Fuzzy inference systems: An integrated modeling environment for collaboration between expert knowledge and data using FisPro

Serge Guillaume [a,*], Brigitte Charnomordic [b]

[a] Irstea, UMR ITAP, BP 5095, 34196 Montpellier, France
[b] INRA-SUPAGRO, UMR 729 MISTEA, F-34060 Montpellier, France

ARTICLE INFO

ABSTRACT

The present paper aims to demonstrate the interest of fuzzy inference systems in system modeling when human interaction is important. It discusses the originality of FIS and their capability to integrate expertise and rule learning from data into a single framework, analyzing their place relatively to concurrent approaches. An open source software implementation is presented, with a focus on the useful features for modeling. Two real world case studies are presented to illustrate the approach and the software utility.

1. Introduction

Fuzzy logic, since the pioneer work by Zadeh (1965), has proved to be a powerful interface between symbolic and numerical spaces, especially by the means of the linguistic variable concept (Zadeh, 1975).

Forty years later, fuzzy inference systems (FIS) have become one of the most famous applications of fuzzy logic. One of the reasons for this success is the ability of fuzzy systems to incorporate human expert knowledge with its nuances, as well as to express the behavior of the system in an interpretable way for humans.

Historically, the first kind of fuzzy rule based systems focused on the ability of fuzzy logic to model natural language (Mamdani & Assilian, 1975).

A second approach, proposed in the mid eighties (Takagi & Sugeno, 1985), was based on automatic learning from data. As discussed by Hullermeier (2005), fuzzy methods in machine learning and data mining have been thoroughly investigated and proved useful.

With these new developments, the age of expert knowledge-based fuzzy systems was over, and the data driven rule generation methods played the main role in fuzzy system design, moving away from the philosophy of fuzzy set theory initially directed to bridge the gap between human reasoning and machine processing, which can be summarized as *Computing with words*.

This drift has been clearly stated by Dubois and Prade (1997): "Fuzzy controllers, and fuzzy rule-based modeling which have become the most popular and visible side of applied fuzzy set theory, are only the emerged part of the fuzzy iceberg, and as time passes this technology seems to owe less and less to fuzzy set theory itself, and mainly becomes a tool for approximating functions".

As a reaction, fuzzy logic researchers developed a new field of research. As shown in Guillaume (2001), the use of fuzzy formalism is not sufficient to ensure the interpretability of a knowledge base. Three conditions have to be fulfilled. First, semantic integrity should be respected within the partition. Secondly, the number of rules should be small. The third condition is specific to complex systems with a large number of input variables: rules must not systematically include all input variables, but only the important ones in the rule context. This kind of rules is usually referred to as incomplete rules. As the interpretability constraints may conflict with the numerical error minimization objective of automatic learning methods, several works have been carried out to propose a trade off between interpretability and accuracy (Casillas, Cordón, Herrera, & Magdalena, 2003).

Though these efforts allow to propose FIS as an integrated framework for system modeling, to get the most out of expert knowledge and data, not enough attention as deserved has been paid to the potential interest of this approach. FIS are still mostly used as predictive models, their performance being evaluated in terms of accuracy, and the behavior of their inference engine is rarely discussed.

Operational, in a process/system context, denotes a working method or a philosophy that focuses principally on cause and effect relationships (or stimulus/response, behavior, etc.) of specific interest to a particular domain at a particular point in time. In this sense, a rule system is a good way to model a specific kind of knowledge, the so-called *operational* knowledge. Fuzzy concepts, whose content, value, or boundaries of application can vary according to context, operator and conditions, instead of being

---

* Corresponding author.
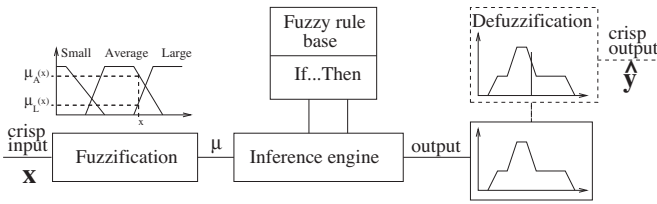   *E-mail address:* serge.guillaume@irstea.fr (S. Guillaume).

Fig. 1. A fuzzy inference system (FIS).

fixed once and for all, arise naturally in the operational approach of system modeling, hence the relevance of FIS for that matter.

The work presented here aims to be a contribution to this research trend, both asserting the need for fuzzy logic in system modeling, and proposing an advanced software for exploratory design and analysis of FIS, allowing to understand the system operation and its sensitivity.

Its objectives are (i) to discuss the interest of FIS as modeling tools with imposed interpretability constraints in methods and software implementation, (ii) to compare, using the case studies, the FIS with other models (e.g. statistical multiple regression) and to analyze two inference mechanisms: implicative rules and conjunctive rules. An open source software implementation of FIS design and optimization is proposed in *FisPro*,[1] which corresponds to ten years of research and software development in the field of learning interpretable FIS from data.

Detailed illustrations will be provided through two case studies of real world problems in Agriculture and Environmental modeling, a field where expert knowledge and data often complement each other.

The structure of the paper is as follows: Section 2 recalls FIS principles and analyses the specific role of fuzzy logic in system modeling. Section 3 explains the ways of cooperation between knowledge and data in a modeling approach based on FIS. A state of the art of fuzzy software and *FisPro* main features are presented in Section 4. The case studies are presented in Section 5. The first one deals with pesticide loss modeling in agricultural spraying, the second one illustrates an inference mechanism which, though little used, allows to model logical constraints, and in this way, is closer to classical logic than the inference mechanism used for fuzzy controllers. Finally a conclusion is given with some perspectives in Section 6.

2. Is there a need for fuzzy logic in system modeling?

In a recent paper, Zadeh (2008) himself asks the somewhat provoking question *Is there a need for fuzzy logic?* He gives general answers by highlighting some unrecognized features of fuzzy logic.

The current work discusses a special aspect of this same question centered on system modeling with FIS. A system is taken in the very general sense of an object including inputs, outputs and a model to map inputs onto outputs.

In this section we first recall the general FIS structure, then we examine some concurrent approaches before explaining the originality of FIS.

2.1. FIS structure

This section only focuses on some specific points useful for the proposed framework. For general details about FIS the reader may refer to Lee (1990) or Dubois and Prade (1996).

A typical fuzzy inference system is shown in Fig. 1.

2.1.1. Variable partitioning

The readability of fuzzy partitioning is a pre-requisite condition to build an interpretable rule base. The necessary conditions for interpretable fuzzy partitions have been studied by several authors Ruspini (1982), Valente de Oliveira (1999), and Glorennec (1999). Let us recall the main points:

- Distinguishability: Semantic integrity requires that the membership functions represent a linguistic concept and different from each other.
- A justifiable number of fuzzy sets.
- Coverage: Each data point, *x*, should belong significantly, $\mu(x) > \epsilon$, at least to one fuzzy set. ϵ is called the coverage level.
- Normalization: All the fuzzy sets should be normal.
- Overlapping: All the fuzzy sets should significantly overlap.

These requirements are all fulfilled by the strong fuzzy partitions, illustrated in Fig. 2, such as:

$$\begin{cases} \forall x & \sum_{f=1,2,\ldots,m} \mu_f(x) = 1 \\ \forall f & \exists x \quad \mu_f(x) = 1 \end{cases} \tag{1}$$

where *m* is the number of fuzzy sets in the partition and $\mu_f(x)$ in the membership degree of *x* in the *f*th fuzzy set. Eq. 1 means that any point belongs at most to two fuzzy sets when the fuzzy sets are convex.

2.1.2. Rule base

There exist different kinds of fuzzy rules (Dubois & Prade, 1996). The most common ones are called conjunctive rules and are used in Mamdani or Sugeno FIS. The underlying relationship between the input and output spaces is modeled using a conjunction operator (usually the minimum). They represent joint sets of possible input and output values.

The other kind of fuzzy rules (Dubois, Prade, & Ughetto, 2003; Weisbrod, 1998) models the input output relationship using a fuzzy implication, they are called implicative rules and generalize the classical logic rules. Their meaning is quite different from the conjunctive rules one: they encode constraints and each rule defines a fuzzy restriction on the set of possible values. The difference of nature between conjunctive and implicative rules impacts rule combination: while several conjunctive rules are combined disjunctively (as they widen the scope of a single rule), implicative rules are combined conjunctively, because several constraints lead to a more restricted feasible set of allowed situations than a single constraint.

Conjunctive rules are suitable to manage positive evidence, to extract relationships from data or to implement similarity reasoning, while implicative rules can be used to model knowledge, which can be seen as negative information in the sense that it points out forbidden values. A more detailed comparison between the two kinds of rules is included in Jones, Charnomordic, Dubois, and Guillaume (2009).

Implicative rules allow a non trivial generalization of the Modus Ponens (Martin-Clouaire, 1989).

The classical version, $A \wedge (A \rightarrow O) \models O$, becomes: $A' \wedge (A \rightarrow O) \models O'$.
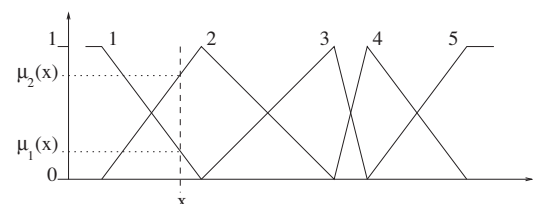

Fig. 2. A five linguistic term strong fuzzy partition.

2.1.3. Output distribution

The rule aggregation yields a possibility distribution representing the inferred output.

With conjunctive rules the output distribution is usually not normalized. A defuzzification stage is required and yields a crisp value from the output possibility distribution.

With implicative rules, the output distribution is normalized, and defuzzification is optional.

Both inference mechanisms are illustrated in the last case study of Section 5.

2.2. Concurrent approaches

As this work focuses on modeling, we will not address the question of fuzzy control in the present paper. Depending on the kind of model under design and on the available knowledge, FIS modeling may either compete or be used in a complementary way with other techniques.

Control systems are based on a deep knowledge of the process to be modeled. Whenever such a mathematical model exists, there is no need for approximate techniques to encode well established relationships.

Nevertheless fuzzy models may be useful for managing soft transitions between two steady-state points, like in industrial process supervision. In this case they are complementary of control theory.

Discrete event system formalisms (Petri nets, Grafcet, timed automata, etc.) are intrinsically designed to formalize the time dependence of a dynamic system, which at any given time has a state given by a multidimensional vector in an appropriate state space. FIS have no such characteristic, they provide a *static* view of a system at a given time, which can be included as part of a dynamic system.

FIS can also be compared to *statistical methods*. Both approaches are able to build models from a data sample and to take into account data imprecision or *imperfection*, as statistics is based on probability theory. Statistics proved to be efficient in a wide range of situations, and thanks to probabilistic assumptions, statistical models often yield a useful confidence interval.

Nevertheless, from a concurrent point of view, let us focus on some drawbacks. Statistical techniques require a large amount of data to produce significant results. Most of the techniques are limited to data management, only a few of them are able to take into account expert knowledge. Bayesian inference, by the means of prior distributions, and decision trees using *a priori* defined partitions, are the most popular ones.[2] The statistical models are often difficult to interpret: for instance it is problematic to give a meaning to some regression coefficients or to transformed variables. Zadeh (2008) advocates that most of the theories can be generalized using the fuzzy formalism. This also applies to statistics. A large effort within the Fuzzy Logic research community has been dedicated to this topic with more or less convincing results. The fuzzy generalization of the well known k-means clustering algorithm, called fuzzy *c*-means (Bezdek, 1981) is a valuable example.

Statistics can also be used in cooperation with other modeling techniques, including FIS, especially in data pre-processing: univariate or multivariate analysis, outlier detection, for instance.

Artificial Intelligence provides many knowledge representation techniques (such as graphs or ontologies) or data mining methods (such as association rules). How can FIS, used as a modeling framework integrating knowledge and data, be compared with them?

---

[2] Few papers in the literature use this opportunity, in almost all cases prior distributions are estimated from data and partition bounds are automatically generated.

According to Dubois and Prade (2003): "An important part of the concern and research in fuzzy logic and possibility theory does focus on issues such as knowledge representation, approximate reasoning and reasoning under uncertainty, which are central to artificial intelligence".

Without going into detail, let us point out that knowledge representation methods intrinsically handle symbolic knowledge, and data mining tools are designed for coping with data. A few bridges exist between these two topics of research (Adomavicius & Tuzhilin, 2001; Ling, Kang, Johns, Walls, & Bindoff, 2008; Sester, 2000).

2.3. Originality of FIS

In the entirety of the modeling techniques, FIS are fit to be used for human machine cooperation as they provide a single framework (membership functions, rules, operators, inference mechanism,...) carrying a given semantics and allowing the correspondence between numerical and symbolic elements.

In our opinion, the main reason to employ FIS is the need of semantics. Many research papers do not tackle this point. They take for granted that the utilization of fuzzy logic yields an easy to use human interpretable model, even if they actually design data driven systems as function approximators. Other methods, particularly statistical models, are more fit to that purpose than FIS. Generally speaking, semantic is needed when the human being has to interact with the system. Knowledge formalization or training support tools require transparent models: the user is willing to easily analyze the underlying reasoning. Semantic is also needed when the system output is a symbolic label (instead of a numerical value), such as in quality evaluation or risk management.

As there is no underlying assumption about the data distribution, FIS automatic modeling does not yield a confidence interval as regression or other statistical methods do. Nevertheless, system sensitivity to input values can be assessed by simulation using noisy data, bootstrap, ...

FIS can manage expert knowledge and also be used for knowledge discovery. To take advantage of both kinds of knowledge, one should use them in their respective areas of excellence: the key idea is to keep the expert at the linguistic level and to use the data to define the numerical characteristics of the linguistic terms or to refine the rules.

3. Cooperation between knowledge and data

FIS can be entirely built from data, and many research papers address this question. For an approach to learn interpretable Fuzzy Inference Systems with *FisPro*, the reader can refer to Guillaume and Charnomordic (2011). The present paper focuses on the case when data are not sufficient, and must be complemented with expert knowledge (Guillaume & Magdalena, 2006), which has to be formalized. Experts know the main trends of the most influential variables in the system and are able to describe their behavior using linguistic rules. Expert rules are based on a large experience and present a high level of generalization. Dealing with complex systems, experts face two difficulties: how to define useful variables and how to formalize variable interactions. Data, the extensive form of induced knowledge, are basically pieces of positive evidence. The most important data characteristic is incompleteness: A data sample cannot pretend to cover all the possible situations, especially in the case of complex systems. Therefore it is a cautious attitude to consider any sample as part of a whole. The induced rule quality highly depends on the training set characteristics: the more representative of the whole the examples, the more general the induced rules.

Cooperation between knowledge and data is possible at various levels, as well in designing a model as in using it for prediction or decision support.

- Design: Expert knowledge and data are likely to cooperate in each step of the FIS design process, especially in input and output partitioning (Guillaume & Magdalena, 2006) and rule definition. Given the input/output partitioning, the rule base may include induced as well as expert rules. As an example, expert knowledge may serve to define linguistic terms or rules in areas where no data are available.
- FIS use: Either expert rule and model validation with data, or rule automatic generation and expert assessment of induced knowledge, is possible. In the first case, data are used to define the precise meaning, in the numerical space, of an expert linguistic concept while in the second one, experts give a linguistic meaning to rules induced from data, thanks to system interpretability. A dialectic motion between these two approaches may be beneficial in system modeling.

Fig. 3 shows the outline of the proposed approach for modeling with FIS, allowing to formalize and introduce expert knowledge at all steps: fuzzy partition and rule design, where knowledge can complement automatic design, by adding MF and rules in areas where no data are available, FIS parameter optimization and system validation. In this last step, it is proposed not only to check numerical accuracy, but also to analyze induced knowledge. The system generalization ability can be studied through automatic cross-validation procedures, and also analyzed by studying the areas of low performance and the links between data items and inference rules.

As the use of the fuzzy formalism increases system complexity (more parameters, operators ...), it has to be justified.

FIS based cooperation between expert knowledge and data is suitable when there exists prior knowledge. Expertise often expresses trends in purely linguistic terms, which have to be completed by data to tune the models. The interpretability constraints restrict this approach to low dimensional problems. From our experience, as in many cases there is not a unique system able to model the input–output relationships, several human–machine iterations are needed.

4. Software environments

Exploratory analysis is essential in a modeling approach associating expert knowledge and data. Fuzzy software does not always pay a lot of attention to that point.

In this section, we present a brief state of the art of available fuzzy software, and then we introduce *FisPro*. We describe its most important features, and put them in relation with the modeling approach proposed in the previous section.

4.1. State of the art regarding fuzzy software

Fuzzy software was first developed for the needs of fuzzy control, which became popular in the 1990s. Industrial software as well as academic one became available, and the targeted audience was control engineers, who used fuzzy software as an alternative in the domain of control system design. For industrial software, we can cite the TIL Shell (InfraLogic, 1995) by Togai InfraLogic and Siemens, FIDE by Aptronics (2007) and FuzzyTECH by Inform (GmbH, 2007). Toolboxes were proposed for Matlab, the Fuzzy Logic Toolbox developed by Jang (MathWorks, 2007), the FMID written by Babuška (2003), and Floulib designed by Foulloy (2005). A fuzzy logic add-on is also available for Mathematica (Research, 2007).

As accuracy was crucial in the domain of control systems, FIS optimization and automatic generation soon became available in fuzzy control software and provided function approximation capabilities through automatic learning procedures based on neural networks or genetic algorithms. As a matter of fact, interpretability was not of prime concern in this field. The main point for control system users and designers was the interpolation capability of fuzzy inference systems. Therefore the systems built using automatic learning were not analyzed in regard to their interpretability, though that same interpretability was presumed true and often put forward as an incentive for using fuzzy logic.

Since the end of the 1990s, fuzzy logic expanded to other fields of interest, and more general software was designed by researchers from the fuzzy logic community, with the objective to attract a new audience through data analysis and decision support. It includes simple fuzzy inference engines as FuzzyClips (Orchard, 2004) or Sazonov's specific learning sofware (Sazonov, 2002), fuzzy decision trees (Janikow, 2005), DataEngine from MIT GmbH (GmbH, 2001), or neuro fuzzy classification (Kruse, 2007).

During the special session on software for soft computing that was organized at the 2007 FuzzIEEE conference (IEEE, 2007), some advanced software projects were presented, as FrIDA (Borgelt & Gonzáles-Rodríguez, 2007), a free intelligent data analysis toolbox, or Xfuzzy (Baturone, 2007). Xfuzzy is a development environment, that integrates a set of tools to help the user to cover the several stages involved in the design process of fuzzy logic-based inference systems. The conference also includes a comprehensive review of
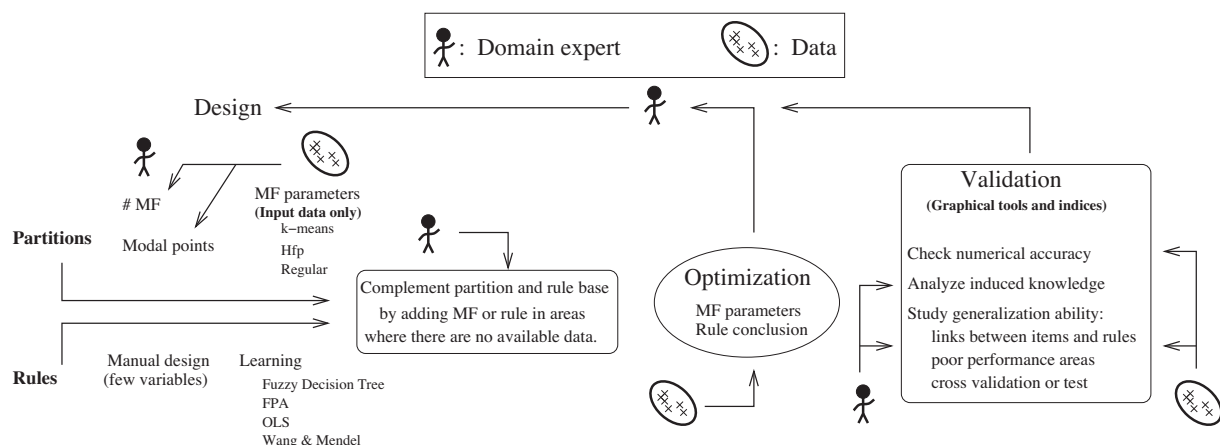

Fig. 3. Outline of an integrated approach for modeling with FIS.

fuzzy software, an interesting discussion of useful features, and a call for building a fuzzy tool kit that supports the take-up of fuzzy systems in business applications (Nauck, 2007).

Potential end users may have very different profiles: Fuzzy Logic researcher, adviser in Data Analysis, researcher in Biology or Economics needing system modeling tools, domain expert in the Food Industry looking for an expert knowledge transmission toolkit, student making a project in Artificial Intelligence...

According to their profile, users have various requirements concerning fuzzy system structure, available methods and interface. However, one common denominator is probably the need for automatically generated systems and for the interpretability of these systems. Indeed, users currently interested in fuzzy software wish to tackle complex systems, which are extremely difficult to write from scratch. Thus they typically need to complete available expert knowledge with knowledge acquired from data. But that is of little interest if the software gives results which cannot be interpreted and discussed with others.

Among fuzzy software products, *FisPro* stands out because of the interpretability of fuzzy systems automatically learnt from data. Interpretability is guaranteed in each step of the FIS design with *FisPro*: variable partitioning, rule induction, optimization.

4.2. Linguistic variable and fuzzy partitioning

Fuzzy partitioning is an important aspect of FIS design.

Even if, in interactive design, other MF shapes are available and fuzzy partitions can be freely adjusted, *FisPro* automatic procedures systematically generate strong fuzzy partitions for inputs, with semi-trapezoidal shapes at the edges and either triangular or trapezoidal shaped MFs elsewhere, as shown in Fig. 2.

Using strong fuzzy partitions ensures semantic integrity. Moreover, compared to partitions made up of unbounded membership functions, such as the widely used Gaussian ones, they also lead to more robust systems, where the number of simultaneously fired rules is limited and kept small (Destercke, Guillaume, & Charnomordic, 2007).

4.3. Rule learning

When rule learning mechanisms are available, it is important to guarantee the interpretability of the learnt rule bases. For that purpose, the same linguistic concepts, therefore the same fuzzy sets, should be shared by all of the rules whatever the rule induction method.

In *FisPro*, all of the rule induction methods use predefined linguistic labels. None of them is allowed to tune the labels nor to add a new concept (fuzzy set) to the partition.

In order to permit fuzzy set sharing and interpretability, some methods well known in fuzzy learning such as Wang and Mendel (1992b) and Orthogonal least squares (OLS) (Chen, Billings, & Luo, 1989; Chen, Cowan, & Grant, 1991; Hohensohn & Mendel, 1994; Wang & Mendel, 1992a) have been revisited. Fuzzy decision trees (Breiman, Friedman, Olshen, & Stone, 1984; Quinlan, 1986; Weber, 1992) are available and meet the interpretability criteria.

The implementation of an interpretable fuzzy OLS is presented in (Destercke et al., 2007). The key idea, which is valid for all these revisited methods, is the use of predefined strong fuzzy partitions for the rule generation, instead of the data based Gaussian membership functions used in the original OLS. Besides the usual learning passes, an extra one is proposed to reduce the number of rule conclusions with a user defined loss of accuracy.

New methods, such as hierarchical fuzzy partitioning (Guillaume & Charnomordic, 2004), have been implemented with the same preoccupation of fuzzy set readability. They generate hierarchical embedded partitions and make FIS refinement possible.

Similarly, FIS optimization (including simplification) is available in *FisPro* and allows tuning respectful of linguistic interpretability.

To increase the FIS robustness and generalization capabilities, rule learning is done based on sampling procedures using learning and test sets, and guided by evaluation indices.

4.4. Inference mechanism

As discussed in Section 2.1.2, there exist different kinds of fuzzy rule bases: conjunctive ones and implicative ones.

Let us consider an interesting property, called inferential independence. Let RB be a consistent rule base. In Boolean logic, when a rule premise is true for a given multidimensional input, the value inferred by RB is that rule conclusion, independently of the presence and content of the other rules. Although this is not usually pointed out, this property does not hold for a fuzzy conjunctive rule base. Given a multidimensional fuzzy input strictly identical to the fuzzy rule premise, overlapping membership functions trigger more than one rule and the rule base inference result is computed as the union of the rule conclusions weighted by the respective matching degrees. With implicative rules and strong fuzzy partitions, the inferential independence property holds.

Each kind of rule base: conjunctive and implicative, has its pros and cons, see Jones et al. (2009) for a detailed comparison. In *FisPro*, rule learning methods are implemented using conjunctive inference, and implicative rule bases are available for expert design. Graphical visualization may help to understand the differences between the inference mechanisms, and specific tools have been developed.

4.5. A friendly interface

Only the most original and important features of *FisPro* friendly interface are described in this paper. The user documentation available on line describes all features in detail.

For an efficient approach in exploratory analysis and system modeling, special attention has been put on the dynamical behavior of a FIS following user modifications.

- Each variable or rule can be activated/deactivated within the fuzzy system. Fuzzy partitions can be edited and the fuzzy system operators modified. The current data file is displayed in a table, and each data row can also be enabled/disabled. All operations are dynamically handled and all current windows are updated, including the inference result ones.
- To help the user to assess the rule representativeness, an option that evaluates the *links between rules and examples* is available. An accessible detailed cross-summary gives for each rule, the samples that fire this rule above a given matching degree, and for each sample, the rules that are fired.
- Inference can be done manually or on the current data file, with evaluation criteria which take into account the numerical accuracy as well as the significance of data items regarding the FIS. Response surfaces are also available for an exploratory analysis of the system behavior.

5. Case study

This section aims to illustrate the potential of fuzzy inference systems to deal with real world applications involving both data and expert knowledge.

The first study is a supervised learning case, and the second one is essentially based on expertise modeling. First we present the evaluation indices, that will be used in the case studies.

5.1. FIS evaluation indices

FIS evaluation indices are available in FisPro to assess the prediction capabilities of a FIS for a given dataset. These indices characterize the performance and coverage.

5.1.1. Coverage index

Examples are labeled active or inactive for a given rule base. An example is *active* if its maximum matching degree over all the rules is greater than a user defined threshold, *inactive* otherwise.

Following this definition, a *coverage index* is calculated by applying the formula $CI = \frac{A}{N}$ where A is the number of active examples, and N the file size. The *coverage index* is a quality index complementary to the classical accuracy index.

5.1.2. Performance indices

Two error indices are automatically computed, the mean absolute error:

$$MAE = \frac{1}{A} \sum_{i=1}^{A} |\widehat{y_i} - y_i|$$

and the maximum error $MaxError = \max_{i=1}^{A} |\widehat{y_i} - y_i|$.

5.2. Modeling pesticide loss

The modeling objective is to propose a relationship between the proportion of product lost in the atmosphere and some micro-meteorological variables, during vine spraying. This relationship can be modeled under various formalisms. The most common one is the multiple linear regression, but we show how fuzzy rules can also be used. A thorough description of both the problem statement and the results can be found in (Gil, Sinfort, Guillaume, Brunet, & Palagos, 2008). In this section, we compare the results of the multiple linear regression with the FIS modeling ones. FIS modeling will allow to incorporate domain knowledge in the fuzzy partitions, and to learn interpretable rules as an input–output relationship.

The spraying is achieved using air assisted devices to aid the transport of the droplets toward the target. Two sets of nozzles were used in the experiments, leading to two kinds of droplets called *Fine* and *Very fine*, with a respective Volume Median Diameter (VMD) of 134 and 65 μm.

The considered explanatory variables are the following ones:

- *W*: Wind speed (m/s)
- *T*: Air temperature (°C)
- ΔT: Wet bulb temperature depression (°C)
- *z/L*: Atmosphere stability parameter

In the present paper, the study is restricted to the *Fine* case. The sample size is 32 experiments.

5.2.1. Multiple linear regression approach

A stepwise multiple linear regression[3] is first carried out. This is the classical approach in this application domain. It yields the following model:

$$\widehat{Y_i} = 9.719 - 0.229(T) + 0.109(W * \Delta T)$$

characterized by a determination coefficient of $R^2 = 0.70$.

The plot in Fig. 4 shows that the fit is relatively satisfactory.

Nevertheless, such a predictive model is very difficult to analyze (what is the meaning of the interaction term?) and is not a source of valuable knowledge.

Regression trees are likely to provide another kind of information as they discriminate between variables to select the most important ones. The pruned crisp regression tree, shown in Fig. 5 has been obtained with the R software *rpart* implementation of the CART method, commonly used by statisticians. As expected, the key variable is the wind velocity and the second one to be introduced is the Air temperature.

The tree is easy to interpret, but crisp regression trees as this one suffer from well known drawbacks. The automatic binary variable partitioning for a given split may or may not be meaningful to the user. The results are highly sensitive to the split conditions and there is no interpolation between rules due to the crisp thresholds. Indeed, the possible inferred values are limited by the number of leaves, three in this case, giving poor convincing results.

The Quinlan (1986) derived implementations of decision trees allow multiple non binary splits learning using predefined partitions, but they do not interpolate either.

These drawbacks are likely to be overcome by fuzzy decision trees (FDT).

5.2.2. FDT design

The FDT design starts with the building of the fuzzy partitions. Then the tree is generated and pruned. Finally an optimization sequence is run on the equivalent FIS.

- Partition design
  When available, expert knowledge is introduced in the model through the fuzzy partition design. As shown in Fig. 6, the *Wind Speed* variable partitioning is done according to the Beaufort scale, leading to highly interpretable rules, as each linguistic label corresponds to a Beaufort degree.
- Tree generation and equivalent FIS
  The tree displayed in Fig. 7 is a pruned fuzzy regression tree obtained using *FisPro*. This kind of system can be used to recommend suited spraying periods: *avoid windy times but also, even if the wind velocity is moderate, prefer times when air temperature is high to minimize losses*.

A fuzzy tree equivalent FIS may automatically be derived from the fuzzy decision tree, by generating a rule for each path to a leaf. Thanks to the overlap in the variable partitions, and to the multiple membership, interpolation is allowed giving the output results plotted in Fig. 8. The determination coefficient is similar to the one gained using the linear regression, $R^2 = 0.70$.
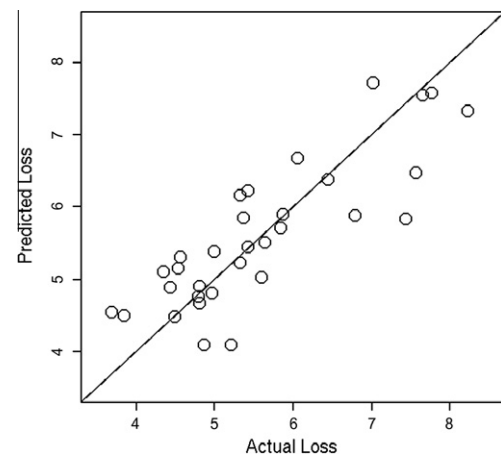

Fig. 4. Linear regression results for pesticide loss data.
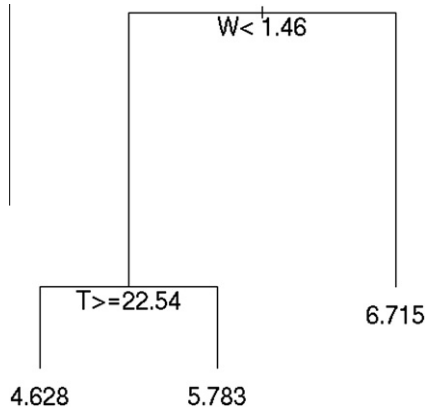
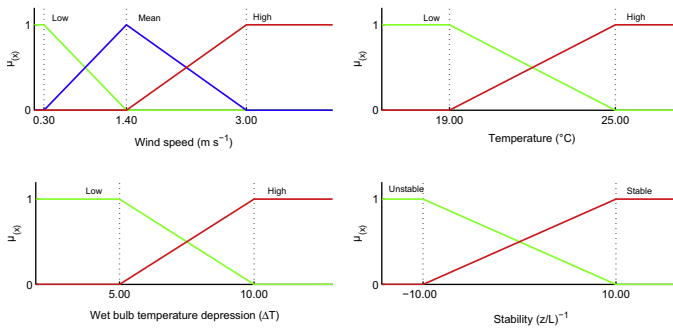**Fig. 5.** The crisp regression tree induced from pesticide loss data.

Fig. 6. Fuzzy input partitions for pesticide loss data.

5.2.3. Optimization

FisPro includes an optimization module. The algorithm is based upon the work by Solis and Wets (1981) and Glorennec (1996). It is summarized in Algorithm 1.

Algorithm 1. The Solis and Wets algorithm

> **input**: Initial vector $X^{(0)}$, noise magnitude *Nmag*
> **output**: The optimized vector, $X = X^{(Max)}$

1   Initialization: $k = 0$, $M^{(0)} = 0$.
2 **while** $k \leqslant$ Max **do**
3   Generate a Gaussian vector $G^{(k)}$,
4   with mean $M^{(k)}$ and noise magnitude *Nmag*
5   **if** $E(X^{(k)} + G^{(k)}) < E(X^{(k)})$ **then**
6       $X^{(k+1)} = X^{(k)} + G^{(k)}$
7       $M^{(k+1)} = 0.2M^{(k)} + 0.4G^{(k)}$
8   **else if** $E(X^{(k)} - G^{(k)}) < E(X^{(k)})$ **then**
9       $X^{(k+1)} = X^{(k)} - G^{(k)}$
10     $M^{(k+1)} = M^{(k)} - 0.4G^{(k)}$
11   **else**
12       $X^{(k+1)} = X^{(k)}$
13       $M^{(k+1)} = 0.5M^{(k)}$
14   **end**
15     $k = k + 1$
16 **end**

As the algorithm memorizes the *good* directions in the research space, only a set of connected parameters can be optimized within a given run. The sequence proposed in *FisPro* consists in optimizing each of the selected input variable partitions in turn, then if the output is fuzzy, the output partition, and finally the rule conclusions.

This sequence is run to optimize the fuzzy decision tree equivalent FIS. As the number of data items is small (32), the ten-fold cross validation is done on a 85/15% basis, i.e. 27 items for each learning set, and 5 for the corresponding test set. Only the *T* and *ΔT* variables are optimized, as the *z/L* one is absent from the rules, and the *W* (Wind speed) partition is an expert one, which should be left untouched. The final FIS is computed as the median FIS: each parameter is replaced by the median of the ten optimized systems.
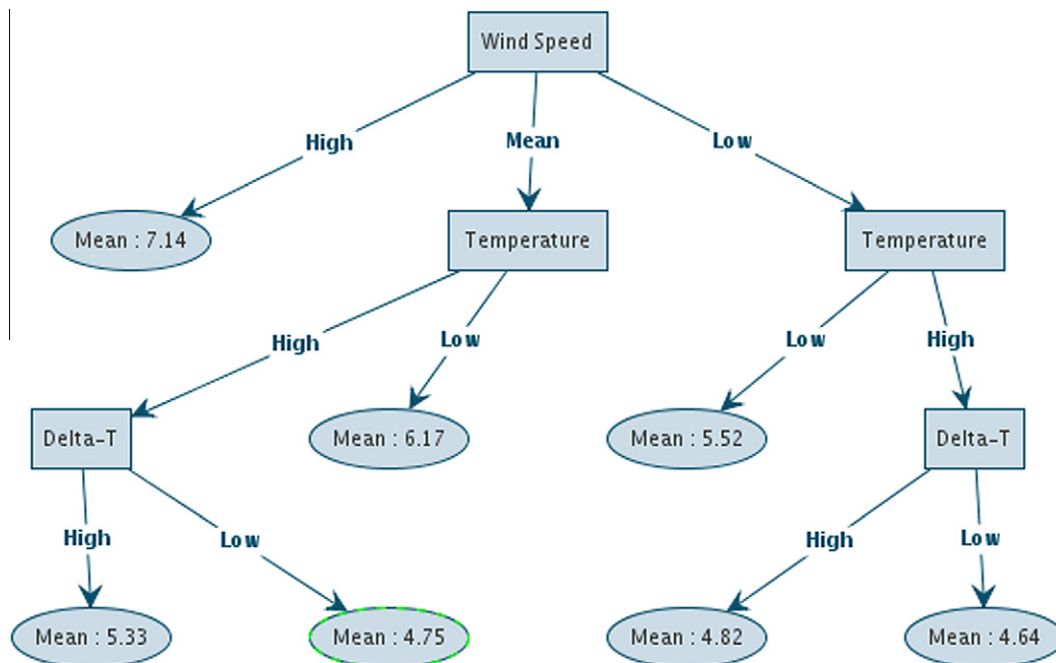

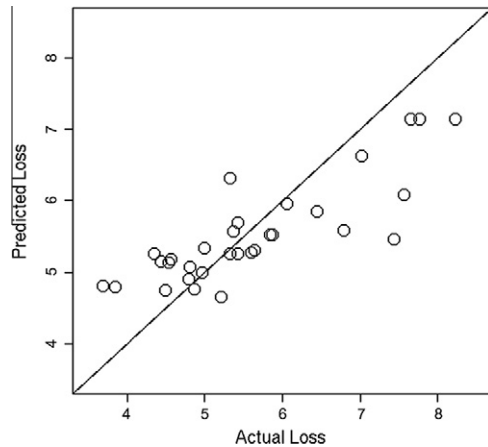**Fig. 7.** Fuzzy decision tree for pesticide loss data (fine spray).

Fig. 8. Fuzzy inference system results for pesticide loss data.

Table 1
Loss FIS: initial (I) and final (F) rule bases – Integers in the rule premise are MF numbers.

Rule premise			Rule conclusion	
W	ΔT	T	I	F
1		1	5.52	5.51
1	1	2	4.64	4.69
1	2	2	4.82	4.79
2		1	6.17	6.33
2	1	2	4.75	4.48
2	2	2	5.33	5.26
3			7.14	7.16

Table 2
Loss FIS optimization results: MAE averaged over the test sets.

Initial	0.65
Optim	0.62
Gain (%)	4.6
Final	0.54
Gain (%)	14.8

The two modified partitions are displayed in Fig. 9, together with the initial ones, and the rule base conclusions are printed in Table 1 for the initial and the final median FIS.

The optimization results are summarized in Table 2. *Optim* stands for the optimized FIS MAE, averaged over the ten test samples and *Final* is the final FIS MAE averaged over the same test sets.

Considering the whole data set, the final R^2 corresponding to the median FIS is 0.79, i.e. a 12% relative gain.

The optimization procedure allows a significant improvement of the FIS accuracy without modifying the FIS structure. The fuzzy partitions count with the same number of membership functions, the modifications lie in the numerical space and do not affect the linguistic interpretation. The same occurs for the rules: as the premises remain unchanged the rule base analysis is still valid.

This case study illustrates various ways of cooperation between expert knowledge and data. First the fuzzy partitions are expert designed to guarantee a high level of semantics, especially for the wind speed variable. Then, the data are used for the rule learning. Thanks to the system interpretability, the expert domain is able to analyze the system behavior. Finally, data is used again to optimize some system parameters while keeping the FIS structure and its properties. The accuracy is improved and the semantics is preserved.

5.3. System design using implicative rules

In this case study the main input output relationships are known. The goal is no longer rule induction and system analysis, like in the previous section, but expert knowledge modeling taking into account the data *imprecision*. The formalism of implicative rules (see Section 2.1.2) seems to be suited: rules are seen as constraints, each of them restricting the set of possible output values. Thus rules are combined using a conjunction operator. This yields a behavior quite different from a conjunctive rule base system, which is illustrated in the following.
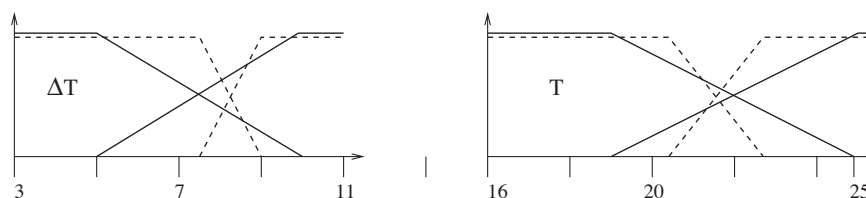
The problem under study is a predictive diagnosis for a hard-cooked type cheese-making process. Two parameters are important to determine cheese firmness: MC (Moisture Content), the cheese moisture content percentage at the end of the making process and DEE (Dry Extract Evolution), the loss of water during the first 15 days of the maturation process. The aim is to predict the cheese firmness at the end of maturation (4 to 10 months or longer) according to these two parameters. The two measurements (MC and DEE) come from sensors tainted with significant imprecision.

Input expert partitions are shown in Fig. 10. For each partition, a given input is plotted in dashed lines as a trapezoidal membership function. A vertical central dotted line corresponds to the precise input value, and the membership function to the approximate one, representing the measurement together with its imprecision.

The rule system presented here is a simplified system that does not take into account the whole process complexity:

- If MC is high and DEE is low then cheese will be soft
- If MC is high and DEE is high then cheese will be normal
- If MC is low and DEE is low then cheese will be normal
- If MC is low and DEE is high then cheese will be hard

5.3.1. Implicative inference

A screenshot of *FisPro* interactive inference window, using the precise data displayed in Fig. 10, MC = 54.5 and DEE = 0.6, is shown in Fig. 11. The aggregation result is plotted in the first row of the Firmness column, while each of the four following rows illustrate the inference mechanism for the corresponding rule.

The Resher–Gaines implication, see Jones et al. (2009) for details, is used. This operator yields a crisp interval, which coincides with the core of the output obtained by all other residuated implications.
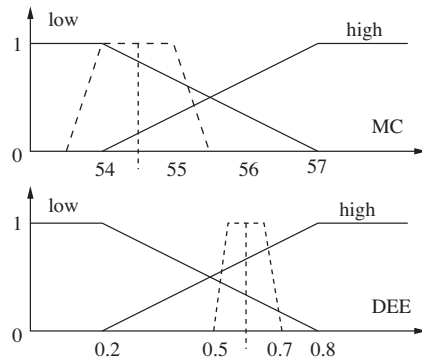

Fig. 9. Loss data FIS: the two modified partitions (in dashed lines).

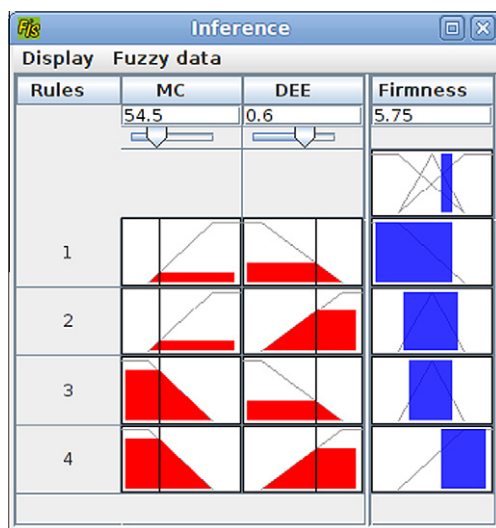**Fig. 10.** Input partitions – Precise and imprecise values used for the inference are plotted in dashed lines.

Fig. 11. Firmness inference with implicative rules and a precise value.

Fig. 12. Firmness inference with implicative rules and an imprecise value: FisPro window and detail, the output is the union of the three doors.

The output partition displayed in Fig. 11 (top right) and in Fig. 12 (bottom) is not a Strong Fuzzy Partition. It is automatically built from a two term Strong Fuzzy Partition, and it is called a Quasi Strong Partition (Jones et al., 2009). It ensures both the interpretability and the consistency required by the conjunctive aggregation.

For *crisp data*, the FITA[4] inference mechanism is used, i.e. the inference is done rule by rule, and then the aggregation is performed.

The screenshot of *FisPro* interactive inference window, using the precise data displayed in Fig. 10, MC = 54.5 and DEE = 0.6, is shown in Fig. 11.

Let us comment on this figure. The membership degrees within the rule premises are combined with the *min* operator, then the Resher-Gaines implication operator is applied to the fuzzy set that appears in the rule conclusion, resulting in a door shape possibility distribution. These intermediate possibility distributions are intersected to give the final one, plotted in the top right cell of the Firmness column: this is the aggregation step.

Thanks to the conjunctive aggregation, the output width is meaningful, it contains all the values which satisfy the constraints expressed by the rules. Note that an empty result would point out a rule base inconsistency (Dubois, Prade, & Ughetto, 1997). If needed, a precise value can be inferred using a defuzzification step, for in-
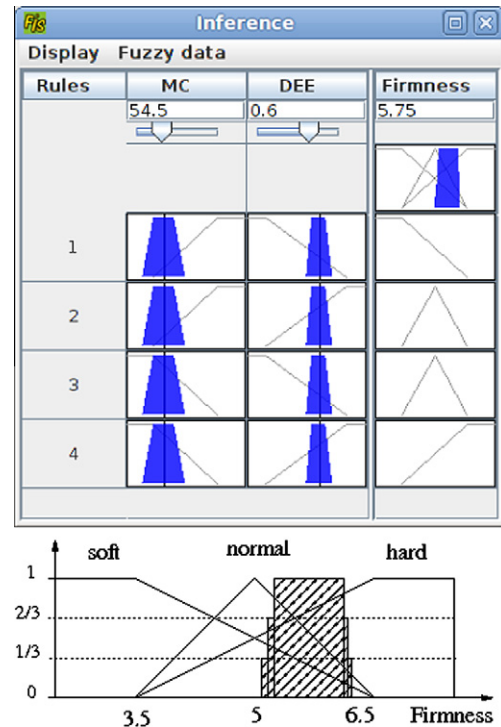
stance, the mean of the output possibility distribution core. This value (5.75) is displayed just below the Firmness field in Fig. 11.

Dealing with imprecise data, the inference cannot be done rule by rule. Only the FATI[5] mechanism is correct, and it is much more difficult to carry out in practice. In the *FisPro* implementation, the imprecise data are approximated by nested doors. Fig. 12 shows the Fispro inference window and the aggregation details, for the imprecise data plotted in Fig. 10, and with a three level nested door decomposition.

The input imprecision is respected in the output distribution.

As the output is a *normalized* possibility distribution, it can be interpreted as such, without the need for defuzzification. Thus the support and kernel widths increase with the input imprecision. This appears in Fig. 12, where the core and support of the possibility distribution are wider than the ones resulting from the implicative inference with the precise value, plotted in Fig. 11. This follows expectations, and it is an important point for decision support systems handling imprecise data, as the imprecision remains interpretable and meaningful. Furthermore, as input imprecision is taken account of all through the implicative inference process, implicative FIS allow to design cascading FIS respectfully of approximate inputs, and thus to design interpretable hierarchical fuzzy inference systems.

5.3.2. Conjunctive inference

To point out the differences with the well known Mamdani FIS, the inference results of the same rule base, using a conjunctive Mamdani rule base, with the same data, are now presented.

The screenshot from *FisPro* is shown on top of Fig. 13.

The inferred output overlaps the three output fuzzy sets. Consequently, it is difficult to interpret this result without defuzzification. Centroid-like defuzzification gives us a firmness equal to 6.3. Note that defuzzification is highly sensitive not only to the

---

[4] First Infer Then Aggregate.

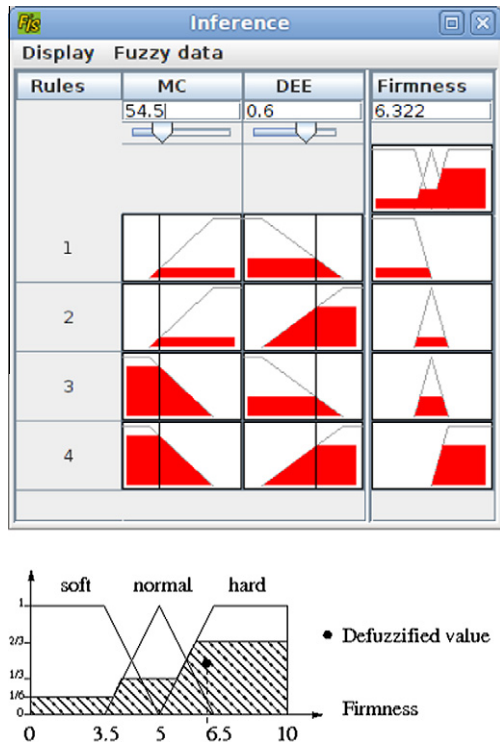[5] FATI means "First Aggregate Then Infer".

Fig. 13. Firmness inference with a conjunctive rule system and a crisp value-FisPro window and defuzzification detail.
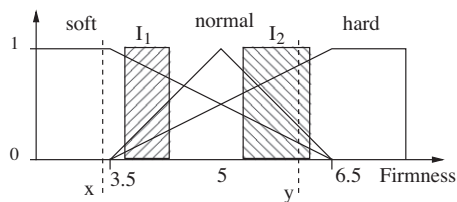

Fig. 14. Precision and accuracy labels.

Table 3
Labels according to the target (row) and estimated (column) values.

	I_1	I_2
x	Correct and precise	Incorrect
y	Incorrect	Correct but imprecise

Table 4
Implicative rule base validation.

Correct and precise	21
Correct but imprecise	49
Incorrect	33

defuzzification method, but also to the membership function shapes.

The inference of imprecise data with the conjunctive rule base is similar to what is done in the case of precise data, as the imprecise data are used to compute a single matching degree with each of the fuzzy sets within the partition.

This degree is then used in the same way than the membership degree of the precise value. The result is not plotted here as, due to defuzzification, it does not yield any interpretable result when examining the output possibility distribution versus the input imprecision.

With conjunctive inference the inferred output is a subnormalized possibility distribution, with no kernel, and it always requires a defuzzification step to yield a crisp output. Furthermore the support width is not related to the input imprecision.

5.3.3. Data validation

A set of 103 representative cheese sample has been analyzed by an expert panel. The final firmness result from the expert score aggregation. This sample has been used to validate the implicative rule base.

The inference process yields a possibility distribution. For the Resher-Gaines implication operator, this distribution reduces to an interval, as all the values have a possibility degree equal to 1 or 0. It is, of course, possible, to defuzzify this distribution, by computing a real value using this interval, for instance the mean, minimum or maximum. But, this step is not mandatory with an implicative rule base. Moreover, it would hide some interesting properties of the distribution, such as its range.

To take advantage of this inference mechanism, a more complex *label* is proposed to characterize the inferred output distribution with respect to the target value. The label is composed of two distinct integers:

- Precision: this value depends on the output interval width. To avoid the use of a threshold value, it is defined as the number of distinct MF in which the membership degrees of any point within the interval are the highest.
- Accuracy: this value is the prediction error, quantified as a number of MF. It is computed as the smallest difference between the indexes of the MF the target value mainly belongs to and the MF previously defined in the precision label.

The precision ranges from 1 to f, the number of MF in the partition. As we use a 3-MF output fuzzy partition, the precision can be *precise*, *imprecise* or *very imprecise*. The accuracy is between 0 and $f - 1$, any strictly positive value is considered as Incorrect.

To illustrate the label significance, Fig. 14 shows two firmness values, x and y, and two hypothetical inferred intervals, a narrow one on the left of the figure filled with a slash pattern, I_1, and a wider one on the right, filled with a backslash pattern, I_2. The corresponding labels are given in Table 3.

We now present the implicative FIS inference results, obtained on the experimental data set, and labeled in terms of precision and accuracy. Results are summarized in Table 4.

The use of *precision* and *accuracy* labels makes it possible to analyze the results with regard to these two criteria, contrary to what is usually done with Mamdani or Sugeno conjunctive FIS, where only numerical accuracy is discussed, without specifically considering the impact of input data imprecision. First of all, Incorrect predictions, due to examples that are in contradiction with the expert rules, show that these rules are not as general as it could be thought. Nevertheless, though simple, the rule base, made up of only 4 rules, is able to reflect some of the main data trends: 70% of the predictions are correct. A significant part of these good predictions are still imprecise, and this imprecision can be reduced by refining the rules with new variables. Indeed, with an implicative rule base, refinement leads to more constraints applied onto the data.

6. Conclusion

Cooperation between knowledge and data is still an open challenge in system modeling. In this paper, we show how fuzzy infer-

ence systems can be used for that purpose, by providing a powerful interface between the symbolic space used for knowledge representation and the numerical one. We present the main steps of an integrated approach for modeling with FIS and discuss its practical application.

Some ways of cooperation are illustrated through two case studies. In the first one, a study of pesticide loss during air spray, data play the central role in the modeling process, even though expert knowledge, when available, is also involved, for instance to base the wind velocity linguistic variable definition on the Beaufort scale. Valuable operational information can be derived from the rule base, for making recommendations about pesticide spraying.

The second case study discusses how expert knowledge about a cheese making process is formalized into an implicative rule base. Thanks to the conjunctive aggregation of implicative rules, the output distribution width is meaningful. When the input values are not precise, the imprecision is respected in the output distribution. From an engineering point of view, implicative rules allow the incremental design of systems. Adding a new variable to a given rule makes it more specific. There is no need to modify the existing rules. It can be noted that additional work is needed to learn and optimize implicative rule bases.

The approach framework is implemented in a free software, *FisPro*. Its originality stems from rule interpretability and exploratory tools to study the FIS behavior. Open, it welcomes contributions from scientists or engineers to increase its functionality and its scope. Some characteristics are given below.

- *FisPro* is open source, and to facilitate the inclusion in other programs, its core is delivered as a C++ library. The interface is written in Java, and available in several languages.
- Hierarchical FIS are available. This allows the reuse of a FIS output as input to another FIS.
- An optimization module allows to tune all parts of Mamdani and Sugeno FIS, regardless of the way they were built, by hand or through automatic learning.
- Sample generation is available prior to learning, as well as learning scripts including cross validation procedures.

Fuzzy rule base merging, for instance expert and induced rules, using the HILK method (Alonso, Magdalena, & Guillaume, 2008) is implemented in the *Generating Understandable and Accurate fuzzy models in a Java Environment (GUAJE)* software (Alonso, 2010), which is based on *FisPro*.

FisPro has been used for various modeling projects (Alonso, Magdalena, & González-Rodríguez, 2009; Bossomaier, Standish, & Harré, 2010; Bergasa, Nuevo, Sotelo, Barea, & Lopez, 2006; Rajaram & Das, 2010), and we hope that the approach presented in this paper will help in new modeling tasks.

References

Adomavicius, G., & Tuzhilin, A. (2001). Expert-driven validation of rule-based user models in personalization applications. *Data Mining and Knowledge Discovery, 5*(1-2), 33–58.

Alonso, J. M. (2010). Guaje: Generating understandable and accurate fuzzy models in a java environment. http://www.softcomputing.es/guaje.

Alonso, J. M., Magdalena, L., & González-Rodríguez, G. (2009). Looking for a good fuzzy system interpretability index: An experimental approach. *International Journal of Approximate Reasoning, 51*(1), 115–134.

Alonso, J. M., Magdalena, L., & Guillaume, S. (2008). Hilk: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems, 23*(7), 761–794.

Aptronics. (2007). Fide power design tool http://www.aptronix.com.

Babuška, R. (2003). *Neuro-fuzzy methods for modeling and identification*. Heidelberg, Germany: Physica-Verlag GmbH.

Baturone, I., Moreno-Velo, F. J., Sánchez-Solano, S., Barrios, A. B., Jiménez, P. B., Gersnoviez, A., & Brox, M. (2007). Using xfuzzy environment for the whole design of fuzzy systems. In IEEE (2007) (pp. 517–522).

Bergasa, L. M., Nuevo, J., Sotelo, M. A., Barea, R., & Lopez, M. E. (2006). Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems, 7*(1), 63–77.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective functions algorithms*. New York: Plenum Press.

Borgelt, C., & Gonzáles-Rodríguez, G. (2007). Frida – A free intelligent data analysis toolbox. In IEEE (2007) (pp. 1892–1896).

Bossomaier, T., Standish, R. K., & Harré, M. (2010). Simulation of trust in client-wealth management adviser relationships. *International Journal of Simulation and Process Modelling, 6*(1), 40–49.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. Wadsworth International Group, Belmont CA.

Casillas, J., Cordón, O., Herrera, F., & Magdalena, L. (2003). *Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: An overview. Interpretability issues in fuzzy modeling*. Springer, pp. 3–22.

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

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

Destercke, S., Guillaume, S., & Charnomordic, B. (2007). Building an interpretable fuzzy rule base from data using orthogonal least squares- application to a depollution problem. *Fuzzy Sets and Systems, 158*, 2078–2094.

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

Dubois, D., & Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems, 90*, 141–150.

Dubois, D., & Prade, H. (2003). Fuzzy set and possibility theory-based methods in artificial intelligence. *Artificial Intelligence, 148*, 1–9.

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

Dubois, D., Prade, H., & Ughetto, L. (2003). A new perspective on reasoning with fuzzy rules. *International Journal of Intelligent Systems, 18*, 541–567.

Foulloy, L., Galichet, S., Boukezzoula, R. (2005). Floulib. http://www.polytech.univ-savoie.fr/index.php?id=859.

Gil, Y., Sinfort, C., Guillaume, S., Brunet, Y., & Palagos, B. (2008). Influence of micrometeorological factors on pesticide loss to the air during vine spraying: Data analysis with statistical and fuzzy inference models. *Biosystems Engineering, 100*(2), 184–197.

Glorennec, P.-Y. (1996). *Constrained optimization of FIS using an evolutionary method. Genetic algorithms and soft computing*. Physica-Verlag, pp. 349–368.

Glorennec, P. -Y. (1999). Algorithmes d'apprentissage pour systèmes d'inférence floue. Editions Hermès, Paris.

GmbH, M. (2001). Dataengine. http://www.dataengine.de.

GmbH, I. (2007). Fuzzytech. http://www.fuzzytech.com.

Guillaume, S. (2001). Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems, 9*(3), 426–443.

Guillaume, S., & Charnomordic, B. (2004). Generating an interpretable family of fuzzy partitions. *IEEE Transactions on Fuzzy Systems, 12*(3), 324–335.

Guillaume, S., & Charnomordic, B. (2011). Learning interpretable fuzzy inference systems with fispro. *International Journal of Information Sciences, 181*, 4409–4427.

Guillaume, S., & Magdalena, L. (2006). Expert guided integration of induced knowledge into a fuzzy knowledge base. *Soft computing, 10*(9), 773–784.

Hohensohn, J., Mendel, J. M. (1994). Two pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems. In *Proceedings of IEEE Conference on Fuzzy Systems Orlando, Florida* (pp. 696–700).

Hullermeier, E. (2005). Fuzzy methods in machine learning and data mining: Status and prospects. *Fuzzy sets and Systems, 156*, 387–406.

IEEE (Ed.), (2007). FUZZ-IEEE 2007, IEEE International Conference on Fuzzy Systems, Imperial College, London, UK, 23–26 July, 2007, *Proceedings of IEEE*.

InfraLogic, T. (1995). The world's source for fuzzy logic solutions http://www.ortech-engr.com/fuzzy/togai.html.

Janikow, C. Z. (2005). Fuzzy decision tree/forest. http://www.cs.umsl.edu/janikow/fid/.

Jones, H., Charnomordic, B., Dubois, D., & Guillaume, S. (2009). Practical inference with systems of gradual implicative rules. *IEEE Transactions on Fuzzy Systems, 17*(1), 61–78.

Kruse, R. (2007). Neural networks and fuzzy systems. http://fuzzy.cs.uni-magdeburg.de/.

Lee, C. C. (1990). Fuzzy logic in control systems: Fuzzy logic controller-part ii. *IEEE Trans. on Systems, Man, and Cybernetics, 20*(2), 419–435.

Ling, T., Kang, B. H., Johns, D. P., Walls, J., & Bindoff, I. (2008). Expert-driven knowledge discovery. In Latifi, S. (Ed.), *Proceedings of the fifth international conference on information technology: New generations* (pp. 174–178).

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of Man–Machine Studies, 7*, 1–13.

Martin-Clouaire, R. (1989). Semantic and computation of the generalized modus ponens: The long paper. *International Journal of Approximate Reasoning, 3*(2), 195–217.

MathWorks. (2007). Fuzzy logic toolbox http://www.mathworks.com/products/fuzzylogic.

Nauck, D. D. (2007). Gnu fuzzy. In IEEE (2007) (pp. 1019–1024).

Orchard, R. A. (2004). Fuzzy extension to the clips expert system shell (fuzzyclips). http://www.nrc-cnrc.gc.ca/fra/licences/iti/non-commerciale9.html.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Rajaram, T., & Das, A. (2010). Modeling of interactions among sustainability components of an agro-ecosystem using local knowledge through cognitive mapping and fuzzy inference system. *Expert Systems with Applications, 37*(2), 1734–1744.

Research, W. (2007). Mathematica fuzzy logic http://www.wolfram.com/products/applications/fuzzylogic.

Ruspini, E. H. (1982). *Recent developments in fuzzy clustering*. New York: Pergamon Press (pp. 133–147).

Sazonov, E. (2002). Open source fuzzy inference engine for java http://people.clarkson.edu/esazonov/FuzzyEngine.htm.

Sester, M. (2000). Knowledge acquisition for the automatic interpretation of spatial data. *International Journal of Geographical Information Science, 14*(1), 1–24.

Solis, F. J., & Wets, R. J. (1981). Minimization by random search techniques. *Mathematics of Operation Research, 6*, 19–30.

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

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

Wang, L.-X., & Mendel, J. M. (1992a). Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE Transactions on Neural Networks, 3*, 807–814.

Wang, L.-X., & Mendel, J. M. (1992b). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics, 22*(6), 1414–1427.

Weber, R. (1992). Fuzzy-id3: A class of methods for automatic knowledge acquisition. In *2nd International conference on fuzzy logic and neural networks* (pp. 265–268).

Weisbrod, J. (1998). A new approach to fuzzy reasoning. *Soft Computing, 2*, 89–99.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.

Zadeh, L. A. (1975). The concept of linguistic variable and its application to approximate reasoning – Parts I, II and III. *Information Sciences, 8–9*, 301–357 (pp. 43–48).

Zadeh, L. A. (2008). Is there a need for fuzzy logic? *Information Sciences, 178*(13), 2751–2779.