A Linear-Complexity Rule Base Generation Method for Fuzzy Systems

Liviu-Cristian Duțu, Gilles Mauris, Philippe Bolon

Univ. Savoie Mont Blanc, LISTIC, F-74000 Annecy, France
{liviu-cristian.dutu, gilles.mauris, philippe.bolon}@univ-savoie.fr

Abstract

Rule base generation from numerical data has been a dynamic research topic within the fuzzy community in the last decades, and several well-established methods have been proposed. While some authors presented simple, empirical approaches, but which generally show high error rates, others turned to complex heuristic techniques to improve accuracy. In this paper, an extension of the classical Wang-Mendel method is proposed. While keeping a linear complexity, the new method achieves performances close to those of more complex methods based on cooperative rules (COR). Results on synthetic data show the potential of the proposed method as a complexity-accuracy trade-off.

Keywords: inductive rule learning, fuzzy rule-based model, double-consequent linguistic rules, complexity-accuracy trade-off, predictability

1. Introduction

Fuzzy rule-based systems, abbreviated FRBS, were successfully used in many real-world problems, with applications ranging from system modeling and diagnosis [1, 2] to dynamic control processes [3]. The vast popularity of FRBS resides mainly in their transparent and interpretable nature, which differentiates them from common “black-box” techniques such as multi-layer perceptron or support-vector machines. Besides, FRBS can be designed to offer very accurate models since it has been shown that they can approximate any real continuous function to arbitrary precision [4]. This comes however with an important loss in interpretability, and an application-specific compromise must be reached, yielding two main paradigms in fuzzy system modeling: accuracy-oriented and interpretability-oriented [5].

In both cases, FRBS are characterized by a bi-directional flow of knowledge, which can be either injected to the system by a human expert and then validated on experimental data (expert-driven FRBS), or automatically extracted from data following an optimization procedure and later verified by the system designer (data-driven FRBS). Although various combinations of the above instances can be found in the literature, e.g. [6], in this paper we will mainly focus on the latter, where FRBS design is considered in the absence of human experts.

Therefore, according to [7] the identification of the knowledge base of a FRBS is divided into the generation of its data base (DB) on the one hand, dealing with the fuzzy partitions of the input and output variables and their associated membership functions, and the generation of the system rule base (RB) on the other hand.

Although the above formalism can be adapted to any kind of FRBS, in this paper we will restrict ourselves to the case of Mamdani-type systems. More precisely, considering that the DB was a priori learned or imposed, we propose an original method to generate the RB for a Mamdani system from numerical input-output data pairs. The proposed method is based on the classical Wang-Mendel (WM) approach [8] combined with the double-consequent cooperative rules (COR) strategy [9]. The method is shown to have linear complexity in terms of number of evaluations needed, therefore providing a suitable framework for those applications where accuracy and speed are equally important. Moreover, with the use of linguistic variables [10], and by restricting the number of fuzzy rules, system interpretability is also preserved.

Considering the above, the rest of the paper is structured as follows. Section 2 provides an overview of best known RB generation methods, with a special focus on the WM-based approaches and the COR methodology. Then, in Section 3 the proposed method is detailed and its theoretical properties are studied. Section 4 is dedicated to the experimental results, where the considered method is tested on two function modeling tasks and its performances are compared with those of three reference methods in the literature. Finally, in Section 5 some concluding remarks are presented.

2. Overview of Rule Base Learning Methods

The RB is the core of every FRBS. Formally speaking, it represents a collection of fuzzy “IF-THEN” rules which connect the input variables of the system to its output in a way that is easy to read and interpret by humans. In the early years of fuzzy logic the RB was almost exclusively built using expert knowledge, and the rules obtained synthesized human expertise in the field. However, the lack of
Given a partition of the input and output variables of the system and a dataset of input-output numerical data [8], and which can be directly applied to obtain complete linguistic rules. Their method, successfully used in many real-world problems [13, 14, 15], is briefly described below.

### 2.1. WM Rule Base Generation Method

Given a partition of the input and output variables of the system and a dataset of input-output numerical examples, the steps to generate the RB through the WM method are:

1. For each example \( E = (x_1, x_2, ..., x_m, y) \), i.e., \( m \) input variables and one output variable, determine its membership degrees in every fuzzy partition of the input and output spaces;

2. Compute the fuzzy partitions, i.e., labels, with maximum degree (best covering label), in each of the input and output variables spaces; thus, we can associate to \( E \) a rule of the form:

   \[
   \text{IF } x_1 \text{ is } A_{x_1}^{l_1} \text{ and } x_m \text{ is } A_{x_m}^{l_m} \text{ THEN } y \text{ is } B_{y}^{k_y}
   \]

   where \( A_{x_i}^{l_i}, \forall i, \) and \( B_{y}^{k_y} \) are the linguistic labels best covering the example \( E \) in each input and output subspaces, respectively;

3. Iterate step 2) for every example \( E \) in the dataset, to obtain a candidate RB. Then assign an importance degree to each rule \( R \):

   \[
   D(R) = \mu_{A_{x_1}^{l_1}}(x_1) \cdot \cdots \cdot \mu_{A_{x_m}^{l_m}}(x_m) \cdot \mu_{B_{y}^{k_y}}(y)
   \]

   where \( A_{x_i}^{l_i}, \forall i, \) and \( B_{y}^{k_y} \) are the membership degrees of the linguistic labels best covering the example \( E \) in each input and output subspaces;

Steps 1) – 3) are illustrated in Figure 1 for a numerical example \( E \) with two input variables \((x_1, x_2)\) and one output variable \((y)\).

4. The preceding stages are likely to generate conflicting rules, i.e., rules with the same antecedents but different consequents. To settle these disputes, we need to group all rules having the same antecedents, and from each group, choose the rule with the maximum degree \( D(R) \), to form the final RB.

The WM method requires a single pass through the dataset of examples to generate the RB. Besides its proven efficiency, the method is very fast and easy to implement. In the next sections we will present two of its most common extensions proposed in the literature.

### 2.2. The WM Method Completed

A first extension was proposed by Wang in [16], where the rule learning process is carried out through the following steps:

1. Given a numerical input-output example of the form \( E = (x_1, x_2, ..., x_m, y) \in \mathbb{R}^{m+1} \), determine its membership degrees in every fuzzy partition of the input space only; then associate to \( E \) the best covered input subspace:

   \[
   E \Rightarrow (A_{x_1}^{l_1}, A_{x_2}^{l_2}, ..., A_{x_m}^{l_m})
   \]

   where \( A_{x_i}^{l_i}, \forall i \) are the input variables linguistic labels best covered by the example \( E \); please note that no output label is associated to \( E \) at this stage;

2. Compute the cumulative membership degree of \( E \) to its input subspace:

   \[
   \omega(E) = \prod_{i=1}^{m} \mu_{A_{x_i}^{l_i}}(x_i)
   \]

   It should be noticed that \( \omega(E) \) does not include the membership degree for the output variable.

   The above steps are iterated for all available examples, which are then clustered into the different input subspaces;

3. Next, let \( SE = (A_{x_1}^{l_1}, A_{x_2}^{l_2}, ..., A_{x_m}^{l_m}) \) be a given input subspace. Let \( K \) denote the number of examples situated in this subspace, i.e. \( E_k, k = 1, ..., K \). Each example has a corresponding membership degree \( \omega(E_k) \). Afterward, a weighted mean of these membership degrees is performed as follows [16]:

   \[
   a_{y}^{SE}(SE) = \frac{\sum_{k=1}^{K} \omega(E_k) y(E_k)}{\sum_{k=1}^{K} \omega(E_k)}
   \]

   where \( y(E_k) \) is the numerical output associated to the example \( E_k \).

   The output label assigned to the input subspace \( SE \), thus forming a complete linguistic rule, is the label \( B_{y}^{l_y} \) which best covers the numerical value \( a_{y}^{SE}(SE) \).
\[ \mu_{R^*_q}(av^{(SE)}) \geq \mu_{R^*_g}(av^{(SE)}) , \quad \forall q \]

Step 3) is iterated for all input subspaces containing at least one data point, thus obtaining the system RB. Though in [16] the author proposes a strategy to adjust the membership functions of the variables based on \( av^{(SE)} \) and a variance measure, we will limit ourselves to the case where the membership functions are \textit{a priori} imposed.

2.3. The WM Extension Based on the Cooperative Rules (COR) Strategy

A second class of extensions of the classical WM approach focuses on step 4) of the algorithm, in order to allow two consequents for the same pair of input antecedents, i.e. two output labels for the same input subspace.

Even though the \textit{double-consequent} rules emerged from Ishibuchi’s early works on a TSK system [17], the idea was refined by Cordón in [9] and adapted for linguistic modeling through a Mamdani FRBS. While the double-consequent rules approach can be used to extend several classical RB generation methods (see [9, 18]), in this section only the WM extension will be presented.

Therefore, under this strategy, step 4) of the original WM algorithm is altered, and from each group of rules having the same antecedents, the two rules with the highest importance degree \( D(R) \) are chosen. This process is illustrated on a numerical example in Figure 2. Let us point out that in those groups containing a single rule, that rule is the only one chosen. Spanning all available groups and applying the above mechanism allows one to construct a \textit{temporary rule base} containing \( NR \) rules, with:

\[ N \leq NR \leq 2N \quad (1) \]

where \( N \) is the number of rules obtained with the original WM method.

Next, a combinatorial optimization algorithm is used to search for the subset of rules in the temporary RB which minimizes some error criterion. In [9] the authors used a genetic algorithm to find the optimal set of rules using a population of chromosomes having \( NR \) binary genes, \( g_i \in \{0,1\}, \forall i = 1,\ldots, NR \). Therefore, if \( g_i = 1 \), then its corresponding rule from the temporary RB is included in a final rulebase, which is later used to evaluate the fitness function of the system.

It should be noticed that genetic rule learning dates back to 1991, when Thrift proposed in [19] a mechanism to encode the RB’s consequent array as an integer-valued chromosome.

Throughout the years, genetic and evolutionary techniques were extensively used in RB learning (see [23] for a survey), with successful applications in many fields such as medicine [24] or HVAC systems [6].

Since the introduction of the double-consequent methodology, several extensions have been proposed in the literature where genetic algorithms are replaced by other optimization techniques such as: simulated annealing [18], ant colony optimization [20, 21] or local search [22]. Although some differences exist between these approaches, they are all instances of a general methodology called \textit{cooperative rules strategy (COR)}, stating that good co-operation among rules, rather than good individual rules, is important for a FRBS.

However, the major drawback of these methods is associated with the high number of evaluations performed to find the optimal set of rules. In this regard, we outline that the search space tackled grows exponentially with the number of fuzzy labels of the input and output variables. This aspect limits their applicability for real-time or quasi real-time controllers which need to quickly update their structure as new data is delivered.

This is why in the following section we will introduce an original RB generation method designed to considerably reduce the number of evaluations performed, while still providing high accuracy FRBS. The pillars of our proposal are the WM method, presented in Section 2.1, and the double-consequent rules strategy, illustrated in Figure 2.

3. The Selection-Reduction Rule Base Generation Method

While greatly improving system performances, the COR methods perform a high number of evaluations to find the optimal set of rules. On the other hand, while performing a single pass through the dataset, the WM method generally provides higher error rates, and is sensitive to noise in the data.

Therefore, we believe that an appropriate trade-off between complexity and accuracy should be reached in order to provide a predictable framework in the context of RB generation.

The proposed method divides RB learning into two independent stages: the selection of the most relevant fuzzy rules, starting from a temporary RB of double-consequent rules; and then the reduction of the new RB obtained.

3.1. Rule Selection Stage

The steps performed in the rule-selection stage are the following:

1) Consider an original RB (ORB) obtained with the WM algorithm presented in Section 2.1. Let \( N \) denote its number of rules. Further consider a double-consequent RB (DCBR) obtained from ORB as illustrated in Figure 2. Let \( NR \) denote the number of rules of DCRB, where \( NR \) follows the set of inequalities in (1).

2) Let FRB denote the final RB of the system. Initialize it: \( FRB = ORB \).
3) Next, the relative complement (RC) of ORB in DCRB is computed as the set-theoretic difference between the two rule bases:

\[ RC = \{ DCRB \} \setminus \{ ORB \} \]

Therefore, RC is a set containing only the rules with the second-best importance degree \( D(R) \). The rules with the best importance degree are all in ORB. In this way DCRB is separated into two sets: ORB and RC. Note that the cardinals of these sets are not necessarily equal.

4) All second best-degree rules contained in RC are inserted one by one in FRB in lieu of their corresponding best degree rules (rules having the same pair of antecedents, but a different consequent).

5) On a predefined dataset of numerical examples, the system is tested using FRB as rule-base.

- If the global error of the system is reduced, then the original rule (having the best importance degree \( D(R) \)) is discarded, and the second-best degree rule is retained as final rule for its combination of antecedents; the global error of the system is updated;
- Otherwise, the original rule (having the best importance degree \( D(R) \)) is kept as final rule in FRB.

Iterating steps 4) and 5) for all rules in RC allows us to obtain an optimized rule-base (FRB). The selection stage is then followed by a simple rule reduction stage.

3.2. Rule Reduction Stage

This stage aims to improve both accuracy and interpretability of FRBS by removing those “malignant” rules in FRB which negatively affect system performances. Hereinafter, a rule \( R_i \in \text{FRB} \) is said to be malignant if the following holds:

\[ \text{Err}(\{ \text{FRB} \}) > \text{Err}(\{ \text{FRB} \} \setminus R_i) \]

where \( \text{Err}(\{ \text{FRB} \}) \) is the global error of the system using the complete rule-base \( \text{FRB} \) and \( \text{Err}(\{ \text{FRB} \} \setminus R_i) \) is the global error of the system using a rule-base obtained by removing \( R_i \).

The steps performed in the rule-reduction stage are the following:

1) Consider an auxiliary rule-base (ARB) and initialize it: \( \text{ARB} = \text{FRB} \).
2) One by one, each rule \( R_i \in \text{ARB} \) is removed:

\[ \text{ARB} = \{ \text{ARB} \} \setminus R_i \]

3) The system is tested with ARB on the predefined dataset of examples. If the global error is reduced below an a priori defined threshold, then \( R_i \) is definitively discarded from FRB:

\[ \text{FRB} = \{ \text{FRB} \} \setminus R_i \]

Following these two stages, the RB obtained is potentially better and more compact than the original rule-base ORB. Therefore, the metaheuristic search of the COR methods is replaced by two independent stages.

3.3. Theoretical Properties of the Selection-Reduction Method

Before presenting the behavior of our proposal on a function modeling task, let us first outline some of its most important theoretical properties, such as computational complexity, expected accuracy and compactness.

3.3.1. Computational complexity

In order to determine the complexity of the proposed method, we will use as indicator the total number of evaluations (TE) needed to complete the RB extraction process.

Considering the above algorithm, in the case of the Selection-Reduction (S-R) method, \( TE \) is the sum of the evaluations carried out in the two stages:

\[ TE = |\{ RC \}| + |\{ \text{FRB} \}| = |\{ DCRB \}| \]

\[ \Rightarrow TE = NR \] (2)
where \(|\{\cdot\}|\) represents the set cardinal. Therefore, \(TE\) is limited by:

\[ N \leq TE = NR \leq 2N \quad (3) \]

Equation (3) shows that the total number of evaluations performed scales linearly with \(N\), the original number of rules obtained with the WM method. Therefore, the computational complexity of the S-R method is \(O(N)\).

Moreover, once the double-consequent rule-base (DCRB) is obtained in step 1) of the Selection stage, the exact number of evaluations needed is already known. Thus, depending on the context, we can either choose to carry out the whole S-R procedure, or to keep using the original rule-base (ORB).

Let us better illustrate this behavior in the case of a system working in real-time, receiving new data from its sensors every \(T\) seconds. Knowing the time needed for a single RB evaluation \((t_e)\), the system can accurately estimate the time needed for a complete RB extraction on the new data using the S-R method. Three possibilities arise:

(a) the system will complete the whole S-R procedure if:

\[ t_e \times NR < T; \]

(b) the system will only complete the Selection stage if:

\[ \begin{cases} t_e \times |\{RC\}| < T \\ t_e \times NR \geq T \end{cases} \]

(c) the system will not start the procedure if it cannot complete at least the Selection stage, i.e. if \(t_e \times |\{RC\}| > T\); in this case, the original rule-base (ORB) will be used, while waiting for a human operator to change the periodicity \(T\) of the data.

Therefore, the linear-complexity of the S-R method, coupled with its predictable nature, provides an appropriate framework for those adaptive FRBS employed in a real-time context.

3.3.2. Properties of the S-R Method

Besides its linear and predictable complexity, the proposed method exhibits some more interesting properties, which are briefly discussed below:

- the S-R method yields at least the same quantitative performances on the training set as the original WM method, while producing at most the same number of rules;
- the maximum number of evaluations needed is \(2N\), and it is a priori known; besides, the S-R method can be adjusted to handle \(c\)-consequent rules \((c > 2)\) without loosing its \(O(N)\) complexity (we recall that \(O(c \cdot N) \equiv O(N)\));

- moreover, if the context allows it, i.e. if there are no time constraints, the final RB provided by the S-R method could be used as starting point in a heuristic search for a better RB using a combinatorial optimization technique such as genetic algorithms, simulated annealing or ant colony optimization;

Therefore, the proposed method offers an improvement in terms of accuracy compared to the WM method, and an improvement in terms of complexity and predictability compared to the COR methods. It can be seen as a compromise between accuracy and complexity, with the advantage of predictability.

4. Experimental Results

In this section the S-R method is tested on some synthetic data, and its behavior is compared with the three methods presented in this paper: the original Wang-Mendel method [8] (WM), the completed version of the WM method [16] (WM-C) and the COR genetic algorithm WM method [9] (COR-GA-WM).

In all cases, the input and output variables are defined by uniformly partitioned triangular membership functions, as shown in Figure 3. For each situation, the number of linguistic labels is specified. The inference operators used are the same for all methods: the arithmetic product \(T\)-norm as conjunction operator (as used in [8]), the Mamdani \(min\) \(T\)-norm as implication and \(max\) \(T\)-conorm as aggregation operators, along with the centroid method of defuzzification.

The four methods are tested with two performance indicators, namely:

- the mean-square error of the system:

\[ MSE = \frac{1}{M} \sum_{i=1}^{M} (P(i) - A(i))^2 \]

where \(M\) is the number of data samples evaluated, \(P(i)\) is the output value predicted by the system and \(A(i)\) is the actual output value of the data sample.

For the two iterative methods (S-R and COR-GA-WM) \(MSE\) is used as fitness function, to evaluate the performance of the system.

- the normalized global error of the system:

\[ \Delta = \frac{\sum_{i=1}^{M} |P(i) - A(i)|}{\sum_{i=1}^{M} MaxErr_i} \]

with

\[ MaxErr_i = \begin{cases} A(i) - min(I), & \text{if } A(i) \geq k \\ max(I) - A(i), & \text{otherwise} \end{cases} \]

where \(M, P\) and \(A\) as defined above; \(MaxErr_i\) is the maximal error that the system can make in predicting the output value of data sample \(i\),
and $k$ is the midpoint of $I$, which is the interval of variation of the output variable.

Besides the two performance indicators above, the final number of rules ($NR$) and the total number of evaluations performed ($TE$) are also used to compare the RB generation methods.

The parameters considered for the COR-GA-WM method are: a population of 100 binary-coded individuals, a maximum number of generations of 100 (with a stall criterion of 25 generations), crossover and mutation probabilities of 0.8 and 0.01 (per gene), respectively.

Since FRBS are universal function approximators [4], the performances of the above methods will be tested on two 3-D function modeling tasks, with different initial conditions.

### 4.1. Matyas Function Modeling

The first function to be modeled is the Matyas function, represented in Figure 4, and defined by the equation below:

$$F_{Matyas}(x_1, x_2) = 0.26 \cdot (x_1^2 + x_2^2) - 0.48 \cdot x_1 \cdot x_2$$

$x_1, x_2 \in [-1, 1]$; $F_{Matyas} \in [0, 1]$;

The function is frequently used to test the efficiency of optimization algorithms [25], and shows a global optimum at $(0, 0)$, where $F_{Matyas}(0, 0) = 0$.

The training dataset was composed of 6400 points, uniformly sampled in the 3-D space of the function. The training dataset was used by all methods to learn the RB.

The test dataset was composed of 2116 samples, i.e. $\sim 25\%$ of the (training + test) data samples, generated by randomly selecting points in the input space and determining the function value at those points. The same procedure was used for function modeling in [18].

Two initial conditions were considered. In the first condition, three uniformly partitioned fuzzy labels were used for each linguistic variable, and in the second condition, seven.

The results of all methods in the above conditions are collected in Table 1 and Table 2. In both cases, the S-R and COR-GA-WM methods outperform the non-iterative ones (WM and WM-C), and when three fuzzy labels are used for each linguistic variable they obtain the same quantitative results. The S-R method however, converges quickly and performs only $TE = 15$ evaluations, while the COR-GA-WM needs $TE = 2700$ evaluations.

When seven fuzzy labels are used for each linguistic variable, thereby causing an exponential growth of the search space, the COR-GA-WM needs no less than $TE = 7000$ evaluations in order to converge, while the S-R method performs only $TE = 87$ evaluations. Despite the huge difference in complexity, the two methods achieve approximately the same results (in terms of MSE and $\Delta$) on both training and test datasets. Moreover, thanks to its reduction stage, the S-R method obtains a slightly more compact RB with $NR = 45$. 

![Figure 3: Uniform partition of the universe of discourse in five fuzzy subsets using triangular-shaped membership functions.](image)

![Figure 4: Illustration of the Matyas function.](image)

<table>
<thead>
<tr>
<th></th>
<th>WM</th>
<th>WM-C</th>
<th>S-R</th>
<th>COR-GA-WM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE$_{tra}$</td>
<td>0.0999</td>
<td>0.0221</td>
<td><strong>0.0198</strong></td>
<td><strong>0.0198</strong></td>
</tr>
<tr>
<td>MSE$_{tst}$</td>
<td>0.0890</td>
<td>0.0197</td>
<td><strong>0.0146</strong></td>
<td><strong>0.0146</strong></td>
</tr>
<tr>
<td>$\Delta_{tra}$</td>
<td>34.59%</td>
<td>15.49%</td>
<td><strong>14.82%</strong></td>
<td><strong>14.82%</strong></td>
</tr>
<tr>
<td>$\Delta_{tst}$</td>
<td>32.53%</td>
<td>14.39%</td>
<td><strong>12.46%</strong></td>
<td><strong>12.46%</strong></td>
</tr>
<tr>
<td>$NR$</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>$TE$</td>
<td>–</td>
<td>–</td>
<td>15</td>
<td>2700</td>
</tr>
</tbody>
</table>

Table 1: Results of the Matyas function modeling with three labels for each linguistic variable. The best result for each indicator is shown in boldface.

<table>
<thead>
<tr>
<th></th>
<th>WM</th>
<th>WM-C</th>
<th>S-R</th>
<th>COR-GA-WM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE$_{tra}$</td>
<td>0.00504</td>
<td>0.00421</td>
<td><strong>0.00108</strong></td>
<td><strong>0.00106</strong></td>
</tr>
<tr>
<td>MSE$_{tst}$</td>
<td>0.00468</td>
<td>0.00397</td>
<td><strong>0.00112</strong></td>
<td><strong>0.00112</strong></td>
</tr>
<tr>
<td>$\Delta_{tra}$</td>
<td>7.82%</td>
<td>6.86%</td>
<td><strong>3.20%</strong></td>
<td><strong>3.17%</strong></td>
</tr>
<tr>
<td>$\Delta_{tst}$</td>
<td>7.67%</td>
<td>6.63%</td>
<td><strong>3.24%</strong></td>
<td><strong>3.20%</strong></td>
</tr>
<tr>
<td>$NR$</td>
<td>49</td>
<td>49</td>
<td><strong>45</strong></td>
<td><strong>47</strong></td>
</tr>
<tr>
<td>$TE$</td>
<td>–</td>
<td>–</td>
<td>87</td>
<td>7000</td>
</tr>
</tbody>
</table>

Table 2: Results of the Matyas function modeling with seven labels for each linguistic variable. The best result for each indicator is shown in boldface.
4.2. Sphere Function Modeling

The second function tested is the Sphere function, illustrated in Figure 5, and defined as follows:

\[ F_{\text{Sphere}}(x_1, x_2) = x_1^2 + x_2^2, \quad x_1, x_2 \in [-5, 5]; \quad F_{\text{Sphere}} \in [0, 50]; \]

The training and test datasets are generated as described in Section 4.1, with 6400 data points for the training set and 2116 data points for the test set. The same initial conditions, with three and seven fuzzy labels for each linguistic variable, are considered.

The results obtained can be consulted in Table 3 and Table 4 for the two initial conditions, respectively. Hence, in the three fuzzy labels condition, the S-R and COR-GA-WM methods yielded identical results (meaning that the same RB was mined by the two methods), which show significant improvement over WM or WM-C methods. There is however an important gap in complexity between the two methods, as the COR-GA-WM method needs to perform 200 times more evaluations to reach the same performances as the S-R method.

In the second experimental condition, where seven fuzzy labels are used for each linguistic variable, the COR-GA-WM method was more accurate than the S-R method on the training set, but the two show similar performances on the test set. The higher accuracy of the COR-GA-WM method comes from the considerable number of evaluations performed (\(TE = 9300\)), and also from the large number of fuzzy rules selected (\(NR = 65\)). On the other hand, the S-R method obtained comparable performances using a much more compact RB (\(NR = 49\)), and performing only \(TE = 97\) evaluations.

5. Conclusion

In this paper we proposed an original method to generate the rule-base of fuzzy systems from numerical data. The method is based on the classical Wang-Mendel approach corroborated by the double-consequent rules methodology. Employing two independent stages (rule-selection and rule-reduction), the proposed algorithm achieves linear complexity and predictability in terms of number of evaluations performed. Therefore, the Selection-Reduction method induces a tractable behavior, allowing FRBS to make contextual choices about whether or not a rule-base update is feasible.

Engaging a linear exploration of the space of double-consequent rules, the proposed method produces very reliable models, with performances comparable to those of the complex GA-driven approaches. Moreover, we showed that under certain conditions the Selection-Reduction method even manages to match the performances of the heuristic-based COR methods.

Although preliminary experiments on function modeling show promising results, further investigations are needed to assess the effectiveness of the proposed method in different contexts and for different applications. In this regard, an interesting perspective would be to use the solution of the Selection-Reduction method as starting point for the metaheuristic optimization algorithms, in order to speed up their convergence rate, and therefore reduce the number of evaluations performed.

References


