Quest for Interpretability-Accuracy Trade-off Supported by Fingrams into the Fuzzy Modeling Tool GUAJE

David P. Pancho, Jose M. Alonso, L. Magdalena

European Centre for Soft Computing,
Edificio de Investigación, C/ Gonzalo Gutiérrez Quirós s/n,
33600 Mieres, Asturias, Spain
E-mail: {david.perez,jose.alonso,luis.magdalena}@softcomputing.es

Received 19 December 2012
Accepted 7 March 2013

Abstract

Understand the behavior of Fuzzy Rule-based Systems (FRBSs) at inference level is a complex task that allows the designer to produce simpler and powerful systems. The fuzzy inference-grams –known as fingrams– establish a novel and mighty tool for understanding the structure and behavior of fuzzy systems. Fingrams represent FRBSs as social networks made of nodes representing fuzzy rules and edges representing the degree of interaction between pairs of rules at inference level (no edge means no significant interaction). We can analyze fingrams obtaining helpful information such as detecting potential conflicts between rules, unused rules and redundant ones. This paper introduces a new module for fingram generation and analysis included in the free software tool GUAJE. This tool aims to design, analyze and evaluate fuzzy systems with good interpretability-accuracy trade-off. In addition, GUAJE includes several intuitive and interactive tutorials to uncover the possibilities it offers. One of them generates and enhances a fuzzy system, analyzing each improvement through the use of fingrams, and lets the user reproduce the illustrative case study described in this paper.

Keywords: Interpretability-accuracy trade-off, fuzzy modeling, fingrams, GUAJE.

1. Introduction

Fuzzy sets and systems have become a mature research field with many theoretical and applied works starting from Zadeh's seminal work [36]. Among the huge number of research lines developed by the fuzzy community, system modeling with fuzzy rule-based systems (FRBSs) –called fuzzy modeling [23]– has been a fruitful research line for years.

During a long period –from 1965 to 1990– fuzzy modeling was mainly supported by expert knowledge. Researchers concentrated on building fuzzy models made up of a few simple linguistic variables [38] and linguistic rules [37] usually referred as Mamdani rules [27]. Accordingly, those designed fuzzy models were easy to interpret, and interpretability emerged naturally as an important advantage. However, researchers realized that considering only expert knowledge was not enough when dealing with complex real-world problems. Fortunately, it is also possible to build fuzzy models automatically –following a machine learning approach– from experimental data [24].

Thus, in a second period –from 1990 to 2000– researchers focused on automatically creating accurate fuzzy systems from experimental data, although disregarding interpretability. Of course, fuzzy systems are not interpretable *per se*, and automatically generated rules are rarely as readable as desired.

Interpretability has recovered a main role inside the fuzzy community since 2000 [5]. Researchers realized that accuracy and interpretability should be cared together, although, both issues are somehow contradictory. High accuracy usually implies low interpretability and *vice versa*. In practice, fuzzy modeling involves careful design where both interpretability and accuracy must be taken into account along the design process. The quest for the right interpretability-accuracy trade-off has become a great challenge in the last decade [16]. Nowadays, many researchers are actively working on it but a lot still remains to be done.

The notion of fuzzy inference-gram (*fingram*), recently introduced [4], is a powerful tool supporting the quest for interpretability-accuracy trade-off. In short, the behavior of a fuzzy system is analyzed –at inference level– by looking at those pairs of co-fired (simultaneously fired) rules by a given input vector. Then, a social network represents the rule base interaction where each individual entity represents a rule, and edges connecting entities show the relations among rules.

This paper explains how the free open source software GUAJE has been recently enhanced with a new module in charge of fingram generation and analysis [10]. In addition, GUAJE offers several optimization and simplification tools –at both fuzzy partition and fuzzy rule level– devoted to improve the accuracy and interpretability of the entire fuzzy system. The new module for fingrams eases the quest for interpretability-accuracy trade-off along the entire modeling process. Fingrams let us visually analyze and uncover the behavior and consequences of the applied optimization and simplification techniques. As a result, the designer can dynamically change the related parameters and/or improvement strategies with the aim of achieving the best balance between interpretability and accuracy.

The rest of the paper is organized as follows. Section 2 gives a global overview on existing software for fuzzy systems. Then, Section 3 introduces fingrams and their uses in fuzzy modeling. Section 4 presents the open source tool GUAJE, devoted to design and analyze FRBSs, which includes a new module for handling fingrams. Afterwards, we use a simple but highly illustrative use case in Section 5 to show the potentials of fingrams. Finally, some conclusions and future works are sketched in Section 6.

2. Fuzzy systems software overview

Along this long trip (more than forty five years), the fuzzy community has produced many publications regarding both theoretical and practical issues, and as a side effect, a lot of software tools have been developed too.

There exist some powerful and widely known commercial tools like the Fuzzy Toolbox[a] and the Adaptive Neuro-Fuzzy Inference System (ANFIS[b]) both provided by Matlab; the software fuzzyTECH[c]; or the fuzzy package provided with Wolfram Mathematica[d].

Anyway, this contribution focuses on open source tools that have recently reached a high level of development. They offer the richness of quickly incorporation of new developments made by the active research community, playing an important role in academy and industry. Moreover, most of this software is freely downloadable for research and education purposes, and facilitates the design of advanced prototypes for many novel applications.

We would like to highlight the following open source tools for fuzzy modeling –our principal interest in this contribution– because of their good performance and human-friendly interfaces:

- FisPro[e] (Fuzzy Inference System Professional). An open source tool for creating fuzzy inference systems [21]. It includes many algorithms for generating fuzzy partitions and rules directly from experimental data. FisPro aims at simulating physi-

[a] http://www.mathworks.es/products/fuzzylogic/index.html

[b] http://www.mathworks.es/help/toolbox/fuzzy/anfisedit.html

[c] http://www.fuzzytech.com

[d] http://www.wolfram.com/products/applications/fuzzylogic

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

cal or biological systems, making emphasis in reasoning purposes. It eases the integration of expert knowledge and knowledge extracted from data. FisPro has been successfully applied to agriculture and environmental modeling problems [22].

- Xfuzzy[f]. A development environment aimed at producing fuzzy inference-based systems [11,12]. It integrates a set of tools covering all design stages from description to synthesis. Xfuzzy has been recently enhanced with an XML-based language called XFSML [28] that makes easier the interoperability among complementary tools.
- GUAJE[g]. A free software tool for generating understandable and accurate FRBSs in a java environment [6]. It allows combining expert knowledge and knowledge automatically extracted from data. GUAJE integrates several algorithms provided by different open source software tools. Moreover, the user can export models generated by GUAJE to other program formats, like FisPro, Xfuzzy, or the Matlab Fuzzy Toolbox.

As previously hinted, establish a standard language is an increasing important requirement. Regarding fuzzy control, there is a standard language called Fuzzy Control Language (FCL) published by the International Electrotechnical Commission (IEC 61131-7). Notice that, FCL is implemented in the open source library named jFuzzyLogic [17]. A variant of FCL based on XML which is called Fuzzy Markup Language (FML [1]) has been recently proposed and it is under standardization process.

Other important and more ambitious open source tools are KNIME[h] (Konstanz Information Miner) [14], a modular environment which is especially endowed with data manipulation and visualization methods but also with fuzzy rule learning capabilities [13]; FRIDA[i] (Free Intelligent Data Analysis Toolbox) [15] that provides methods for statistical analysis but also with visualization capabilities; and KEEL[j] (Knowledge Extraction based on Evolutionary Learning) [2] that probably contains the most complete collection of algorithms for genetic fuzzy systems. In addition, KEEL offers a user-friendly GUI for designing experiments and an educational data mining tool.

3. Fingrams

The term *fingram* stands for fuzzy inference-gram. It was coined in [4] by inspiration on the term scientogram firstly introduced by Vargas-Quesada and Moya-Anegón [29] in the search for a new tool aimed at visualizing the structure of science [35].

We have recently proposed a methodology for visual representation and exploratory analysis of the fuzzy inference process in FRBSs [30]. With that aim, fingrams represent FRBSs as social networks, giving very useful information about the FRBS behavior. They are made of nodes representing fuzzy rules and weighted edges that show graphically the interaction between rules at inference level.

Different aspects of teamwork between rules can be considered, producing different fingrams. As a first approach, we use co-firing between rules, i.e., rules fired at the same time by a given input vector. Therefore, rules highly related are more frequently fired together. Given a fuzzy system containing N rules and an experimental dataset covering most possible situations, we automatically generate an $N \times N$ weight matrix M describing the interactions between the N rules in terms of frequency of co-firing.

$$M = \begin{pmatrix} 0 & m_{12} & \dots & m_{1N} \\ m_{21} & 0 & \dots & m_{2N} \\ \dots & \dots & \dots & \dots \\ m_{N1} & m_{N2} & \dots & 0 \end{pmatrix} \qquad (1)$$

The co-firing measure (m_{ij}) is defined by the next equation:

$$m_{ij} = \begin{cases} \frac{SFR_{ij}}{\sqrt{FR_i \cdot FR_j}} & , \quad i \neq j \\ 0 & , \quad i = j \end{cases} \qquad (2)$$

[f] https://forja.rediris.es/projects/xfuzzy/

[g] http://www.softcomputing.es/guaje

[h] http://www.knime.org

[i] http://www.borgelt.net/frida.html

[j] http://sci2s.ugr.es/keel/

SFR_{ij} counts the number of samples firing simultaneously rules R_i and R_j. FR_i and FR_j count respectively the total number of samples firing rules R_i and R_j, disregarding if they are fired together or not.

Once matrix M is obtained, then it becomes straightforward the generation of an initial network (undirected graph) made up of N nodes connected through edges whose weights are directly taken from M. However, since rules usually cover the input space with dense overlapping among them, the resultant network is usually quite dense and difficult to understand. Accordingly, we apply a scaling mechanism to simplify the representation what allows the users to focus their attention in the most transcendent relations. The Pathfinder algorithm [18] is chosen due to its mathematical properties including the conservation of links and the representation of the most salient relationships present in the data. Pathfinder considers two main parameters:

- $r \in [1,\infty)$. It defines the *Minkowski r-metric* considered to measure the distance between two nodes not directly connected:

$$D = \left\{ \sum_i D_i^r \right\}^{\frac{1}{r}} \tag{3}$$

  In case r takes value 1, then D results in the sum of the link weights; $r = 2$ yields the usual Euclidean metric; and when $r \rightarrow \infty$ the path weight is the same as the maximum weight associated with any link along the path.
- $Q \in [2, N-1]$. It limits the number of links in the paths for which the triangle inequality is ensured in the scaled network. Hence, Pathfinder removes every path that connect two nodes that violate the triangle inequality, having an associated distance greater than any other path between the same two nodes composed of up to Q links.

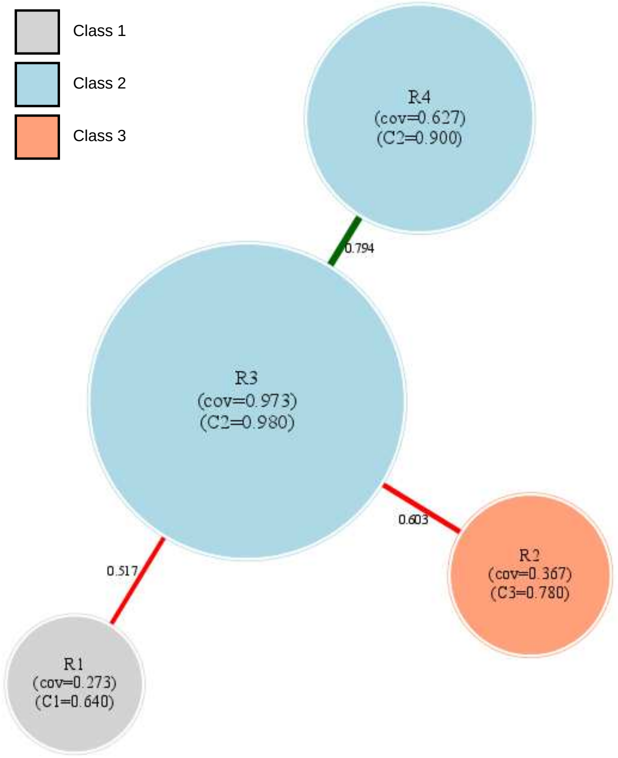
After scaling the network, the resultant network is visualized. Among the family of spring-embedder algorithms, we select the so-called *force-based algorithms* to automatically visualize the resultant networks. Kamada-Kawai [26], one of the most extended methods, assigns coordinates to the nodes trying to adjust, in an aesthetical pleasing way, the distances existing among them with respect to the actual interactions. The combination of rule co-firing, Pathfinder, and Kamada-Kawai places the most important nodes (i.e., those sharing more sources with the rest) toward the center. We call fingram to the final graphical representation of the network.

Fingrams have already been used in classification and regression problems [30]. They adopt different characteristics in each case, showing specific particularities of the problem represented. Fig. 1 shows an illustrative example related to the well-known IRIS classification problem[k]. The dataset contains 3 classes, each one referring to a type of iris plant. Note that rules pointing out the same output class are plotted with the same background color, and rules related to the same class are linked with green edges while red edges highlight potential inconsistency problems. In addition, nodes are labeled with informative values like coverage (*cov*) or goodness (*G*). In the figure, it is easy to appreciate how rule R3 covers most data samples and it goes in conflict against rules R1 and R2.

The analysis of a fingram can report helpful information about the analysis and verification of the related FRBS. We can systematically detect abnormal behaviors through carefully looking at the visual representation of a FRBS. Some simple but very useful examples of fingrams analysis are: identification of rules that cover a small amount of problem instances, perception of rules that exactly cover the same problem instances, detection of a rule that covers problem instances alone, assessment of FRBS comprehensibility, etc.

Regarding the comprehensibility analysis, we assume a large number of co-fired rules means a hardly comprehensible FRBS. Thus, the complexity of understanding the fuzzy inference process in terms of rules co-firing information can be evaluated by the Co-firing Based Comprehensibility Index (*COFCI*) [9]:

[k] IRIS dataset is freely available at the KEEL machine-learning repository [http://sci2s.ugr.es/keel/].



R1: IF Sepal Length is Average AND Sepal Width is High THEN Class is C1

R2: IF Sepal Length is High AND Sepal Width is Average THEN Class is C3

R3: IF Sepal Length is Average AND Sepal Width is Average THEN Class is C2

R4: IF Sepal Length is Average AND Sepal Width is THEN Class is C2

Figure 1: Intuitive example of fingram.

$$COFCI = \begin{cases} 1 - \sqrt{\frac{CI}{MaxThr}}, & \text{if } CI \leqslant MaxThr \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$CI = \sum_{i=1}^{N} \sum_{j=1}^{N} [(P_i + P_j) \cdot m_{ij}] \tag{5}$$

N is the number of rules, P_i and P_j count the number of antecedent conditions in rules R_i and R_j, respectively. m_{ij} is the measure of co-firing (Eq. 2) regarding rules R_i and R_j. $MaxThr$ is a normalization factor determined heuristically.

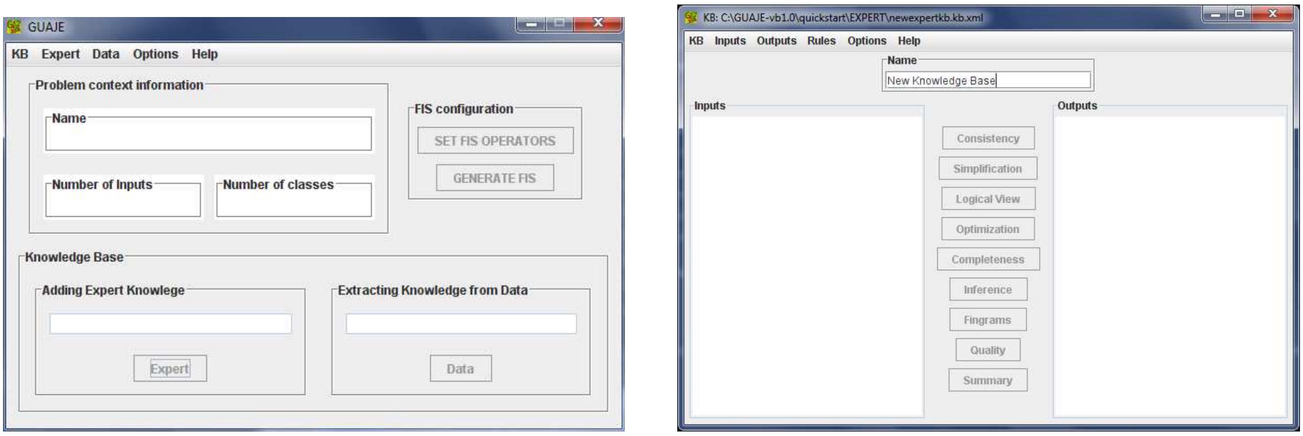
For getting a complete explanation about fingrams creation, interpretation and analysis, the interested reader is kindly referred to [30] where fingrams are deeply described.

4. GUAJE

GUAJE stands for Generating Understandable and Accurate fuzzy systems in a Java Environment [6]. Namely, GUAJE implements the fuzzy modeling methodology named as Highly Interpretable Linguistic Knowledge [7,8] that was conceived with the aim of yielding fuzzy systems endowed with good balance between interpretability and accuracy. GUAJE has been carefully developed in order to become user-friendly. In consequence, it makes the design of interpretable FRBSs easy and intuitive.

Fig. 2 shows the *Main* and *Expert Windows* of GUAJE. This free software actually consists of a modular architecture which is made up of several software modules in charge of the following tasks:

- **Data pre-processing**. It includes data visualization and analysis, re-sampling, etc.
- **Feature selection**. It focuses on identifying the most significant input variables.



Name: New Knowledge Base

(a) Main window of GUAJE.

(b) Expert window of GUAJE.

Figure 2: Screenshots of GUAJE.

- **Partition design**. It deals with the characterization of each input variable as a linguistic variable with a justifiable number of meaningful linguistic terms. The attached membership functions can be defined by an expert and/or they can be automatically derived from data using machine learning techniques.
- **Rule base definition**. The system behavior can be described by a set of linguistic IF-THEN rules. An expert can define the rules and/or GUAJE can derive them from data.
- **Knowledge base verification**. This modeling stage verifies the consistency of the knowledge base previously defined. This analysis must be done at both linguistic and inference levels.
- **Knowledge base visualization**. The module for fingram generation and analysis shows graphically the system behavior in terms of rule co-firing at inference level.
- **Knowledge base improvement**. It tries on getting systems with better interpretability-accuracy trade-off. Two main options are available: linguistic simplification and partition tuning.
- **Knowledge base validation**. It checks if the designed fuzzy system matches with the expert requirements and expectations.
- **Knowledge base evaluation**. The quality assessment module reports tables including several indices for evaluating both interpretability and accuracy.

In the rest of this section we will provide a deeper analysis of those modules that are the most relevant for the remainder of this contribution.

4.1. *Partition design*

This module allows defining the relevant linguistic variables for the problem under consideration. GUAJE permits selecting the number of linguistic terms –from 2 to 9– for each input variable. They must be fully meaningful. Therefore, it is possible to choose among several pre-defined sets of vocabulary tuples (*low-high*, *small-large*, and so on) but also set new linguistic terms customized by the user.

The characterization of each linguistic term can be made by two different approaches. On the one hand, an expert can define them choosing prototype values by hand. On the other hand, GUAJE can derive partitions by using several induction methods. In both cases, GUAJE imposes the use of strong fuzzy partitions [33] with the objective of maximizing interpretability.

4.2. *Rule base definition*

This module provides mechanisms to define the rules which compound the FRBS. The user can create rules by two different, but not incompatible, approaches. Firstly, he/she can define the rules by expert knowledge. As an alternative, machine learning techniques can derive rules from data, either to constitute the rule base or to complement pre-existing expert rules. GUAJE permits the use of *Fuzzy Decision Trees*, *Wang and Mendel*, *Fast Prototyping*, and *Prototype Rule* algorithms [7].

In addition, the user can choose among typical methods for rule conjunction (*minimum*, *product*, or *Łukasiewicz*) and disjunction (*maximum* or *sum*).

4.3. *Knowledge base visualization*

This module shows graphically the interaction among rules at inference level [10].

GUAJE first generates a co-firing matrix regarding the pairs of rules simultaneously fired by each problem instance. Then, Pathfinder scales the graph related to such matrix. Afterwards, Kamada-Kawai algorithm determines the placement of nodes. Then, the graph is enhanced with information related to rules (coverage, goodness, etc.). Finally, the resultant fingram is displayed to the user who can analyze it and interact with it, making zoom in and zoom out, removing nodes, and so on.

4.4. *Knowledge base improvement*

This module aims to obtain a better balance between interpretability and accuracy.

GUAJE provides two ways to improve the interpretability-accuracy trade-off of a given FRBS. On the one hand, rule base simplification permits increasing readability of the FRBS by reducing its complexity (in terms of number of rules, premises, variables, linguistic terms, etc.) but without jeopardizing accuracy beyond a pre-defined threshold. On the other hand, fuzzy partitions can be tuned in order to increase accuracy while keeping comprehensibility (because of imposing several semantic constraints).

4.4.1. *Rule Base Simplification*

The goal is the generation of a more compact FRBS, regarding both fuzzy partitions and rules, thus improving interpretability while preserving accuracy. GUAJE offers three alternatives:

Genetic rule selection: The initial FRBS is used for building the first individual of the population. A binary-coded chromosome with size N (the number of initial rules) is generated. Depending on whether a rule is selected or not, values 1 or 0 are respectively assigned to the corresponding gene. At the beginning all rules are supposed to be selected. The rest of the population is randomly generated. In each generation, parents are selected by binary tournament. Then, uniform crossover and flip-flop random mutation are applied. The best individuals automatically pass to the next generation by elitism. The stopping criteria are the maximum number of generations, or fitness under the predefined threshold. Fitness function is defined as the weighted sum of the accuracy and interpretability indices selected by the user among all those ones provided by GUAJE.

Logical view reduction: First of all, the current rule base is transformed into several truth tables without any change of semantics. This is possible thanks to the propositional view of fuzzy rule-based systems handled by GUAJE. In a second step, the truth tables previously generated are, in turn, minimized by applying truth-preserving operators (but without taking care of accuracy which may go down dramatically depending on how the initial rules were defined). Then, the new set of truth tables is transformed into propositions in the third step, so that constituting a new rule base, different but derived from the original one.

Linguistic simplification: It includes rule-based reduction and partition simplification, at linguistic level. In short, it is an iterative process which first acts on the rules and then on the partitions at each iteration. This cycle is repeated until no more interpretability improvement is feasible without penalizing accuracy beyond a predefined threshold. Firstly, the procedure looks for redundant elements (linguistic terms, premises, rules, etc.) that can be removed without altering the system accuracy. Then, it tries to merge elements always used together. Finally, it removes elements apparently needed but not contributing too much to the final accuracy.

4.4.2. Partition Optimization

The goal is increasing accuracy without jeopardizing interpretability. The optimization task only affects the fuzzy partitions that define the system variables. The partition tuning is constrained to maintain strong fuzzy partitions. Two strategies are considered [3]:

Genetic tuning: An *all-in-one* optimization procedure based on a global search strategy. It is actually a genetic tuning process that looks for adjusting all system parameters at the same time. The procedure starts with a population of randomly generated solutions represented by real-coded chromosomes. Parents are selected by binary tournament at each generation. Then, $BLX - \alpha$ crossover and random mutation are applied. The best individuals are preserved by elitism. The stopping criteria are the maximum number of generations, or fitness under the predefined threshold. Fitness function is defined by the accuracy index selected by the user among all those ones provided by GUAJE.

Solis-Wets: An element by element optimization procedure based on the classical local search strategy proposed by Solis and Wets [34]: It is *a hill climbing method with memorization of the previous successes* [19]. The goal is not to find the global optimum, but to improve accuracy by performing a few iterations. Two alternatives are available: Variable by variable, and label by label.

4.5. Knowledge base evaluation

This module provides a complete overview about the quality (regarding both accuracy and interpretability) of the designed FRBSs.

For accuracy assessment there are universal indices commonly accepted, as the percentage of covered samples (*Coverage*) or the percentage of misclassified samples (*MC*), in classification problems:

$$MC = \frac{1}{d}\sum_{i=1}^{d} err_i;\ \ err_i = \begin{cases} 1, & \text{if } C_i \neq \widehat{C_i} \\ 0, & \text{otherwise} \end{cases} \qquad (6)$$

d is the number of instances. C_i represents the class of instance i. $\widehat{C_i}$ is the inferred class.

On the contrary, there is not any well established and widely recognized interpretability index. Even more, there is a need to look for two kind of complementary indices, objective and subjective ones [5]. In this paper, two objective indices, the number of rules (NR) and the total rule length (TRL), and one subjective index, the $COFCI$ index (Eq. 4), will be used to evaluate the interpretability of FRBSs. Notice that TRL counts the total number of linguistic propositions into the whole rule base.

5. Illustrative case study. Generation and analysis of fingrams with GUAJE.

GUAJE has been enhanced with a new software module for fingram generation and analysis. It is successfully integrated with the rest of modules of the software architecture as it was sketched in the previous section. This section details how to handle (generate, manipulate and analyze) fingrams in GUAJE through an example. For the sake of clarity the case study focuses only on a highly illustrative classification problem even though fingrams can also be applied to any classification or regression problem. We have selected a very well-known benchmark classification problem, WINE[1] just for illustrative purposes.

WINE dataset contains 178 instances coming from the results of a chemical analysis of wines

[1] WINE dataset is freely available at the KEEL machine-learning repository [http://sci2s.ugr.es/keel/].

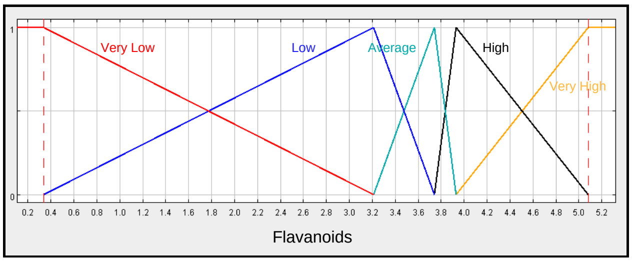


Figure 3: Linguistic terms and membership functions of variable *Flavanoids*.

grown in the same region in Italy from three different cultivars. Thus, the output of the FRBS will be one categorical variable including the 3 classes of wines. In addition, the quantities of 13 constituents (*Alcohol*, *Malic acid*, *Ash*, *Alcalinity of ash*, *Magnesium*, *Total phenols*, *Flavanoids*, *Nonflavanoids phenols*, *Proanthocyanins*, *Color intensity*, *Hue*, *OD280/OD315 of diluted wines*, and *Proline*) are taken as inputs.

The dataset is split into two samples for training and test, trying to avoid overfitting. The training set comprises 75% (133 instances) of the available data picked at random. The test set is compound by the remaining 25% (45 samples).

Then, each input variable is characterized by a strong fuzzy partition which contains five membership functions (of triangular shape). They take values in the universe of discourse determined by the minimum and maximum available data. Fuzzy partitions are automatically derived from training data with the Hierarchical Fuzzy Partitioning (HFP) algorithm [20]. Fig. 3 shows, just for illustrative purpose, the membership functions attached to each linguistic term of variable *Flavanoids*. As it can be appreciated, partitions generated by HFP are always strong fuzzy partitions but not necessarily uniform. Notice that, interpretable fuzzy partitions must represent prototypes that are meaningful for the expert and context-dependant, but this fact does not imply they have to be uniform.

Moreover, we must set a meaningful linguistic term related to each membership function because we deal with linguistic variables. This way, we define the basic vocabulary to be used later in the definition of linguistic rules. We have considered the following linguistic terms: *Very Low / Low / Average / High / Very High.*

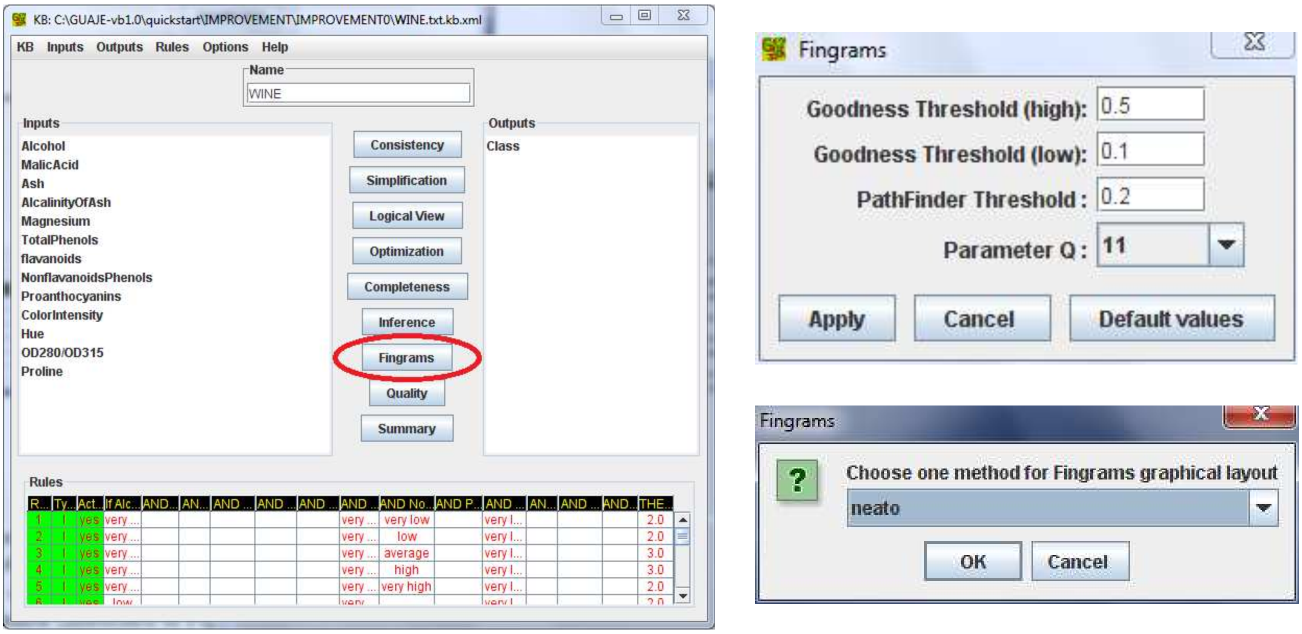
Before defining fuzzy rules, it is time to choose carefully the involved fuzzy operators, because they directly alter the inference mechanism. We have selected *minimum*, *maximum* and *max crisp* as t-norm, t-conorm and defuzzification method respectively, the usual inference scheme considered when dealing with classification problems.

Looking for a set of general rules that exhibit a good interpretability-accuracy trade-off, we have induced rules using Fuzzy Decision Trees (FDT) algorithm [25]. FDT can be seen as a fuzzy version of the popular decision tree induction algorithm defined by Quinlan [31]. The GUAJE implementation of FDT is actually based on the generation of a neuro-fuzzy decision tree which is easily translated into quite general incomplete rules where only a subset of input variables is considered. The result of running FDT (with maximum tree depth equal to 3; and using the fuzzy partitions previously generated) is a rule base made up of 32 rules. As expected, not all inputs are considered in all the rules. In fact, each rule uses only a subset of the input variables (3.4 inputs per rule in average). Picture on the left side in Fig. 4 shows the *Expert Window* of GUAJE after generating fuzzy partitions and rules. The top part of the picture shows the variables (13 inputs and 1 output) while a table represents the generated rule base at the bottom. Each row corresponds to one rule while each column represents one variable.

Table 1 summarizes the quality evaluation along the process. It reports the values computed for the quality indices previously introduced in Section 4.5. *NR* is the number of rules. *TRL* reports the total rule length. *COFCI* stands for co-firing based comprehensibility index. *MC* means misclassified

Table 1: FRBS quality along the design stages.

Quality Index	*Original FRBS*	*R5, R16, R31 R32 removal*	*FRBS after rule base simplification*	*FRBS after partition optimization*
NR	32	28	6	6
TRL	109	101	13	13
COFCI	0.065	0.075	0.787	0.787
MC Training	0.932	0.910	0.932	0.94
Coverage Training	100	98.496	100	100
MC Test	0.867	0.867	0.911	0.911
Coverage Test	97.778	97.778	100	100



Name: WINE

Inputs: Alcohol, MalicAcid, Ash, AlcalinityOfAsh, Magnesium, TotalPhenols, flavanoids, NonflavanoidsPhenols, Proanthocyanins, ColorIntensity, Hue, OD280/OD315, Proline

Outputs: Class

Rules

R...	Ty...	Act...	If Alc...	AND...	AN...	AND ...	AND ...	AND ...	AND ...	AND No...	AND P...	AND ...	AN...	AND ...	AND...	THE...
1	I	yes	very ...						very ...	very low		very l...				2.0
2	I	yes	very ...						very ...	low		very l...				2.0
3	I	yes	very ...						very ...	average		very l...				3.0
4	I	yes	very ...						very ...	high		very l...				3.0
5	I	yes	very ...						very ...	very high		very l...				2.0
6	I	yes	low ...						very ...			very l...				2.0

Goodness Threshold (high): 0.5

Goodness Threshold (low): 0.1

PathFinder Threshold : 0.2

Parameter Q : 11

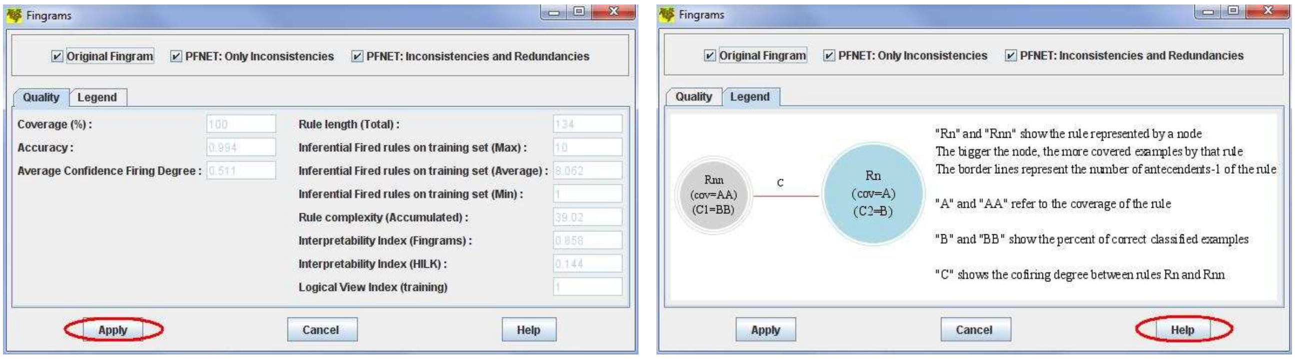
Choose one method for Fingrams graphical layout: neato

Figure 4: Generation of fingrams.

cases. *MC* and *Coverage* are computed regarding both training and test datasets. The column entitled as *Original FRBS* corresponds to the current FRBS (the one displayed in Fig. 4). The rest of columns are related to the next design steps which will be discussed in the rest of this section.

Once the *Original FRBS* is generated, we can use fingrams with the aim of uncovering the FRBS behavior at inference level through a graphical analysis. First, we must set some parameters (see pictures on the right side in Fig. 4):

- **Goodness Threshold**. Upper and lower thresholds for estimating the goodness of coverage regarding each single rule. The goodness measure informs about how well each rule classifies the problem instances that it covers. A rule covers one problem instance when the rule firing degree for that instance is greater than a predefined threshold (0.1 in this work).
- **Pathfinder Threshold**. This parameter is used for pruning the initial graph (removing those edges with weights smaller or equal than the threshold), before running Pathfinder.
- **Q**. This is the specific parameter of Pathfinder



☑ Original Fingram ☑ PFNET: Only Inconsistencies ☑ PFNET: Inconsistencies and Redundancies

Coverage (%) : 100

Accuracy : 0.994

Average Confidence Firing Degree : 0.511

Rule length (Total) : 134

Inferential Fired rules on training set (Max) : 10

Inferential Fired rules on training set (Average) : 8.062

Inferential Fired rules on training set (Min) : 1

Rule complexity (Accumulated) : 39.02

Interpretability Index (Fingrams) : 0.858

Interpretability Index (HILK) : 0.144

Logical View Index (training) 1

(a) Quality view.

☑ Original Fingram ☑ PFNET: Only Inconsistencies ☑ PFNET: Inconsistencies and Redundancies

"Rn" and "Rnn" show the rule represented by a node
The bigger the node, the more covered examples by that rule
The border lines represent the number of antecendents-1 of the rule

"A" and "AA" refer to the coverage of the rule

"B" and "BB" show the percent of correct classified examples

"C" shows the cofiring degree between rules Rn and Rnn

(b) Legend view.

Figure 5: GUAJE fingrams window.

which limits the number of links in the paths respecting the triangle inequality. GUAJE suggests assigning to Q the maximum number of rules that can be simultaneously fired, which is estimated in an inferential way regarding the available dataset. In consequence, the network scaling will take shorter time. Anyway, by default $Q = N - 1$, with the aim of assuring that all paths are properly analyzed.

Second, we have to choose one layout algorithm (bottom right side in Fig. 4) among those provided by GUAJE (*neato*, *fdp*, *circo*, and so on). We have chosen *neato* which is an implementation of the Kamada-Kawai algorithm.

Then, the GUAJE window of Fig. 5(a) appears. The body of the window is structured in the form of a tabbed panel. The *Quality* tab gives an overview of the quality of the designed FRBS. It provides a list of quality indices[m] regarding both accuracy (on the left) and interpretability (on the right). Moreover, the user can interpret fingrams according to the information presented in the *Legend* tab (Fig. 5(b)).

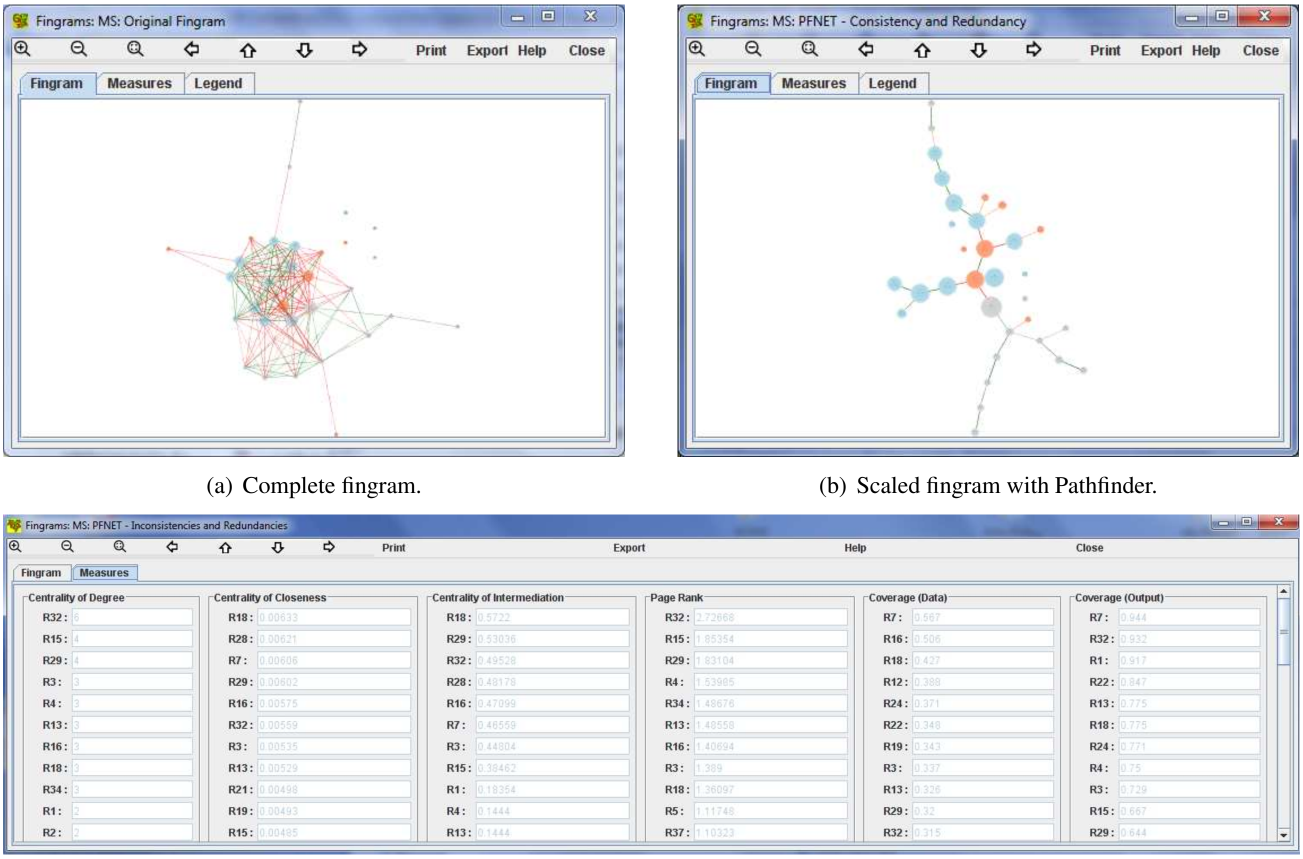
Once selected the proper options, the pictures in Fig. 6 are displayed. Thanks to the use of SVG format the user can interact with the graph through zooming, moving, and/or exploring in depth some zones of interest in the entire network. In addition, when the user passes the mouse over a node or an edge, a text pops up with the linguistic description of the related rule or link. Moreover, the user can disable rules by clicking on its corresponding node, i.e., a rule is temporally deactivated in the rule base, and the fingram is generated again without taking care of that rule. In consequence, fuzzy systems design becomes an interactive process which is effectively guided by decisions drawn from the expert analysis of fingrams.

The GUAJE window for fingram analysis is illustrated in Fig. 6. In addition to the visualization panel, it contains other two tabs: the *Legend* tab with a specific legend of the fingram presented; and the *Measures* tab (Fig. 6(c)) which gives several rule rankings based on some of the most popular measures in the context of social network analysis, such as *Page Rank* or *Centrality*.

The complete fingram is usually quite dense and difficult to analyze as it can be appreciated in Fig. 6(a). Thus, the network scaling stage becomes essential to provide a good and efficient fingram analysis. Therefore, GUAJE uses Pathfinder[n] to keep only the most significant links, yielding as result the scaled fingram depicted in Fig. 6(b). Looking carefully at the scaled fingram, it is easy to detect rules that cover regions with a few examples and are therefore good candidates to be studied in detail. For instance, rules R5, R16, R31 and R32 appear iso-

[m] Such indices are thoroughly explained in the GUAJE user manual.

[n] GUAJE actually makes use of MST-PathFinder [32]. It is a recently published variant of Pathfinder algorithm able to generate large science maps in cubic time.


Fingrams: MS: Original Fingram
Print
Export
Help
Close
Fingram
Measures
Legend
Fingrams: MS: PFNET - Consistency and Redundancy
Print
Export
Help
Close
Fingram
Measures
Legend
Fingrams: MS: PFNET - Inconsistencies and Redundancies
Print
Export
Help
Close
Fingram
Measures

Centrality of Degree	Centrality of Closeness	Centrality of Intermediation	Page Rank	Coverage (Data)	Coverage (Output)
R32: 6	R18: 0.00633	R18: 0.5722	R32: 2.72668	R7: 0.567	R7: 0.944
R15: 4	R28: 0.00621	R29: 0.53036	R15: 1.85354	R16: 0.506	R32: 0.932
R29: 4	R7: 0.00606	R32: 0.49528	R29: 1.83104	R18: 0.427	R1: 0.917
R3: 3	R29: 0.00602	R28: 0.48178	R4: 1.53985	R12: 0.388	R22: 0.847
R4: 3	R16: 0.00575	R16: 0.47099	R34: 1.48676	R24: 0.371	R13: 0.775
R13: 3	R32: 0.00559	R7: 0.46559	R13: 1.48558	R22: 0.348	R18: 0.775
R16: 3	R3: 0.00535	R3: 0.44804	R16: 1.40694	R19: 0.343	R24: 0.771
R18: 3	R13: 0.00529	R15: 0.38462	R3: 1.389	R3: 0.337	R4: 0.75
R34: 3	R21: 0.00498	R1: 0.18354	R18: 1.36097	R13: 0.326	R3: 0.729
R1: 2	R19: 0.00493	R4: 0.1444	R5: 1.11748	R29: 0.32	R15: 0.667
R2: 2	R15: 0.00485	R13: 0.1444	R37: 1.10323	R32: 0.315	R29: 0.644

(a) Complete fingram.

(b) Scaled fingram with Pathfinder.

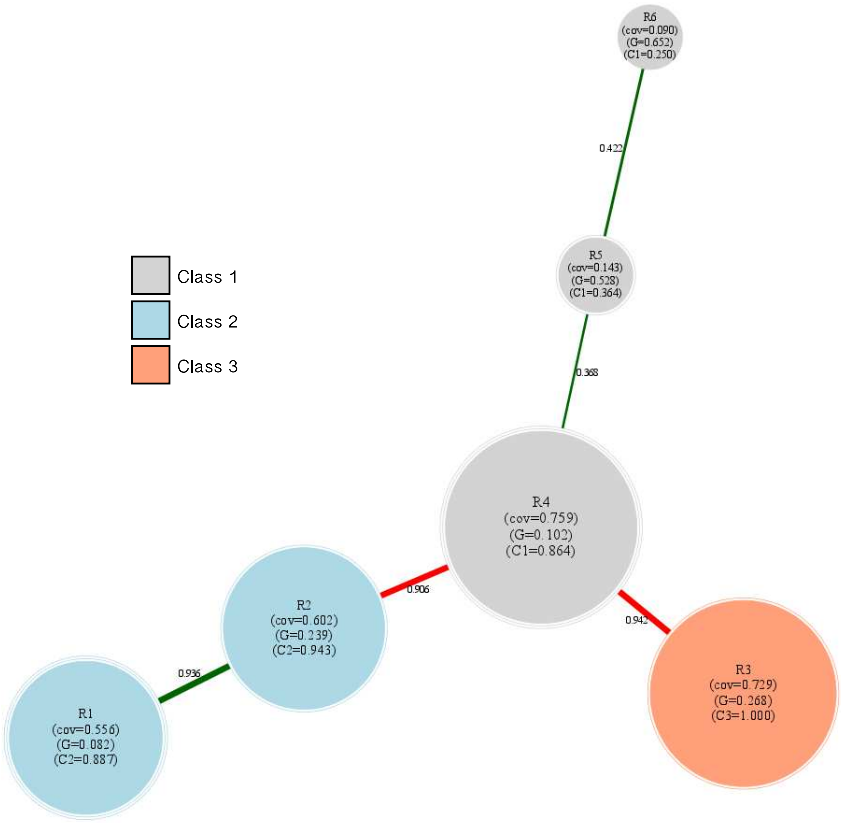
(c) Rule rankings.

Figure 6: Fingram analysis window.

lated and they cover very few examples. Hence, they are good candidates to be removed. In fact, as shown in Table 1, after removing such rules accuracy only slightly decreases.

In the quest for even better interpretability-accuracy trade-off we have opted for applying first linguistic simplification (aimed at improving interpretability while preserving accuracy) and then partition tuning (with the goal of increasing accuracy without jeopardizing interpretability).

The linguistic simplification process yields a more compact FRBS. The number of variables is reduced from 13 to 4, the total number of membership functions goes down from 65 to 11, the number of rules drops from 28 to 6, and the total rule length pass from 101 to 13. This is because simplified rules are much more general than the original ones. The result of simplification can be seen in a new fingram which is depicted in Fig. 7. At first sight it is clear the high level of simplification obtained with a drastically reduction of the number of rules (as a side effect the *COFCI* index improves). Even more, the rules cover all the examples (regarding both training and test sets) because they are more general, and they also produce better accuracy (as it is detailed in Table 1). The rule R3 covers all examples related to class C3. In addition, it significantly overlaps with rule R4, thus yielding potential inconsistencies. Rule R4 turns up as the central rule in the fingram. It covers most data samples going in conflict against R3 but also against R2. Even though rules R2 and R4 cover many examples they are not enough to handle properly their related classes, C2 and C1 respectively. Therefore, they have to be com-



R1: IF Alcohol is Very Low AND Flavanoids is Very Low AND Color Intensity is Very Low THEN Class is C2

R2: IF Alcohol is Very Low AND Flavanoids is Low THEN Class is C2

R3: IF Flavanoids is Very Low AND Color Intensity is Low THEN Class is C3

R4: IF Alcohol is Low AND Flavanoids is Low AND Proline is (Low OR Average) THEN Class is C1

R5: IF Alcohol is (Average OR High) AND Flavanoids is Low THEN Class is C1

R6: IF Flavanoids is Average THEN Class is C1

Figure 7: Fingram of the simplified FRBS.

plemented with other rules, partly redundant with them, like R1, R5 and R6.

With respect to the optimization stage, we have selected the Solis-Wets algorithm able to tune the membership functions yielding marginal changes in the overall behavior of the system (look at quality indices reported in Table 1). For this particular case study, the tuning process only affects to the definition of the linguistic terms *Very Low* and *Low* of variable *Alcohol*. Therefore, only rules R1, R2 and R4 suffer slightly changes. In consequence, the resultant fingram remains almost the same than after simplification.

Finally, it is worthy to remark that GUAJE comes with several intuitive and interactive tutorials. One of them shows the benefits and potentials of fingrams for aiding the design of FRBSs. It details, step by step, first how to build an interpretable fuzzy rule-based classifier and then how to simplify and optimize it, looking for the best balance between accuracy and interpretability supported by fingrams. The illustrative case study presented above can be reproduced by the interested reader with the help of GUAJE and just following the related tutorial.

6. Conclusions and future works

This paper has explained how the software module for fingram generation and analysis is successfully integrated with the rest of modules provided by GUAJE. The new module is a powerful tool for understanding the system behavior at inference level. It becomes really useful in the design of fuzzy systems because the designer can check graphically, at any design stage (rule learning, simplification, optimization, etc.), the interaction between rules but also how each design decision affects to the interpretability-accuracy trade-off.

Please, notice that GUAJE is freely available (under GPL license) as open source software at:

http://www.softcomputing.es/guaje

The number of users of GUAJE is growing up all around the world. The users' feedback helps us to continue enhancing this free software tool. Thus, new releases will be available with improvements in the visualization of fingrams. In short-term we want to make even more dynamic and user-friendly the interaction with the user. Thus, he/she should be able to alter the graph layout through making drag and drop of some nodes or collapse/expand parts of interest in the entire graph. Regarding mid-term future we plan to develop a new software module that gives fully automatic support to the design of fuzzy systems guided by fingrams.

The co-firing measure presented in this contribution is biased by the training data. Such fact may be avoided by considering other measures of overlapping between rules, that will be part of our future work.

Acknowledgments

This work has been funded by the "Spanish Ministry of Economy and Competitiveness" under projects TIN2011-29824-C02-01 and TIN2011-29824-C02-02. It has also been partly supported by the European Centre for Soft Computing.

References

1. G. Acampora, V. Loia, and A. V. Vasilakos, "Autonomous composition of fuzzy granules in ambient intelligence scenarios," *Human-Centric Information Processing Through Granular Modelling*, Studies in Computational Intelligence, Springer, 265–287 (2009).
2. J. Alcalá-Fdez, L. Sanchez, S. Garcia, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernandez, and F. Herrera, "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, **13(3)**, 307–318 (2009).
3. J. M. Alonso, O. Cordón, S. Guillaume, and L. Magdalena, "Highly interpretable linguistic knowledge bases optimization: Genetic Tuning versus Solis-Wetts. Looking for a good interpretability-accuracy trade-off," *Proc. IEEE International Conference on Fuzzy Systems*, 901–906 (2007).
4. J. M. Alonso, O. Cordón, A. Quirin, and L. Magdalena, "Analyzing interpretability of fuzzy rule-based systems by means of fuzzy inference-grams," *Proc. World Conference on Soft Computing*, San Francisco State University, 181.1–181.8 (2011).
5. J. M. Alonso and L. Magdalena, "Editorial: special issue on interpretable fuzzy systems," *Information Sciences*, **181(20)**, 4331–4339 (2011).
6. J. M. Alonso and L. Magdalena, "Generating understandable and accurate fuzzy rule-based systems in a Java environment," *Lecture Notes in Artificial Intelligence - Proc. 9th International Workshop on Fuzzy Logic and Applications*, **LNAI6857**, 212–219 (2011).
7. J. M. Alonso and L. Magdalena, "HILK++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers," *Soft Computing*, **15(10)**, 1959–1980 (2011).
8. J. M. Alonso, L. Magdalena, and S. Guillaume, "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 (2008).
9. J. M. Alonso, D. P. Pancho, O. Cordón, A. Quirin, and L. Magdalena, "Social network analysis of co-fired fuzzy rules". In R. R. Yager, A. M. Abbasov, M. Z. Reformat, and S. N. Shahbazova, editors, *Soft Computing: State of the Art Theory and Novel Applications*, *Studies in Fuzziness and Soft Computing*, Springer Berlin Heidelberg, **291**, 113–128 (2013).
10. J. M. Alonso, D. P. Pancho, and L. Magdalena, "Enhancing the fuzzy modeling tool GUAJE with a new module for fingrams-based analysis of fuzzy rule bases," *Proc. IEEE International Conference on Fuzzy Systems*, 1082–1089 (2012).
11. I. Baturone, F. J. Moreno-Velo, S. Sánchez-Solano, A. Barriga, P. Brox, A. A. Gersnoviez, and M. Brox, "Using Xfuzzy environment for the whole design of fuzzy systems," *Proc. IEEE International Conference on Fuzzy Systems*, 517–522 (2007).
12. I. Baturone, F. J. Moreno-Velo, S. Sánchez-Solano, and A. Ollero, "Automatic design of fuzzy controllers for car-like autonomous robots," *IEEE Transactions on Fuzzy Systems*, **12(4)**, 447–465 (2004).
13. M. R. Berthold, "Mixed fuzzy rule formation," *International Journal of Approximate Reasoning*, **32**, 67–84 (2003).
14. M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, "KNIME - the Konstanz information miner (version 2.0 and beyond)," *ACM SIGKDD Explorations Newsletter*, **11**, 26–31 (2009).
15. C. Borgelt and G. González-Rodríguez, "FrIDA - a

free intelligent data analysis toolbox," *Proc. IEEE International Conference on Fuzzy Systems*, 1892–1896 (2007).

16. J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, "Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview," In J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, editors, *Interpretability issues in fuzzy modeling*, Studies in Fuzziness and Soft Computing, Springer-Verlag, Heidelberg, **128**, 3–22 (2003).
17. P. Cingolani and J. Alcala-Fdez, "jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation," *Proc. IEEE International Conference on Fuzzy Systems*, 1090–1097 (2012).
18. D. W. Dearholt and R. W. Schvaneveldt, "Properties of Pathfinder networks," In R. Schvaneveldt, editor, *Pathfinder associative networks: studies in knowledge organization*, Ablex Publishing Corporation, 1–30 (1990).
19. P. Y. Glorennec, "*Algorithmes d'apprentissage pour systemes d'inference floue (French Edition)*," Editions Hermès, Paris (1999).
20. S. Guillaume and B. Charnomordic, "Generating an interpretable family of fuzzy partitions," *IEEE Transactions on Fuzzy Systems*, **12(3)**, 324–335 (2004).
21. S. Guillaume and B. Charnomordic, "Learning interpretable fuzzy inference systems with FisPro," *Information Sciences*, **181(20)**, 4409–4427 (2011).
22. S. Guillaume and B. Charnomordic, "Fuzzy inference systems: an integrated modeling environment for collaboration between expert knowledge and data using FisPro," *Expert Systems with Applications*, **39(10)**, 8744–8755 (2012).
23. H. Hellendoorn and D. Driankov, "*Fuzzy model identification*," Springer-Verlag London, UK (1997).
24. E. Hüllermeier, "Fuzzy methods in machine learning and data mining: status and prospects," *Fuzzy Sets and Systems*, **156**, 387–406 (2005).
25. H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi, "Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning," *Fuzzy Sets and Systems*, **81**, 157–167 (1996).
26. T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, **31**, 7–15 (1989).
27. E. H. Mamdani, "Application of Fuzzy Logic to approximate reasoning using linguistic systems," *IEEE Transactions on Computers*, **26(12)**, 1182–1191 (1977).
28. F. J. Moreno-Velo, A. Barriga, S. Sánchez-Solano, and I. Baturone, "XFSML: an XML-based modeling language for fuzzy systems," *Proc. IEEE International Conference on Fuzzy Systems*, 1146–1153 (2012).
29. F. Moya-Anegón, B. Vargas-Quesada, V. Herrero-Solana, Z. Chinchilla-Rodríguez, E. Corera-Álvarez, and F. Muñoz-Fernández, "A new technique for building maps of large scientific domains based on the cocitation of classes and categories," *Scientometrics*, **61**, 129–145 (2004).
30. D. P. Pancho, J. M. Alonso, O. Cordón, A. Quirin, and L. Magdalena, "Fingrams: Visual representations of fuzzy rule-based inference for expert analysis of comprehensibility," *IEEE Transactions on Fuzzy Systems*, **21(6)**, doi:10.1109/TFUZZ.2013.2245130 (2013).
31. J. R. Quinlan, "Induction of decision trees," *Machine Learning*, **1**, 81–106 (1986).
32. A. Quirin, O. Cordón, J. Santamaría, B. Vargas-Quesada, and F. Moya-Anegón, "A new variant of the Pathfinder algorithm to generate large visual science maps in cubic time," *Information Processing and Management*, **44**, 1611–1623 (2008).
33. E. H. Ruspini, "A new approach to clustering," *Information and Control*, **15(1)**, 22–32 (1969).
34. F. J. Solis and R. J. B. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, **6(1)**, 19–30 (1981).
35. B. Vargas-Quesada and F. Moya-Anegón, "*Visualizing the structure of science*," Springer-Verlag (2007).
36. L. A. Zadeh, "Fuzzy sets," *Information and Control*, **8**, 338–353 (1965).
37. L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man and Cybernetics*, **3**, 28–44 (1973).
38. L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Parts I, II, and III. Information Sciences*, **8, 8, 9**, 199–249, 301–357, 43–80 (1975).