Understanding the Inference Mechanism of FURIA by means of Fingrams

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

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

Abstract

This paper shows the use of Fingrams –Fuzzy Inference-grams– aimed at unveiling graphically some hidden details in the usual behavior of the precise fuzzy modeling algorithm FURIA –Fuzzy Unordered Rule Induction Algorithm–. FURIA is recognized as one of the most outstanding fuzzy rule-based classification methods attending to accuracy. Although FURIA usually produces compact rule bases, with low number of rules and antecedents per rule, its interpretability is arguable, being penalized by the absence of linguistic readability and a complex inference mechanism. Fingrams offer a methodology for visual representation and exploratory analysis of fuzzy rule-based systems. FURIA-Fingrams, i.e. fuzzy inference-grams representing fuzzy systems learnt with FURIA, make easier understanding the FURIA inference mechanism thanks to the possibilities they offer: detecting instances not covered by any rule; highlighting important rules; clarifying the so-called stretching mechanism; etc.

Keywords: Interpretability, Inference, Precise Fuzzy Modeling, Visual Analysis

1. Introduction

Interpretability is recognized as an appreciated and valuable capability of fuzzy systems. It represents the capability of fuzzy systems to model the operation of real systems in a human comprehensible way [1, 2]. Therefore, it becomes an essential requirement for those applications that involve extensive interaction with human beings. E.g., decision support systems in medicine [3, 4] must be understandable, for both physicians and patients, with the intention of being widely accepted and successfully applicable. Unfortunately, fuzzy systems are not interpretable per se; they have to be designed carefully to fulfill that characteristic.

There are two main approaches when modeling fuzzy rule-based systems (FRBSs): producing linguistic or precise fuzzy modeling [5].

On the one hand, linguistic fuzzy modeling yields fuzzy rules composed of linguistic variables [6] taking terms with a real-world meaning [7]. Therefore linguistic fuzzy modeling favors interpretability.

On the other hand, precise fuzzy modeling constructs fuzzy systems that prioritize accuracy at the cost of jeopardizing interpretability, specially making harder the understanding of the system behavior at inference level [8]. They usually deal with weighted rules, advanced defuzzification strategies, a high number of rules, variables or antecedents per rule, etc.

Along the years different alternatives have been proposed to visualize data mining models [9]. Visual representations contain rich information to understand the behavior and characteristics of models, supporting their comprehension and decision-making.

The visual representation of fuzzy systems permits the user to obtain and analyze elements of the systems in meaningful and informative ways. There are not many papers tackling with visual analysis of fuzzy systems. Buck et al. presented in [10] visual representations to deal with fuzzy numbers and fuzzy vectors. Pham et al. provided in [11] a complete analysis of visualization requirements for fuzzy systems. Gabriel et al. proposed in [12] to visualize and explore multi-dimensional FRBSs in a 2D graphical representation including data samples and rules. Pancho et al. developed a methodology to represent fuzzy rule interaction at inference level through fuzzy inference-grams (Fingrams) [13].

Fingrams have arisen as a powerful tool for visualizing and analyzing FRBSs. Fingrams give a global view of fuzzy systems, and allow us to understand its behavior at a high level of abstraction. They present fuzzy systems as graphs where rules are related each other reflecting how they cover the input space. Different metrics and visual artifacts have been proposed to reflect the particularities of different kinds of fuzzy systems. Namely, FRBSs for classification and regression [13] but also fuzzy association rules [14]. It is worthy to note the capability of Fingrams to graphically depict the inference mechanism of fuzzy systems. Fingrams let us visualize importance and complexity of rules, how instances are covered by rules, how many instances are not covered, and so on.

This paper shows how the use of Fingrams can help to understand over the precise fuzzy modeling algorithm FURIA [15]. FURIA –abbreviation of Fuzzy Unordered Rule Induction Algorithm– is one
of the most outstanding fuzzy rule-based classification methods attending to accuracy. Even though FURIA produces compact rule bases, with low number of rules and antecedents per rule, its interpretability is arguable. FURIA is penalized by the absence of linguistic readability because it lacks of global semantics. On the contrary, rule antecedents are specific for each rule and they do not have linguistic terms associated. In addition the inference mechanism of FURIA occludes interpretability. It is based on a winner class mechanism with weighted rules in combination with the so-called rule stretching method which is in charge of handling uncovered instances. In consequence, it produces a close-to-black-box inference mechanism, very hard to predict and understand.

The rest of the manuscript is organized as follows. Section 2 summarizes the characteristics of FURIA and Fingrams. Section 3 goes in depth with the analysis of FURIA through Fingrams over two illustrative examples. The first example introduces the particularities of FURIA-Fingrams (Fingrams depicting fuzzy systems learnt by FURIA) in a high dimensional problem. The second example focuses on the stretching mechanism of FURIA with a case of use that intensively takes advantage of it. Finally, some conclusions and future work are pointed out in Section 4.

2. Preliminaries

2.1. FURIA

FURIA is a precise fuzzy modeling algorithm recognized world-wide as one of the most accurate fuzzy classification rule learning algorithms [15]. Its popularity has grown during last years with more than 100 works citing the original publication and demonstrating to be a robust method, performing properly in a bunch of scenarios [16, 17].

FURIA learning method follows RIPPER building strategy [18]. Nevertheless, differently from RIPPER, FURIA manages fuzzy rules instead of crisp ones, considers the order of rules irrelevant, and does not construct a default rule.

FURIA rule induction algorithm starts with the fuzzification of rule antecedents provided by RIPPER, passing from crisp intervals to trapezoidal fuzzy sets. Thus, the original crisp intervals determine the cores of the new fuzzy sets while supports are extended trying to maximize the coverage of data instances concordant with rule output. Then, antecedents are ranked according to their relative importance. The final rule format is as follows:

\[ R_i : \text{IF } X_1 \text{ is } A_1^i \text{ AND ... AND } X_n \text{ is } A_n^i \text{ THEN } Y \text{ is } B^i \left( w^i \right) \]  

(1)

where \( A_h^i \) is the rule antecedent for variable \( X_h \) \((h \in [1, n])\) and it is defined by the four points that are characteristics of a trapezoidal fuzzy set, \( A_h^i = \{ a_h^i, b_h^i, a_h^i, a_h^i \} \). \( a_h^i \) and \( a_h^i \) are the bounds of the core \( \mu_{A_h^i} (x) = 1 \) if \( x \in [a_h^i, a_h^i] \) while \( a_h^i \) and \( a_h^i \) limit the support \( \mu_{A_h^i} (x) > 0 \) if \( x \in (a_h^i, a_h^i) \). \( B^i \) denotes the rule output class. \( w^i \) is the weight associated to rule \( R_i \). FURIA rules are weighted according to their Certainty Factor (CF) which reflects the proportion of data instances correctly covered, i.e. instances in concordance with rule output, with respect to the total instances covered by the rule.

Given a data instance, the inference mechanism operates differently if the instance is covered or not by the set of induced rules:

- In case the instance is covered by the set of induced rules, FURIA predicts as output the winner class coming from the sum of activation degrees (weighted by CF) of all induced rules per class.
- Otherwise, FURIA dynamically creates a new set of rules from the induced ones, taking advantage of the so-called rule stretching mechanism. It checks rule by rule the whole set of induced rules for the given instance. For each induced rule, antecedents are removed from the least to the most important one, passing to the analysis of the next rule when the instance is covered by the stretched rule or there are no more antecedents to remove. Notice that importance of antecedents is implicit in the order in which they appear in each rule as it was given by the rule induction mechanism in FURIA. If all antecedents were removed for an individual rule, then such rule would become an empty rule and it would be discarded. On the contrary, the new rule would be added to the set of stretched rules.

The new stretched rules are weighted according to the Certainty Factor of the original rules and the number of antecedents kept after finishing the stretching procedure. The system predicts as output the class given by the winner rule in the new rule set. However, if all rules were discarded by the stretching mechanism, then the class with the highest frequency in the dataset would be taken as output.

FURIA creates compact FRBSs that achieve high performance thanks to its specific inference mechanism. Unfortunately, the comprehension of such inference mechanism is not straightforward although it is a key issue to properly interpret the behavior of systems built up with FURIA.

The interested reader can find a deeper explanation of FURIA in [15].

2.2. Fingrams

Fingrams are graphs formed by nodes and edges that overview at a glance the complete inference process of fuzzy systems [13]. Rules are represented by nodes with size proportional to the number of
instances covered, and edges reflect relations between rules, i.e. how pairs of rules jointly cover the input space; thus the larger the number of instances commonly covered, the stronger the relation.

The relations are calculated according to a metric that reflects how rules cover the instances in a given dataset. The simplest metric (as shown in eq. 2) relates two rules ($R_i$ and $R_j$) according to the number of instances covered in common by them ($|D_i \cap D_j|$) with respect to the total number of instances they individually cover ($|D_i|$ and $|D_j|$) where $D_i$ and $D_j$ are the set of instances covered by rules $R_i$ and $R_j$ respectively, and $|.|$ represents the cardinality of sets.

$$m_{ij} = \frac{|D_i \cap D_j|}{\sqrt{|D_i||D_j|}}, \ m_{ij} \in [0,1]$$ (2)

Due to the usual high interaction between rules in fuzzy systems, Fingrams usually appear highly dense and complex to analyze. Therefore, a suitable filtering of elements is demanded to produce a clearer graph where its backbone emerges. We take advantage of Pathfinder scaling algorithm [19] which maintains all the nodes but only the most relevant links looking at proximity between pairs of nodes.

A pleasant graphical representation is quite important to easily identify and understand the behavior of the FRBS under study. Kamada-Kawai algorithm [20] layouts Fingram elements in 2D, following aesthetical criteria.

Fingrams can already deal with FURIA, fuzzy association rules [14], fuzzy rule-based classifiers and regressors [13]. The different adaptations involve specific metrics and show relevant information according to their characteristics.

A specific software, Fingrams Generator\(^1\) [21], permits the creation of Fingrams no matter how the depicted FRBS was generated. Also, a few software tools already allow the creation and analysis of Fingrams, such as the fuzzy modeling tool GUAJE [22] or the data mining suites KEEL [23] and KNIME [24].

### 3. Analyzing FURIA through Fingrams

This section sketches the use of Fingrams to analyze and comprehend FURIA.

Hühn and Hüllermeier summarized in [15, 25] the experimentation done with FURIA over 45 datasets. They demonstrated that FURIA outperforms the precision of other algorithms, but nothing is mentioned about the systems interpretability. It is worthy to note that FURIA includes elaborated tricks, mainly in its inference mechanism, which hamper its comprehension.

Let us consider Fingrams to yield some light on how FURIA actually works in practice. FURIA-Fingrams are compound by two graphical representations. The first representation shows the set of induced rules while the second one presents the set of stretched rules. Moreover, a rectangular node depicts the data instances not covered by any rule in both representations. It should be noticed that uncovered instances penalize the precision of FRBSs and their early detection is essential for the correct design of the system.

In the remainder of this section, we analyze through Fingrams two systems constructed by FURIA from two of the datasets considered in [15]. The first system is learnt from a high dimensional and balanced dataset. The second system intensively takes advantage of the FURIA stretching mechanism.

#### 3.1. Understanding the FURIA inference mechanism in a high dimensional problem

For this case study, we selected the dataset known as synthetic-control (dataset 40 in [15]). It contains 600 instances of control charts synthetically generated. Each data instance includes 61 attributes and belongs to one of the six established classes (Normal, Cyclic, Increasing trend, Decreasing trend, Upward shift and Downward shift).

According to experimental results reported in [15] for the dataset under study: FURIA obtained an average classification rate of 89.75%, the average number of rules was 15.9, and the average number of antecedents per rule was 2.7.

We run FURIA over the whole dataset and the algorithm induced the list of rules given in Table 1. The rule base is made up of 20 rules which involve in the antecedents only 33 out of all the 61 given attributes. These rules are apparently very simple and they are quite accurate as can be deduced from the fact that there are rules handling all target output classes and most rules have CF higher than 0.9 (minimum CF is actually 0.86). Nevertheless, understanding the system behavior just reading and interpreting these rules is not easy. Firstly, the generated trapezoidal fuzzy sets are specific for each rule and they have no linguistic terms attached. This fact strongly hampers the readability of the rule base. Secondly, having a global view of the input space gets away from human capabilities, due to the huge number of attributes. This makes unfeasible to figure out a representation of the given dataset in 2D or 3D. The same difficulties arise when thinking about identifying main interaction among rules. In consequence, trying to guess the output class predicted by this rule base for a given data instance is not a straightforward task.

Fig. 1 presents the Fingram depicting the induced rule set given in Table 1.

Edges between rule nodes indicate rule interaction and they are computed by equation 2. Green edges relate rules of same output class whereas red edges relate rules of different output class. Notice that studying the graph structure in detail we observe how all edges but one (edge relating R2 and

\(^1\)http://www.sourceforge.net/projects/fingrams/
The synthetic-control dataset.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Antecedents</th>
<th>Then Class</th>
<th>Certainty Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>ant1</td>
<td>Decreasing trend (CF=0.940)</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>ant2</td>
<td>Increasing trend (CF=0.978)</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>ant3</td>
<td>Cyclic (CF=0.982)</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>ant4</td>
<td>Downward shift (CF=0.938)</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>ant5</td>
<td>Normal (CF=0.953)</td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td>ant6</td>
<td>Upward shift (CF=0.983)</td>
<td></td>
</tr>
</tbody>
</table>

There is a special node, labeled as “UNCOVERED INSTANCES”, which shows in a striped chart the proportion of instances of each class not covered by the induced rule base. This node size is proportional to the total number of uncovered instances. In this problem, only 7 out of the 600 instances (cov=0.012) are uncovered (1 of class Decreasing trend, 2 of Downward shift, 2 of Cyclic, 1 of Increasing trend, 1 of Upward shift). In order to handle the uncovered instances, FU-RIA triggers the stretching mechanism. Table 2 shows the textual description of the stretched rules and Fig. 2 shows the related Fingram. Notice that it is slightly different from the previous one.

Firstly, the stretched rule identifier makes reference to the identifier of the originating induced rule, i.e. the induced rule from which the stretched rule is derived. Thus, stretched rules are named RXX.YY with RXX being the identifier of the originating rule and YY being the number of antecedents in the rule.

Figure 1: Fingram of the set of rules induced for the synthetic-control dataset.

Table 1: Textual description of the induced rule base for the synthetic-control dataset.
Table 2: Textual description of the set of stretched rules for the synthetic-control dataset.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Antecedents</th>
<th>Consequent</th>
</tr>
</thead>
<tbody>
<tr>
<td>R3.3</td>
<td>IF col60 in [-∞,-∞], 15.02, 15.14</td>
<td>THEN class is Decreasing trend (CF=0.272)</td>
</tr>
<tr>
<td>R5.1</td>
<td>IF col44 in [-∞,-∞], 16.43, 19.85</td>
<td>THEN class is Decreasing trend (CF=0.267)</td>
</tr>
<tr>
<td>R7.3</td>
<td>IF col47 in [-∞,-∞], 23.41, 23.47 AND col21 in [25.86, 26.11, -∞, -∞] AND col16 in [23.45, 24.86, -∞, -∞]</td>
<td>THEN class is Downward shift (CF=0.573)</td>
</tr>
<tr>
<td>R8.1</td>
<td>IF col54 in [-∞,-∞], 23.92, 24.25</td>
<td>THEN class is Downward shift (CF=0.137)</td>
</tr>
<tr>
<td>R8.3</td>
<td>IF col54 in [-∞,-∞], 23.92, 24.25 AND col10 in [28.87, 29.35, -∞, -∞] AND col57 in [14.40, 15.02, -∞, -∞]</td>
<td>THEN class is Downward shift (CF=0.501)</td>
</tr>
<tr>
<td>R11.3</td>
<td>IF col55 in [41.65, 42.00, -∞, -∞] AND col20 in [33.54, 33.65, -∞, -∞] AND col21 in [31.43, 31.53, -∞, -∞]</td>
<td>THEN class is Increasing trend (CF=0.636)</td>
</tr>
<tr>
<td>R12.1</td>
<td>IF col54 in [42.32, 43.71, -∞, -∞]</td>
<td>THEN class is Increasing trend (CF=0.230)</td>
</tr>
<tr>
<td>R16.1</td>
<td>IF col46 in [-∞,-∞], 30.02, 36.95</td>
<td>THEN class is Normal (CF=0.078)</td>
</tr>
<tr>
<td>R16.2</td>
<td>IF col44 in [-∞,-∞], 30.02, 36.95 AND col57 in [23.76, 24.11, -∞, -∞]</td>
<td>THEN class is Normal (CF=0.292)</td>
</tr>
<tr>
<td>R16.3</td>
<td>IF col44 in [-∞,-∞], 30.02, 36.95 AND col57 in [23.76, 24.11, -∞, -∞] AND col55 in [-∞,-∞], 32.84, 33.15</td>
<td>THEN class is Normal (CF=0.512)</td>
</tr>
<tr>
<td>R17.1</td>
<td>IF col55 in [27.69, 33.23, -∞, -∞]</td>
<td>THEN class is Normal (CF=0.038)</td>
</tr>
<tr>
<td>R17.2</td>
<td>IF col55 in [27.69, 33.23, -∞, -∞] AND col17 in [-∞,-∞], 31.99, 32.54</td>
<td>THEN class is Normal (CF=0.089)</td>
</tr>
<tr>
<td>R17.3</td>
<td>IF col55 in [27.69, 33.23, -∞, -∞] AND col17 in [-∞,-∞], 31.99, 32.54 AND col52 in [24.37, 25.35, -∞, -∞]</td>
<td>THEN class is Normal (CF=0.532)</td>
</tr>
<tr>
<td>R18.2</td>
<td>IF col44 in [35.67, 35.80, -∞, -∞] AND col17 in [-∞,-∞], 33.68, 34.81</td>
<td>THEN class is Upward shift (CF=0.368)</td>
</tr>
<tr>
<td>R19.1</td>
<td>IF col42 in [36.01, 39.20, -∞, -∞]</td>
<td>THEN class is Upward shift (CF=0.229)</td>
</tr>
</tbody>
</table>

The second case study deals with the dataset called metStatRST (dataset 22 in [15]). It presents 336 instances with the mean values of weather conditions taken in different stations placed in Germany. It includes three attributes: average rainfall; sunshine duration; and temperature per year. Output class corresponds to the German State where the met station is located (12 classes/States).

According to results published in [15], this is a hard problem, with 12 highly unbalanced classes, where the best performance was 42.02% and it was achieved by the genetic learning algorithm called SLAVE [26]. FURIA produced a fuzzy system which performed poorly with only 33.56% correctly classified instances. The average number of rules was 15.9, and the average number of antecedents per rule was 2.7.

We run FURIA over the whole dataset and the algorithm induced the list of rules given in Table 3. The rule base is made up of 12 rules which involve all the 3 given attributes in the antecedents. These rules cover only 8 out of the 12 output classes. Even worse, CF is quite low (always under 0.87) for all rules. This means that covered instances are not predicted correctly.

Let us analyze rule interaction with the help of Fingrams in order to find out why FURIA does not work properly in this problem.

Fig. 3 presents the Fingram related to the set of induced rules given in Table 3. It is worthy to note how the global structure of this Fingram is very different from the ones described in the previous section. Induced rules are quite specific (individual coverage ≤ 0.101) and they mostly cover instances only 7 data instances. In addition, the coverages of stretched rules are very low due to the so small amount of given instances. Even worse, instances handled by the stretching mechanism (take a look at “UNCOVERED INSTANCES” node in Fig. 1) correspond to only 5 classes but the set of stretched rules produces 6 classes as output. This implies that class Normal is predicted incorrectly in some of the 7 instances.

3.2. Going in depth with the FURIA stretching mechanism

Figure 2: Fingram of the set of stretched rules for the synthetic-control dataset.

Figure 3 presents the Fingram related to the set of induced rules given in Table 3. It is worthy to note how the global structure of this Fingram is very different from the ones described in the previous section. Induced rules are quite specific (individual coverage ≤ 0.101) and they mostly cover instances only 7 data instances. In addition, the coverages of stretched rules are very low due to the so small amount of given instances. Even worse, instances handled by the stretching mechanism (take a look at “UNCOVERED INSTANCES” node in Fig. 1) correspond to only 5 classes but the set of stretched rules produces 6 classes as output. This implies that class Normal is predicted incorrectly in some of the 7 instances.
at their own, with very low interaction to each other. This fact produces very few edges and most nodes appear isolated.

Differently from previous case, here many instances are not covered by the induced rules. The “UNCOVERED INSTANCES” node is the largest one. Actually, more than half of the instances (171 out of 336 instances) are covered by none of the induced rules. So, they trigger the stretching mechanism. In consequence, the system performance strongly depends on the ability of the stretching mechanism to properly handle such instances. It generates 23 stretched rules. They are depicted in the Fingram given in Fig. 4. Notice that for the sake of space we have not included here their textual description since we have already proved in the previous examples that the interpretation of Fingrams conveys much richer information than that of the mere textual description.

Like in the previous case study, once the stretching mechanism ends there are not uncovered instances anymore. Thus, the “UNCOVERED INSTANCES” node disappears in Fig. 4. Again, stretched rules are more densely related than induced rules, meaning that they are more general and cover part of the input space in common. In addition, most relations connect rules with different output class, what is remarked through red edges in the graph.

Rules with just one antecedent (RXX.1) gets a central position in the graph because they are quite general and jointly cover instances with many others. For example, Rule R3.1 covers near half of the instances handled by the stretching mechanism. Even more, Rules R4.1 and R5.1 cover most of instances in common producing a high relation between them \( m_{5.1 \cdot 4.1} = 0.917 \).

Finally, we can conclude that FURIA fails to produce an accurate fuzzy system in this case because of several factors:

- Even though the dataset contains only three attributes, the classification problem becomes extremely hard mainly due to the big number of highly unbalanced output classes.
- Induced rules only cover a reduced part of the whole dataset. Even worse, most covered instances are not properly classified. Notice that
4 out of 12 classes are margined and not predicted.

- Half of the data instances have to be managed by the stretching mechanism. It succeeds to cover them; but it fails to predict the right output classes. Notice that stretched rules are directly derived from induced ones. Obviously, starting from a poor set of induced rules FURIA stretching mechanism is expected to perform poorly.

4. Conclusions and future work

This paper has introduced the use of Fingrams for dealing with FURIA, an outstanding precise fuzzy modeling algorithm which constructs accurate but hardly to interpret fuzzy rule-based classifiers.

We take advantage of the so-called FURIA-Fingrams that permit a comfortably visualization and analysis of fuzzy systems learnt by FURIA. FURIA-Fingrams are compound by a twofold visualization that presents on the one hand the set of induced rules and on the other hand the set of stretched rules.

Additionally, a visual artifact represents instances not covered by the given set of rules. Notice that the detection and analysis of uncovered instances is key in fuzzy modeling because such instances directly penalize precision.

We worked over a couple of illustrative examples showing the difficulties FURIA inference mechanism presents and the opportunities Fingrams offer to illuminate its inference mechanism. Fingrams have demonstrated their capability to unveil FURIA particularities, such as details related to the stretching mechanism.

In the near future, FURIA-Fingrams will be included in the next release of the free software Fingrams Generator. Later on, we will extend Fingrams to assist the comprehension of other complex precise fuzzy systems.

Acknowledgment

This work was partially supported by the Spanish Ministry of Economy and Competitiveness under Grant TIN2014-56633-C3-3-R.

References


