Starting with FisPro

fispro@supagro.inra.fr

FisPro (*Fuzzy Inference System Professional*) allows to create fuzzy inference systems and to use them for reasoning purposes, especially for simulating a physical or biological system. Fuzzy inference systems are briefly described in the fuzzy logic glossary given in the user documentation. They are based on fuzzy rules, which have a good capability for managing progressive phenomenons.

First of all, the FisPro implementation allows to design fuzzy systems from the expert knowledge available in a given field, for instance in winemaking. This approach is illustrated by an example given in the user guide *Quickstart with FisPro*.

FisPro also allows the complete design of a fuzzy inference system from the numerical data related to the problem under study. Many automatic learning methods unfortunately lead to "black box" systems. In FisPro, constraints are imposed to the algorithms to make the reasoning rules easy to interpret([1]), so that the user understands how the fuzzy system operates. This novel approach is one of the originalities of the software. Some examples are given in the user guide *Induction with FisPro*.

Both approaches, expert rule design and automatic induction, can be combined to create more complete and better performing systems. FisPro offers educational tools that illustrate the reasoning mechanism, and other tools to measure the system performance on datasets.

This software is made of two distinct parts: a C++ function library, which can be used independently, and a graphical Java interface, which implements most functionalities if the C++ library. It is portable, and can run on most existing platforms.

Authors

- Design and C++ implementation:
  Serge GUILLAUME, Irstea, UMR ITAP
  (`http://ser.gui.free.fr/homepage`),
  Brigitte CHARNOMORDIC, INRA, UMR MISTEA (`http://www.inra.fr`)
- Java interface : Jean-Luc LABLEE, Irstea, UMR ITAP
- Contributions:
  - C++
    - François OLIVIER, optimization module, from the work by Pierre-Yves GLORENNEC, *Algorithmes d'apprentissage pour systèmes d'inférence floue*, Edns. Hermes, 1999
    - Sébastien DESTERCKE, orthogonal least squares rule induction (ols)
    - Vincent THERRY, with the support of the Envilys company, cascading systems (superfis)
    - Russel STANDISH, optimized and parallelisable functions (OPENMP standard)
    - Hazaël JONES, design of the *implicative systems* part
    - Lydie DESPERBEN, C++ implementation of the *implicative systems* part
  - Java
    - Pierre-Marie BOYER, JNI interface (Java-C++)
    - Mathieu GRELIER, data visualization
    - Anne TIREAU, java interface of the *implicative systems* part.
  - Communication
    - Jean-Michel FATOU, FisPro icons and web site design
    - Moacir Jr PEDROSO, Portuguese translation of the interface
    - Juan Luis CORTI, Spanish translation of the *Quickstart with FisPro* manual

Acknowledgments

The development of the first version of FisPro was sustained by public funds from the French government and from Languedoc-Roussillon regional funds, during a research project, COST 2000-012, coordinated by the TRANSFERTS LR

association
`http://www.transferts-lr.org` with the industrial partnership of the *Cave Cooperative "La Malepère", Arzens, Aude*.

FisPro is an *open source* software, available on the Internet at:

`http://www.inra.fr/mia/M/fispro/`

Elementary notions

A system is also called a FIS, or fuzzy inference system.

The abbreviation MF is used for membership function or fuzzy set (see fuzzy logic glossary, section 3).

When starting FisPro, no system is available.

You have a choice between opening an existing system, or creating a new one, either automatically from data, or by hand, element by element.

This introductory guide explains the way to do it in this last case, suited to expert rule definition.

Notes:

- For numerical input, the decimal separator is the point (.).
- Any editing, new input, new output or new MF, is immediately taken in account in the system. Intermediate pop-up windows can be closed without loosing modifications.
- Some options are context dependent. When unavailable, options are grayed out in the menus.
- Expert definition only uses the *FIS* menu . The *Learning* menu is for more advanced needs (automatic rule induction).
- The *Data* menu allows to open an external data file in text format, to visualize data and to perform batch inference.

Note:

The *Language* option of the *Options* menu allows to choose the language you want for messages and menus.

Contents

1 Create a simple system

The first example creates a very simple system: 1 input, 1 output and 3 rules.

The input is the wine degree, the output is the price paid to the winegrower. The rules make the price change in function of the degree.

- First choose the *New* option in the *FIS* menu.
- The default name *New FIS* appears in the *Name* field. It is an editable text field. Rename it as *coop*.
- The conjunction is the operator used for combining MFs in the rule premise. The default operator is the product.

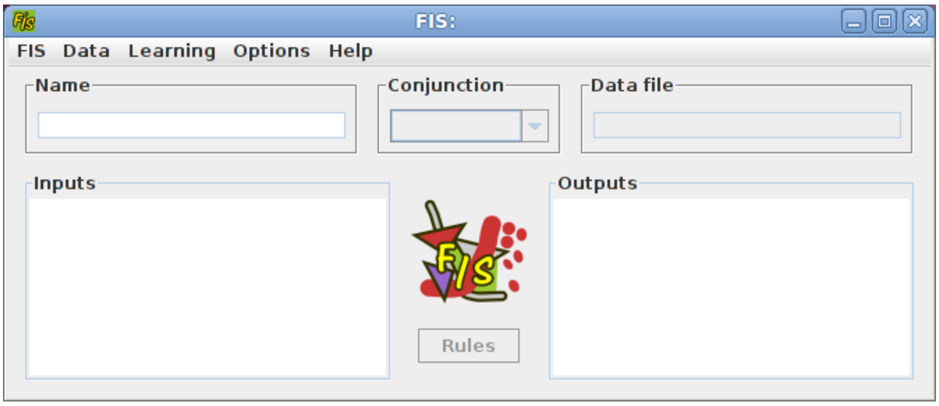


Figure 1: Main window in FisPro

1.1 Define a new input

To add a new input, use either the *New Input* option of the *FIS* menu, or a right mouse click in the *Input* area of the main window. The *Input* window appears.

An input is characterized by its range and its fuzzy partition. The fuzzy partition means the fuzzy sets that describe the input.

It can be active (default) or not.

Rename it *Degree*.

- The default input range is [0,1].
  To change it, select the *Range* menu in the *Input* Window, and enter the new range values: here 9 and 14.

- The easiest way to define a partition is the *Regular Grid* option of the *MF* menu, with the number of MFs corresponding to the number of wanted linguistic terms (default is 3 terms).
- The MFs are displayed in the lower half of the window : semi trapezoidal MFs at range bounds, and triangles elsewhere.
- For FIS clarity, MF names are important as they appear in the rules.

  We now give meaningful names and adjust the vertex location of each MF

  - MF 1 : name *Low*, vertices 9, 11,5 and 12
  - MF 2 : name *Average*, vertices 11,5, 12 and 12,5
  - MF 3 : name *High*, vertices 12, 12,5 and 14

  We obtain the partition given in figure 2.

1.2 Edit an input or an output

To change an input or output, double click on its name in the main window.

1.3 Define a new output

To add an output, use either the *New Output* option of the *FIS* menu, or a right mouse click in the *Output* area of the main window. The *Output* popup appears.

Rename it *Price*. An output is mainly characterized by its range and its nature: crisp output or fuzzy output. The output nature influences the fuzzy inference mechanism:

- With a crisp output, the rule conclusion can be any numerical value.
- With a fuzzy output, the rule conclusion can only be the linguistic term associated with an output MF, for instance *Low*, *Average*, *High*.

Indepedently of the output nature, the inference result is a numerical value.

Other parameters:

- Default value: it is the value of the inference result for this output, if no rule is fired by the inference.

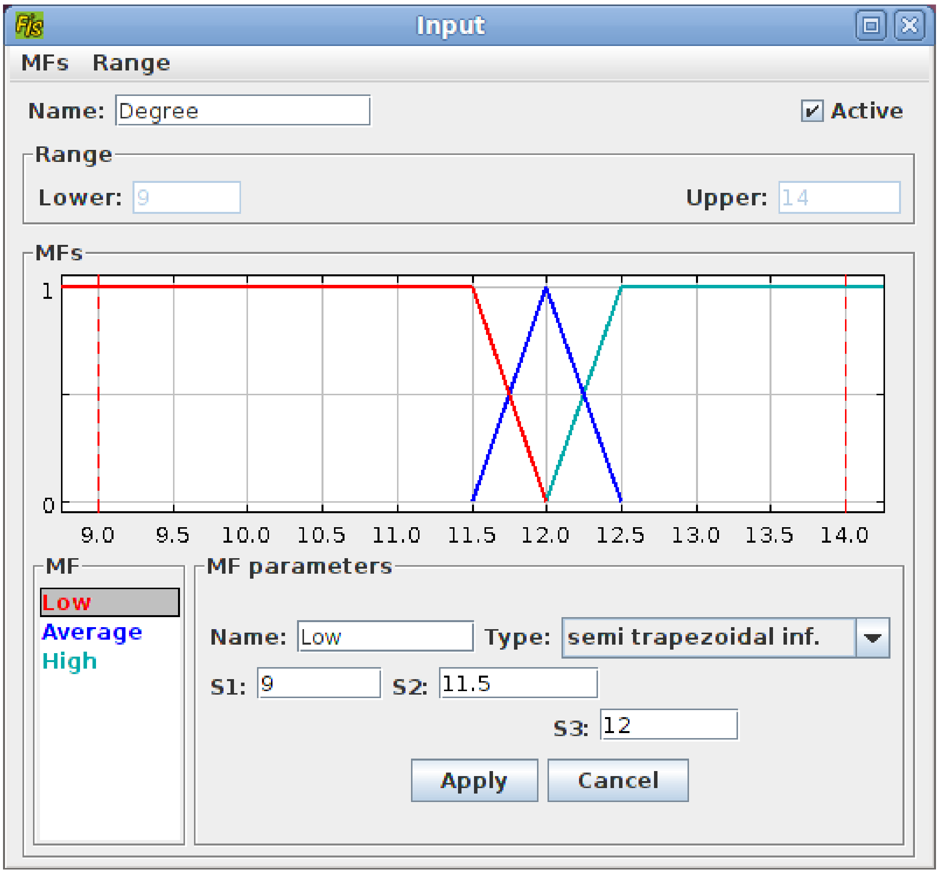


Figure 2: Input definition in FisPro

- Defuzzification and disjunction : choices linked to the way of aggregating the rule conclusions (see 3).
- classif: check this chekbox to round off the inference result to the closest class value (discrete value), for a crisp output. The possible classes are restricted to the rule conclusion values.

  If the output is fuzzy, its fuzzy partition must be defined, in the same way than for an input.

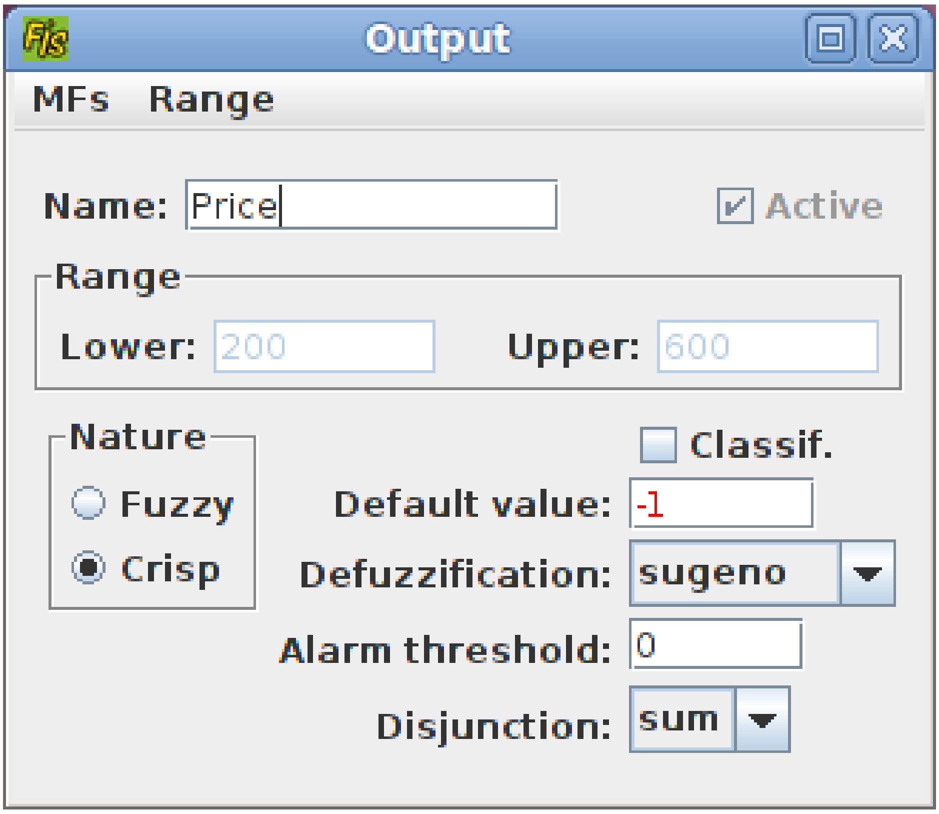


Figure 3: Defining a crisp output in FisPro

1.4 Defining a rule

To create rules, click on the *Rule* field in the main window. The *Rule* popup appears.

A rule is added by using either the *New Rule* option of the *FIS* menu, or a right click on the popup *Rule* column.

Click successively on each column, to select the linguistic term that will appear in the rule, or to enter a numerical output value (for crisp outputs only).

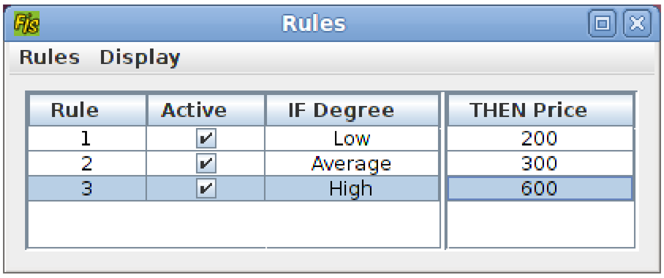


Figure 4: Defining rules in FisPro

1.5 Infer

The *Infer* option of the *FIS* menu graphically shows the inference mechanism. The input values are entered directly or by moving a cursor within each variable range. The inferred output value is displayed, together with several intermediate values, which allow to understand the different stages of the fuzzy reasoning:

For each rule:

- membership degree of each input value to each MF that appears in the rule premise. It is shown as a filled area ratio.

- rule firing strength, or matching degree.

  The matching degree is, in our particular case, equal to the input membership degree, as the system has a single input. In more complex cases, it is obtained by combining the MFs present in the rule premise.

  It is shown as a numerical value if the output is crisp, or as a filled area ratio if the output is fuzzy.

For each output:

- The inferred value is displayed on top, at the right, *480* here.

The *Price* output being crisp, with a Sugeno defuzzification and a sum aggregation, the inferred value is simply a weighted average of the rule conclusions, where the weights are the rule matching degrees.

For any degree in the *Degree* input range, between 9 and 14, we get the corresponding price.

The price progression is smooth, due to the interpolating capacities of the FIS.

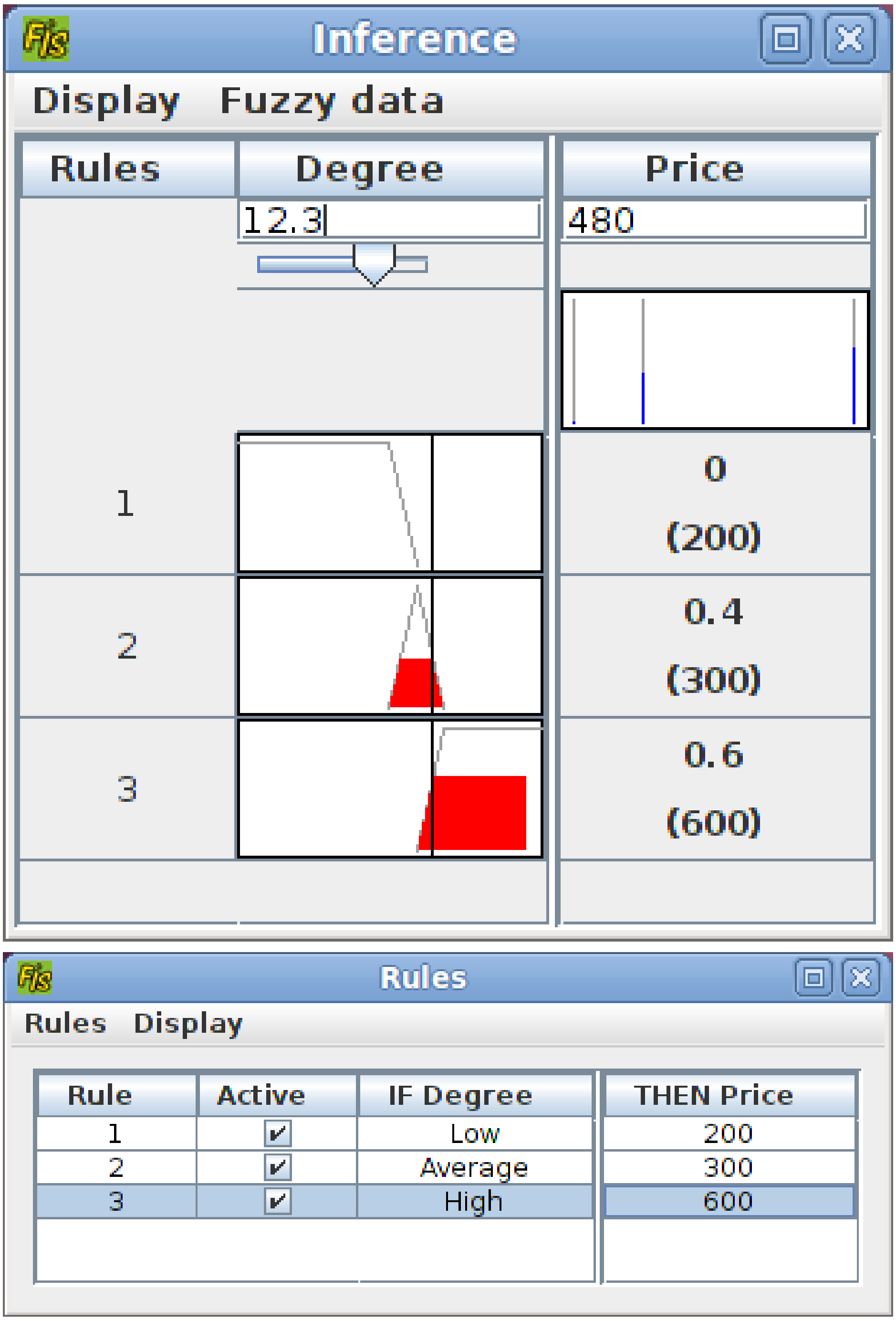


Figure 5: Fuzzy inference in FisPro

2 A more complex system

We will build a more realistic system from the first one, by adding a variable and modifying the rules to take both variables into account.

The extra input is the parcel yield.

Our objective is to reproduce the reasoning synthesized on figure 2.

Price must go down when yield increases, and go up when the alcohoolic degree increases. Below a given degree, or above a given yield, price is set to a fixed value.

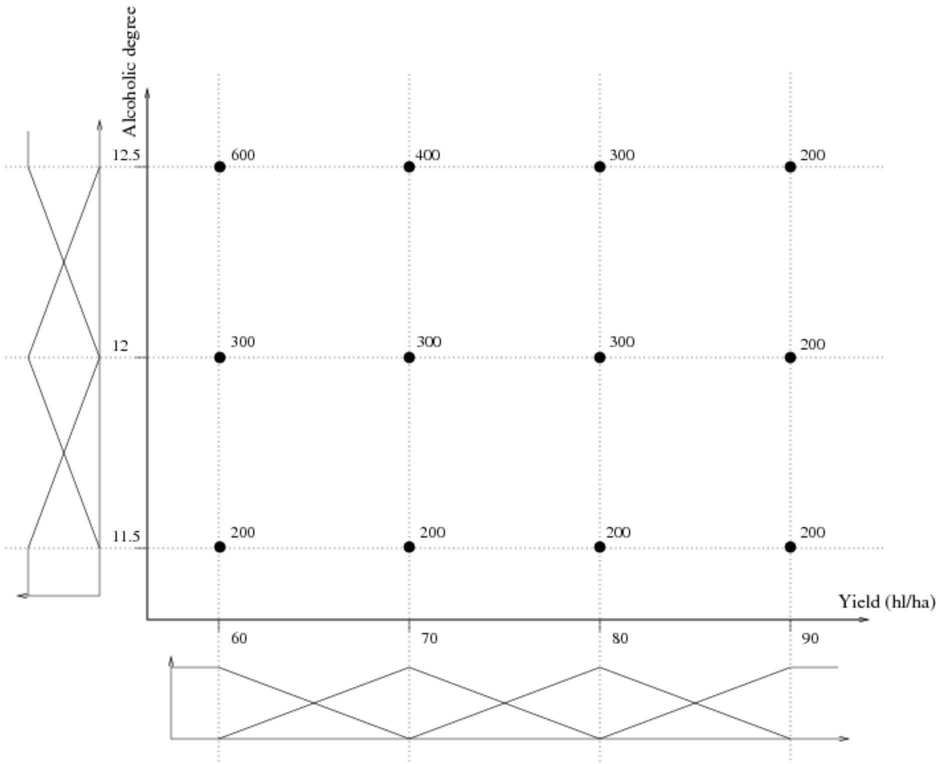


Figure 6: Cooperator remuneration

2.1 The yield variable

To add the *Yield* input, do as indicated in section *Defining an input* (1.1).

Set the range as [50,100], and build a regular partition grid with 4 MFs, with respective names: *Low, Average, High and Very High*.

You will get the partition represented in figure 7.

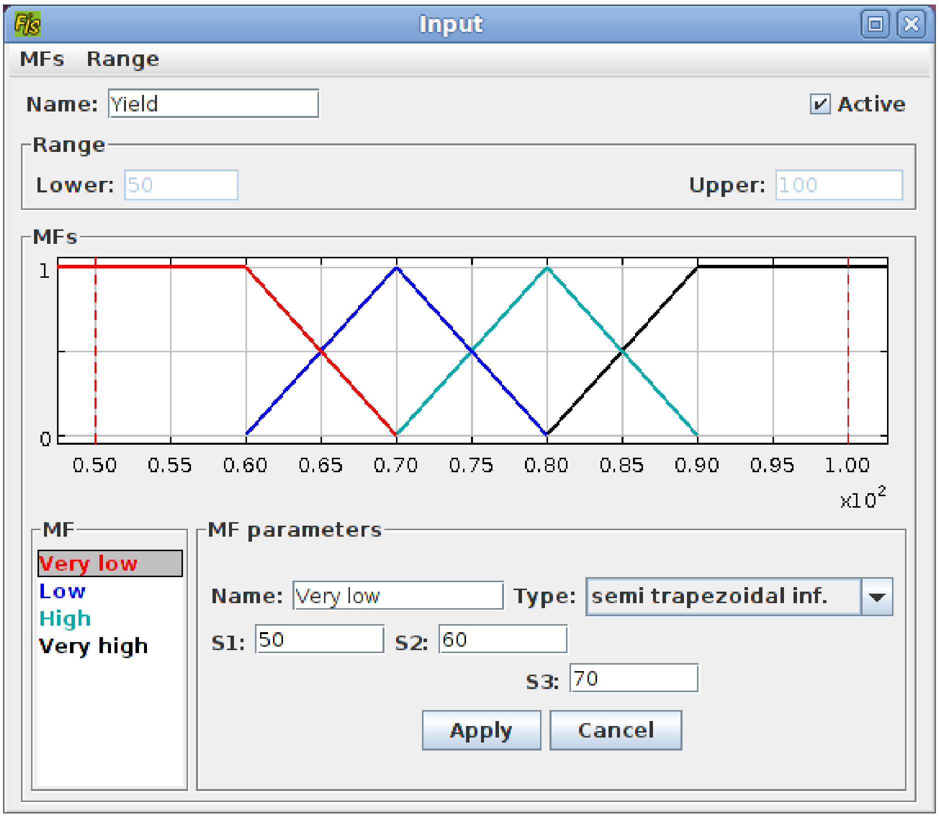
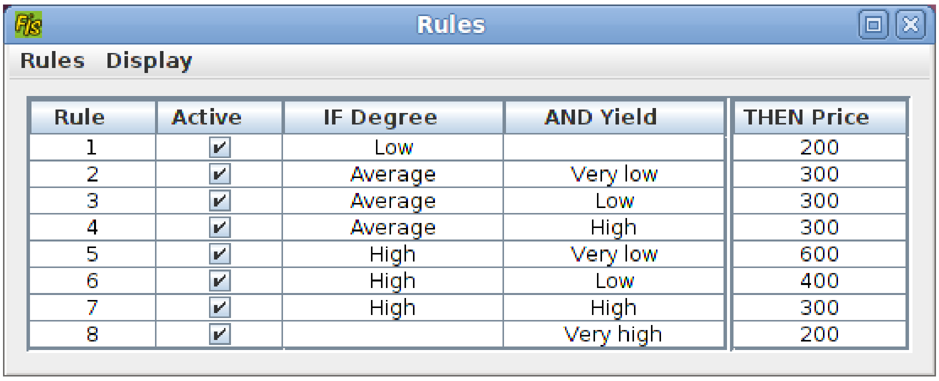


Figure 7: Partition for the Yield input

2.2 Generate the rules

We wish to write the rule base represented in figure 8.



Rule	Active	IF Degree	AND Yield	THEN Price
1	✓	Low		200
2	✓	Average	Very low	300
3	✓	Average	Low	300
4	✓	Average	High	300
5	✓	High	Very low	600
6	✓	High	Low	400
7	✓	High	High	300
8	✓		Very high	200

Figure 8: Rule base

The easiest way is to automatically generate the rule premise, using the *Generate rules* option in the *FIS* menu.

This option generates the rules corresponding to all possible combinations of the input MF variables: *Degree* and *Yield*, 4x3=12 rules in our case. Rule conclusions are initialized with a zero value, and must be modified.

Some of the generated rules can be simplified. This concerns *Low* degrees and *very high* yields. First remove the useless rules, then enter the rule conclusions in the *Price* column, as given in figure 2.

2.3 View the inference results

Use the *Infer* option in the *FIS* menu, and change the variable values in turn. The behavior of the fuzzy system is as expected and easy to understand.

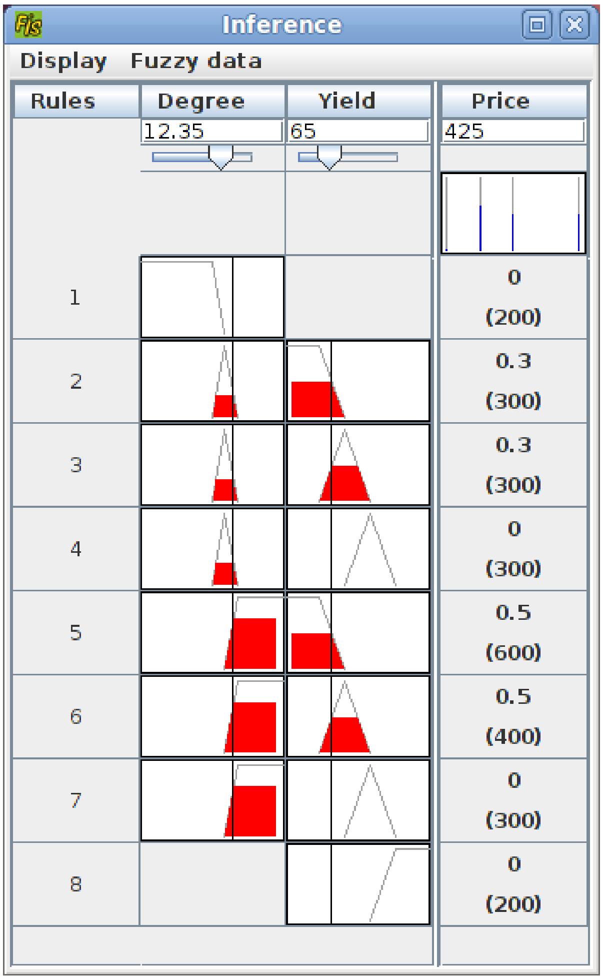


Figure 9: Inference for remunerating cooperators

3 Elementary fuzzy logic glossary

- fuzzy set : A fuzzy set is defined by its membership function. A point in the universe, x, belongs to a fuzzy set, A with a membership degree, $0 \leq \mu_A(x) \leq 1$.

  Figure 10 shows a triangle membership function.

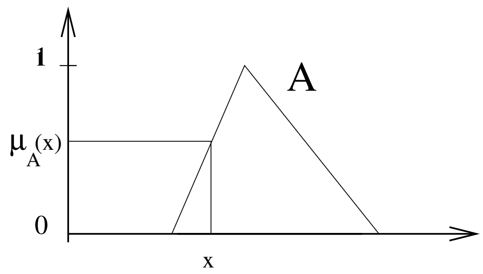


Figure 10: A triangle membership function

- Fuzzy set prototype: a point is a fuzzy set prototype if its membership degree is equal to 1.

- Operators:
  - AND : conjunction operator, denoted $\wedge$, the most common operators are minimum and product.
  - OR : disjunction operator, the most common are maximum and sum.
  - IS : the relation $x\ is\ A$ is quantified by the membership degree of x in the fuzzy set A.

- Partitioning: Partitioning is the definition of the fuzzy sets for a variable definition range. These sets are denoted A_1, A_2, ...

- Standardized fuzzy partition: a fuzzy partition of the X_i variable is called a standardized fuzzy partition if $\forall x \in X_i, \sum_j \mu_{A_j^i}(x) = 1$.

- Item : an item or individual is composed of a p-dimensional input vector x, and eventually of a q-dimensional output vector.

- Fuzzy rule: A fuzzy rule is written as ***If*** *situation* ***Then*** *conclusion*. The situation, called rule premise or antecedent, is defined as a combination of relations such as $x\ is\ A$ for each component of the input vector. The conclusion part is called consequency or conclusion.

- There are two main types of fuzzy rules:

  1. Mamdani type. The rule conclusion is a fuzzy set.
     The rule is written as:

     $$IF\ x_1\ is\ A_1^i\ AND\ x_2\ is\ A_2^i\ \ldots\ AND\ x_p\ is\ A_p^i$$
     $$THEN\ y_1\ is\ C_1^i\ \ldots\ AND\ y_q\ is\ C_q^i$$

     where $A_j^i\ and\ C_j^i$ are fuzzy sets defining the iput and output space partitioning.

  2. Takagi-Sugeno type. The rule conclusion is a crisp value.
     The conclusion of the i_{th} rule for the j_{th} output is calculated as a linear function of the input values: $y_j^i = b_{jo}^i + b_{j1}^i x_1 + b_{j2}^i x_2 + \cdots + b_{jp}^i x_p$, also denoted $y_j^i = f_j^i(x)$.

- Uncomplete rule: A fuzzy rule is said to be uncomplete if its premise is defined by a subset of the input variables. For instance, the rule

  $$IF\ x_2\ is\ A_2^1\ THEN\ y\ is\ C_2$$

  is an uncomplete rule, as the variable x_1 does not appear in its premise. Expert rules are generally uncomplete rules. Formally an uncomplete rule can be rewritten as an implicit combination of logical connectors AND and OR operating on all the variables. If the universe of the variable x_1 is split into three fuzzy sets, the above rule can also be written as:

  $$IF\ \ (x_1\ is\ A_1^1\ OR\ x_1\ is\ A_1^2\ OR\ x_1\ is\ A_1^3)\ AND\ x_2\ is\ A_2^1\ THEN\ y\ is\ C_2.$$

- Matching degree: For a given data item and a given rule, the rule matching degree, or weight, is denoted w. It is obtained by the conjunction of the premise elements: $w = \mu_{A_1^i}(x_1) \wedge \mu_{A_2^i}(x_2) \wedge \ldots \wedge \mu_{A_p^i}(x_p)$, where $\mu_{A_j^i}(x_j)$ is the membership degree of the x_j value in the fuzzy set A_j^i.

- Activation: An item is said to activate a rule, if the rule matching degree for the item is greater than zero.

- Rule prototype: an item is a rule prototype if the rule matching degree for this item is equal to 1.

- Fuzzy inference system (FIS): A fuzzy inference system is composed of three blocks, as shown in Figure 11. The first block is the fuzzification block. It transforms numerical values into membership degrees in the different fuzzy sets of the partition. The second block is the inference engine, with the rule base. The third one implements the defuzzification stage if necessary. It yields a crisp value from the rule aggregation result. The number of rules in the FIS is denoted r.

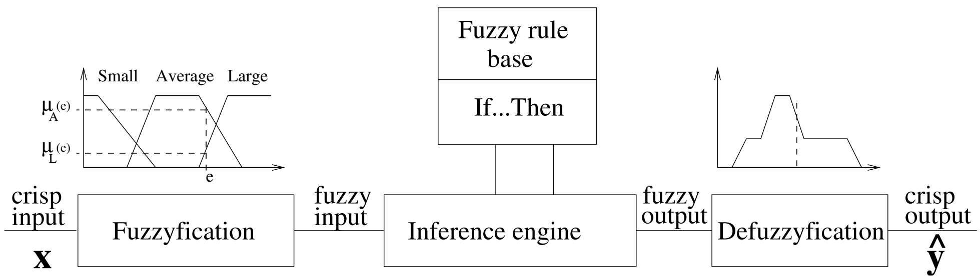


Figure 11: A fuzzy inference system

- System inferred output: denoted $\widehat{y_i}$ for the i_{th} item. The inferred value, for a given input, depends both on the rule aggregation and the defuzzification operators.

  Rule aggregation is done in a disjunctive way, meaning that each rule opens a possible range for the output. The two main operators are the maximun and the sum. The resulting levels are, r being the number of rules and m the number of labels in the output partition:

- max: $\forall j = 1, \ldots, m \qquad W^j = \left\{ \underset{r}{max} \left(w^r(x) \right) \mid C^r = j \right\}$
- sum: $\forall j = 1, \ldots, m \qquad W^j = min \left(1, \left\{ \sum_r \left(w^r(x) \right) \mid C^r = j \right\} \right)$

Several defuzzification operators are available. Figure 12 illustrates the process, when two labels have a non null resulting level, for two main kinds of operators.

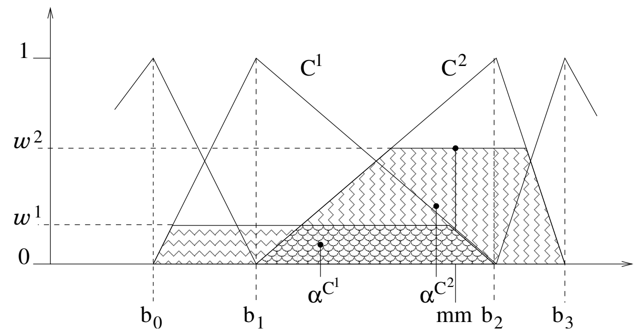


Figure 12: Area defuzzification

The inferred system output for the ith example is noted $\widehat{y}_i$.

The *mean of maxima* operator yields $\widehat{y}_i = mm$. This operator only considers the segment defined by the maximum level. It mainly works within a single linguistic label. Others similar outputs are possible, for example the minimum value of the maximum level or the maximum one.

The *weighted area* technique favors interpolation between linguistic terms. The output is equal to:

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

where m is the number of fuzzy sets in the partition, $\alpha = W^j$ is the resulting level of the jth fuzzy set, α^{C^j} is the x-coordinate of C^j_α centroid, and C^j_α is a new fuzzy set, defined from C^j as:

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

- Supervised learning: It induces an input-output mapping from a data set, called learning data set. It is usually limited to a MISO (multiple input single output) system. The learning data set includes n items.

- Mean square error: denoted MSE, it is equal to: $MSE = \frac{1}{n}\sum_{i=1}^{n} \|\widehat{y_i} - y_i\|^2$

- Mean error: Contrary to the previous one, it is homogeneous to a data item. It is expressed as

$$ME = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \|\widehat{y_i} - y_i\|^2} = \frac{\sqrt{MSE}}{\sqrt{n}} \tag{2}$$

References

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