux capacités d'interpolation du système.

2 Un système plus complexe

Nous allons rendre plus réaliste l'exemple précédent, en ajoutant une variable et en modifiant les règles pour qu'elles tiennent compte des deux variables.

L'entrée supplémentaire est le rendement de la parcelle.

dans notre cas.
Les conclusions sont initialisées avec la valeur 1.

Commencez par simplifier les règles relatives au degré faible et au rendement très élevé, en éliminant les règles inutiles, puis entrez les conclusions voulues dans la colonne du Prix, afin d'obtenir le système de règles ci-dessus.

2.3 Visualiser l'inférence

Utilisez l'option *Inférer* du menu *SIF*, et changez tour à tour les valeurs des deux variables.

3 Petit glossaire de la logique floue

3.1 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 10 montre un ensemble 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 $A_1, A_2, \ldots$

- Partition floue forte : La partition de la variable X_i sera appelée une partition floue forte si $\forall x \in X_i, \sum\limits_j \mu_{A_j^i}(x) = 1$.

- 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 12. 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 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 [**?**].

– 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.

3.2 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 : $\forall j = 1, \ldots, m$
    $W^j = \left\{ \max_r (w^r(x)) \mid C^r = j \right\}$
  – sum :
    – sortie nette $\forall j = 1, \ldots, m$
      $W^j = \left\{ \sum_r (w^r(x)) \mid C^r = j \right\}$

– sortie floue $\forall j = 1, \ldots, m$
$W^j = min\left(1, \left\{\sum_r (w^r(x)) \mid C^r = j\right\}\right)$

– **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**

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

2. opérateur **MaxCrisp**

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

– sortie floue
La figure 13 illustre le processus de défuzzification pour une sortie floue.

1. **pondération par les aires** Cet opérateur favorise l'interpolation entre termes linguistiques. La sortie est calculée suivant l'équation 2.

$$\widehat{y}_i = \frac{\sum_{j=1}^{m} \alpha^{C^j}\ area(C^j_\alpha)}{\sum_{j=1}^{m} area(C^j_\alpha)} \tag{2}$$

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^j_α, et C^j_α un nouvel ensemble flou, défini à partir de C^j comme :

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

2. **moyenne des maxima** La sortie vaut $\widehat{y}_i = mm$ (figure 13). 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 1), mais C^j représente cette fois le milieu du noyau du SEF j.

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

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 14 montre une PFF et la PQF équivalente.

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.

3.4 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.

3.5 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] 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.

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

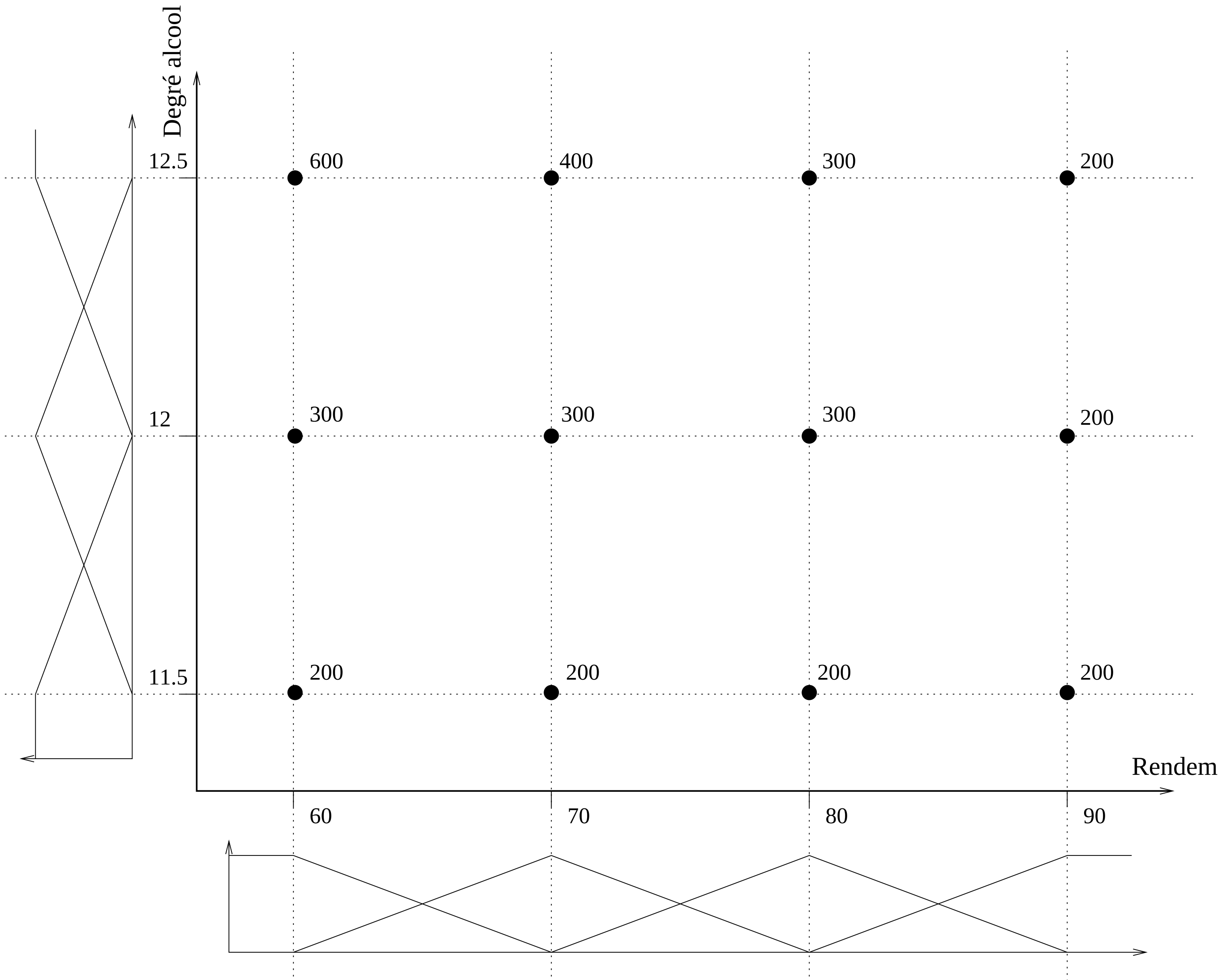


FIG. 6 – Rémunération des coopérants

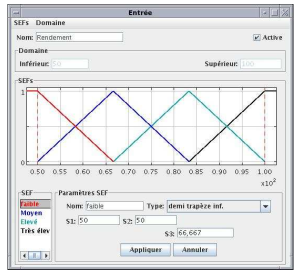
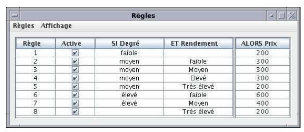


FIG. 7 – Variable rendement



Règle	Active	SI Degré	ET Rendement	ALORS Prix
1	✔	faible		200
2	✔	moyen	faible	300
3	✔	moyen	Moyen	300
4	✔	moyen	Elevé	300
5	✔	moyen	Très élevé	200
6	✔	élevé	faible	600
7	✔	élevé	Moyen	400
8	✔		Très élevé	200

FIG. 8 – Base de règles

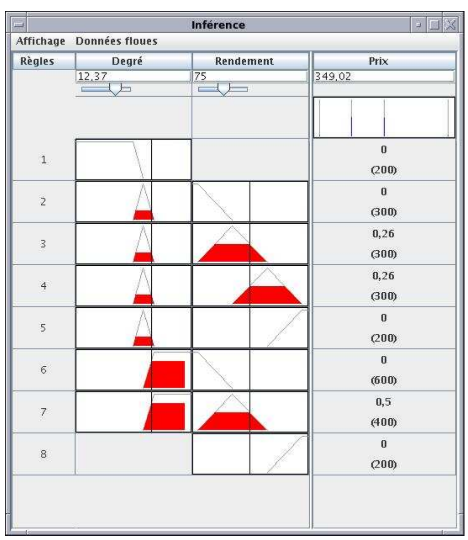


FIG. 9 – Inférence sur la rémunération des coopérants

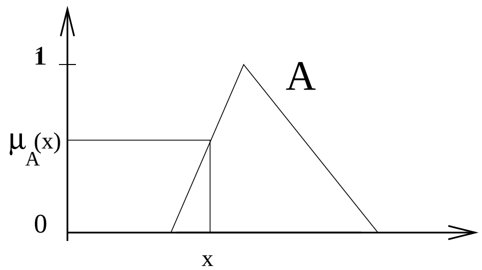


FIG. 10 – Un ensemble flou de forme triangulaire

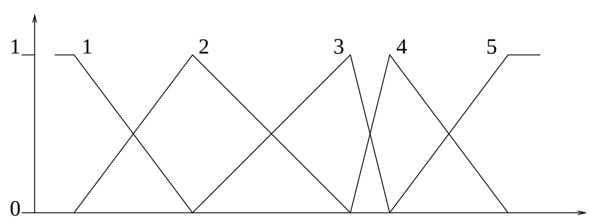


FIG. 11 – Exemple de partition floue forte

FIG. 12 – Un système d'inférence floue

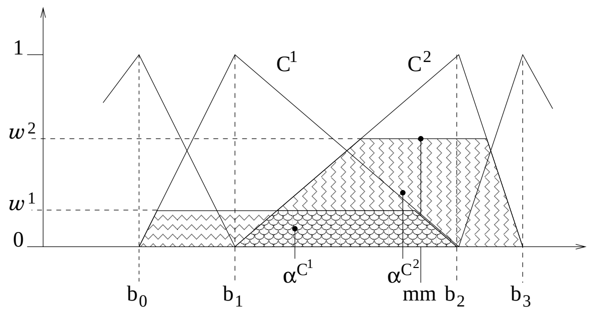


FIG. 13 – Un exemple de défuzzification

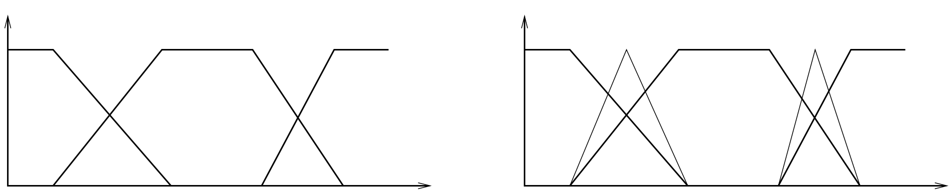


FIG. 14 – Partitions floue forte et quasi forte équivalentes