A Fast and Accurate Rule-Base Generation Method for Mamdani Fuzzy Systems

Liviu-Cristian Dutu, Gilles Mauris, Philippe Bolon

A Fast and Accurate Rule-Base Generation Method for Mamdani Fuzzy Systems

Liviu-Cristian Duţu, Gilles Mauris, *Member, IEEE,* and Philippe Bolon, *Member, IEEE*

Abstract

The problem of learning fuzzy rule-bases is analyzed from the perspective of finding a favorable balance between the *accuracy* of the system, the *speed* required to learn the rules, and finally, the *interpretability* of the rule-bases obtained. Therefore, we introduce a complete design procedure to learn and then optimize the system rule-base, called the Precise and Fast Fuzzy Modeling approach. Under this paradigm, fuzzy rules are generated from numerical data using a parameterizable greedy-based learning method called *Selection-Reduction*, whose accuracy-speed efficiency is confirmed through empirical results and comparisons with reference methods. Qualitative justification for this method is provided based on the co-action between fuzzy logic and the intrinsic properties of greedy algorithms. To complete the Precise and Fast Fuzzy Modeling strategy, we finally present a rule-base optimization technique driven by a novel rule redundancy index which takes into account the concepts of distance between rules and the influence of a rule over the dataset. Experimental results show that the proposed index can be used to obtain compact rule-bases which remain very accurate, thus increasing system interpretability.

Index Terms

fuzzy modeling, inductive rule learning, double-consequent linguistic rules, accuracy-speed trade-off, greedy rule selection, interpretability–accuracy trade-off, rule-base reduction.

L.-C. Duţu, G. Mauris and Ph. Bolon are with the Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance (LISTIC), Polytech Annecy-Chambéry, Université Savoie Mont Blanc, B.P. 80439, 74944 Annecy le Vieux Cedex FRANCE (e-mail: liviu-cristian.dutu@univ-smb.fr; gilles.mauris@univ-smb.fr; philippe.bolon@univ-smb.fr). L.-C. Duţu is also an R&D Engineer at FotoNation in Bucharest, ROMANIA.

I. INTRODUCTION

The origins of system modeling and control through the use of fuzzy set theory as a mathematical background can be traced back to 1975, when Mamdani and Assilian's seminal work [1] introduced the first rule-based controller powered by a fuzzy inference mechanism. Such a system is commonly called Fuzzy Rule Based System or FRBS for short. After more than four decades of practical validation in a wide variety of domains, ranging from industrial process control [2] to human activity modeling [3] or medical diagnosis [4], FRBSs are now largely considered to be the main application of fuzzy logic.

FRBSs gained the attention of the community through their capacity to handle the imprecisions of real-world data and capture the essence of a given phenomenon in a way that is understandable and interpretable by humans. Moreover, as shown in [5], [6], FRBSs act as universal function approximators, meaning that they can be adjusted to approximate any continuous function up to arbitrary precision. Therefore FRBSs offer the possibility of trading accuracy for interpretability and *vice-versa*, in a very transparent and flexible way, until the desired trade-off between the two is met. In this regard, two directions can be differentiated [7]: Precise Fuzzy Modeling, where the system is first designed so as to maximize accuracy and then adjusted to improve interpretability, and Linguistic Fuzzy Modeling, where the main focus is to achieve good interpretability, and only then to improve the accuracy.

Nevertheless, in both cases, a fuzzy system needs to be fueled by *knowledge* in order to work properly. Here we distinguish two more trends in FRBS design: the *expert-driven* approach, where knowledge is directly injected into the system by a human expert, and the *data-driven* approach, where knowledge is autonomously discovered and extracted by the system from numerical data examples. Please note that the trends mentioned above are not mutually exclusive, and hybrid approaches were also explored (e.g. in [8] the authors use a data-driven procedure to fine-tune an expert-defined system). In practical applications, Precise Fuzzy Modeling (PFM) is often associated with the data-driven approach, resulting in very robust and accurate models generally used in dynamic controllers (see [9] for a survey), while Linguistic Fuzzy Modeling (LFM) is associated with the expert-driven approach, resulting in very interpretable models, generally used for system diagnosis [10].

While the accuracy provided by the PFM approach and the interpretability of the LFM strategy

are both important in practical applications, we argue that a third criterion should be considered when designing FRBSs: the *speed* at which knowledge can be extracted from data, i.e. the convergence speed of the system. This criterion is particularly important for embedded systems functioning in real-time. In this case, the strategy used to design FRBSs must be fast enough to allow them to adapt and self-evolve every time new and relevant data comes in. Moreover, when dealing with real-time self-evolving systems, the learning strategy should also be time-deterministic, i.e. being able to know beforehand the time (or number of evaluations) needed to learn the model, and also flexible enough to allow the user to select the proper balance between accuracy and speed.

We further consider that these desirable properties are not properly taken into account by the PFM or LFM approaches. Therefore, this paper proposes a particularly original perspective on data-driven Precise Fuzzy Models, where accuracy, interpretability and convergence speed are combined into a unified fuzzy modeling formalism governed by deterministic learning times. We propose to call this formalism the Precise and Fast Fuzzy Modeling approach (PFFM), as a sub-class of the PFM methodology which focuses mainly on system accuracy and convergence speed, and the balance between them. In order to provide highly interpretable rule-bases, in this paper we will consider only the case of Mamdani systems which intrinsically preserve the linguistic interpretability of fuzzy rules.

Within the Precise Fuzzy Modeling framework, one of the most successful approaches in data-driven fuzzy rule base generation is the Cooperative Rules (COR) methodology [11], [12] which yields optimized rule bases starting from an initial educated guess, using different combinatorial search algorithms. While exploiting the cooperation among rules can significantly improve the accuracy of a system, the convergence speed of such methods is generally slow and cannot be estimated beforehand, thus making *speed* a nondeterministic parameter. In order to tackle this issue, we will present a class of methods inspired by the COR strategy, which seek to optimize the overall balance between accuracy and speed. Therefore, we will show that by using a fast and time-deterministic learning approach, one can obtain high accuracy, which is the core of the PFFM methodology.

Moreover, in order to complete the PFFM formalism, the interpretability issues related to automatic rule base generation must also be addressed. In this regard we will introduce a novel technique to optimize the interpretability of an already defined rule base, using a redundancy

analysis of the rules based on the influence they exert on each other. We will show that this analysis can help to reduce the number of rules (thus yielding higher interpretability), without significantly affecting system accuracy.

Therefore, within the PFFM methodology, the main contributions of this paper are: (i) the introduction of a class of methods to generate the system rule-base from data, based on a greedy exploration of the space of rules; as we will show, the proposed methods achieve accuracy comparable to COR-like techniques, but unlike them, can be trained quickly; (ii) a theoretical analysis for the aforementioned methods, where we'll provide qualitative justification for their usefulness, showing why greedy-based techniques are well adapted for fuzzy rule-base generation; (iii) the introduction of a novel index to measure the redundancy of the rules within a rule-base; this redundancy index will be later used to efficiently prune the rule-base.

Given the above, this paper is structured as follows. Section II proposes a literature survey on some of the most successful data-driven rule-base generation methods, with a special emphasis on their accuracy and speed properties. Next, in Section III, following the PFFM requirements, we will introduce a family of methods designed to provide precise FRBSs, which are also very fast to learn. We will also show how the mathematical properties of our methods can be used in real-world situations and provide some guidelines on the optimization of other components of a FRBS. Once the system rule base is generated, in Section IV we will describe an efficient method to improve its interpretability by removing the highly redundant rules, while still preserving system accuracy. Then, in Section V the complete formalism to generate and optimize the system's rule base will be illustrated using a series of experimental data comprising both synthetic and real-world problems. Comparisons with similar methods from the literature is provided. Ultimately, in Section VI the final conclusions and remarks will be presented.

II. Overview of Data-Driven Rule-Base Learning Methods

Rule-base (RB) identification is arguably the most important aspect of FRBS design, as both the accuracy and interpretability of the system heavily rely on it. Although during the early stages of fuzzy system modeling the rule-base was mainly defined by panels of experts, the ever-growing demands of the industry and the higher complexity of the problems to be tackled asked for new and autonomous ways to generate the rules from numerical examples. We mention here the seminal work of Sugeno [13] which provided a fully-automated strategy to generate

fuzzy rules from numerical data. However, the crisp outputs produced by Sugeno's rules, while improving accuracy, considerably affected the overall interpretability of the system.

Another important milestone in data-driven RB learning was established by Wang and Mendel [14] who proposed a non-iterative method to generate complete linguistic rules from data. Over time, the efficiency of the Wang-Mendel (WM) approach was successfully proven in many real-world applications [15], [16], [17] and it is now considered an important benchmark for researchers. Since the proposed RB generation method detailed in Section III relies on some of the principles of the WM method, and because it is inspired by the COR methodology, we shall first present the key aspects of these methods.

A. WM Method for Rule-Base Generation

Like most RB learning methods, the WM approach performs rule induction by considering that the fuzzy partitions and the membership functions are already set. Therefore, given a certain partition of the input and output fuzzy sets and a set of input-output numerical examples of the form $E = (x_1, x_2, ..., x_m, y)$, with x_i being the *i*-th input variable and y being the output variable, the WM method consists of the following steps:

1) For each numerical example E:
   - Determine its membership degrees in every fuzzy partition of the input and output spaces;
   - Associate to E a rule with the linguistic labels best covered by the example, i.e. fuzzy subsets with maximum degree in each input and output subspace. Let R^E be the rule associated with example E, which takes the following form:

$$\text{IF } x_1 \text{ is } A_{x_1}^{l_{x_1}} \text{ ... and } x_m \text{ is } A_{x_m}^{l_{x_m}}$$
$$\text{THEN } y \text{ is } B_y^{l_y}$$

   where $A_{x_i}^{l_{x_i}}\ \forall i$, are the linguistic labels best covered by E in each input subspace and and $B_y^{l_y}$ is the best covered output label; the membership degree of E in each of these subspaces is $\mu_{A_{x_i}^{l_{x_i}}}\ \forall i$ and $\mu_{B_y^{l_y}}$, respectively;
2) After stage 1) we obtain a *candidate* RB, whose cardinality equals the number of examples at our disposal. For each rule in the *candidate* RB we assign an importance degree:

$$D(R) = \mu_{A_{x_1}^{l_{x_1}}}(x_1) \cdot ... \cdot \mu_{A_{x_m}^{l_{x_m}}}(x_m) \cdot \mu_{B_y^{l_y}}(y)$$

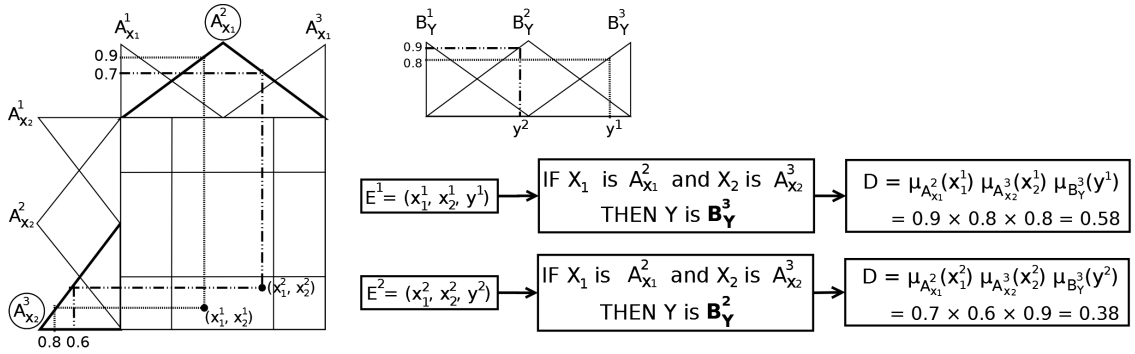

Fig. 1. The WM method for RB generation. A certain rule is generated for each data-point, and then its importance degree is calculated.

3) Since having one rule per numerical example can generate conflicting rules (rules having the same pair of antecedents but different consequents), 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. This stage is illustrated in Fig. 1, where numerical examples E^1 and E^2 generate rules with the same antecedents but different consequents. Based on the importance degree $D(R)$, the final rule for this pair of antecedents will be provided by E^1, since $D(R^{E^1}) = 0.58 > D(R^{E^2}) = 0.38$

The WM method provides a fast and non-iterative way to learn the linguistic rules from data. From a PFFM point of view, the WM method focuses mainly on speed, since a single pass through the training set is enough to generate the RB. However, no effort is made to optimize the accuracy of RB, making the method very sensitive to noise in the data (one noisy sample in each subspace with a high importance degree $D(R)$ is all that is needed to pick the wrong consequent).

After a few years, the original WM method was revised and completed by Wang in [18]. The new method, briefly presented in the following section, aims to reduce the influence of noisy data on the final RB.

B. WM Method Completed

Given a set of numerical examples $E_i = (x_1, x_2, ..., x_m, y) \in \Re^{m+1}$, the steps performed by the WM method completed (WM-C) are the following:

1) For each numerical example E_i, determine its membership degree in every input subspace, and then compute its cumulative membership degree as follows:
$$\omega^{(E_i)} = \prod_{i=1}^{m} \mu_{A_{x_i}^{l_{x_i}}}(x_i)$$
where $\mu_{A_{x_i}^{l_{x_i}}}(x_i)$ are the individual membership degrees in each input subspace.
Please note that $\omega^{(E_i)} = \frac{D(R^{E_i})}{\mu_{B_y^{l_y}}(y)}$, i.e. the output degree $\mu_{B_y^{l_y}}(y)$ is not taken into account.
2) Next, for each input subspace, the weighted mean of the output values (y^k) of the samples in the subspace is computed:
$$av^{(IS)} = \frac{\sum_{k=1}^{K} y^{E_k}\omega^{(E_k)}}{\sum_{k=1}^{K}\omega^{(E_k)}} \in \Re$$
where $E_k, k = 1, ..., K$ are numerical examples belonging to input subspace IS.
3) Finally, the output label best covered by $av^{(IS)}$ is chosen as the consequent for the input space IS. Stages 2) and 3) are repeated for every input subspace with at least one numerical example. It should be noticed that in [18] the author suggests using the weighted variance around $av^{(SE)}$ to adjust the membership functions of the output variable.

Using the weighted mean rather than a single sample can help the WM-C method to solve some of the problems of the original WM method. However, WM-C assumes that the distribution of the weighted output degrees $y^{E_k}\omega^{(E_k)}$ is tightly centered around $as^{(IS)}$, i.e. has a small variance. Under these conditions, noise-corrupted input values $x_i, i = 1, ..., m$ can affect the final value of $av^{(IS)}$, and also the value of the final picked linguistic label.

It is clear that non-iterative methods, although fast, suffer under imperfect data and fail to provide highly accurate FRBS. As speed and precision are equally important for the PFFM formalism, in the following section we will start to investigate iterative RB learning methods designed to maximize system accuracy based on cooperation among rules (COR methodology).

C. Iterative RB Learning - COR Approach

Under iterative approaches, RB generation can be seen as an optimization problem where we seek to find the global minimum of the error surface. While for very simple systems direct enumeration of all the possible RB configurations could be feasible, when the number of variables and fuzzy subsets grow, the number of all possible rule-bases become too large[1] for brute-

[1]For a system having 5 input variables and one output variable, each partitioned into 5 fuzzy subsets, the number of all possible RB is in the order of $5^{(5\times5)} \approx 3 \cdot 10^{17}$.

force techniques, and more subtle approaches are needed. As RB learning is generally not a convex optimization problem, heuristic methods offering near-optimal solutions have become very popular within the fuzzy community.

One of the most successful methods is the so-called "*COoperative Rules*" (COR) strategy, developed by Cordón and Herrera in [11] and refined by Casillas *et al.* in [12]. The COR methodology extends non-iterative methods such as WM, by creating a large pool of possible rule-bases, which is explored using search heuristics. Therefore, the COR approach can be divided into two steps which are detailed below: the generation of the pool of all possible RB, and the heuristic selection of the final RB.

1) RB Pool Generation: The reason we need smart ways to populate the RB pool is that, even for medium-scaled systems, allowing *every possible* RB leads to an incredibly large pool which cannot be handled in reasonable time. Generation of this pool under the COR strategy is done by allowing *two* or more consequents, i.e. output labels, for the same pair of antecedents, i.e. input subspace, resulting in a series of *double-consequent* rules. It should be noticed that the concept of double-consequent rules was first developed by Ishibuchi to improve the performance of a TSK system [19], and later adapted by Cordón and Herrera to handle linguistic consequents.

Generation of double-consequent rules and double-consequent rule-bases ($DCRB$) as an extension of the WM algorithm is illustrated in Fig. 2. Therefore, for a given input subspace, the rules generated by all data-points belonging to this subspace are sorted with respect to their importance degree $D(R)$, and the two rules with the highest importance degrees are added to the $DCRB$ (rules associated with data-points P_1 and P_3, in this example). Please note that the same procedure can be used to generate *c-consequent* rules, with $c > 2$. If some input subspaces have no data-points, no rules will be generated, and if other subspaces have only one data-point, then only one rule will be generated.

If we let N be the number of rules obtained by the original WM method, then the number of c-consequent rules obtained is NR, satisfying the inequality below:

$$N \leq NR \leq cN \tag{1}$$

where $c = 2$ in the case of double-consequent rules. It should be noticed that this extension principle is not restricted to WM, and can also be applied to other methods (see [11], [12] for details).

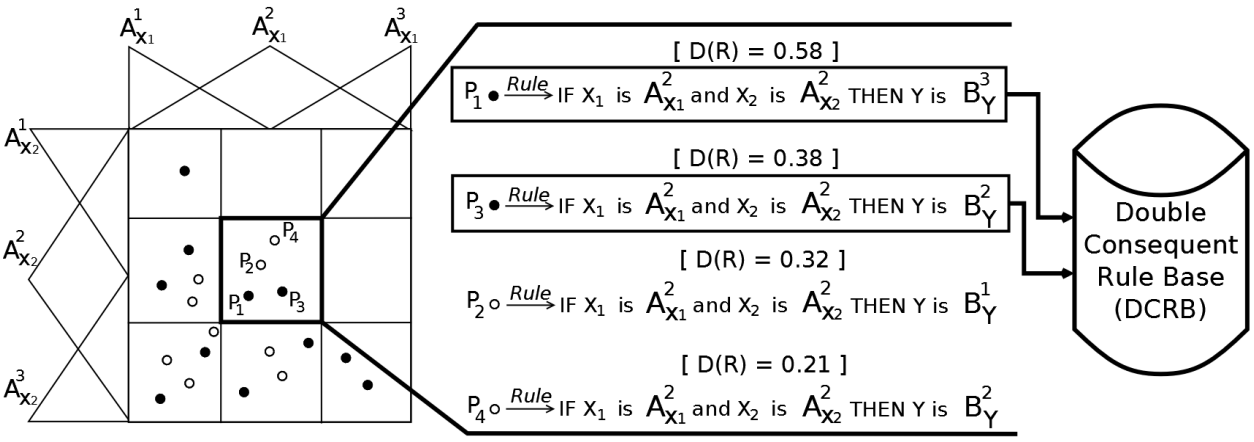

Fig. 2. Graphical illustration of the double-consequent principle. For each input subspace, the *two* rules with the highest importance degrees are picked.

At this point, the RB pool can be defined as the set of all possible rule-bases that can be obtained using the NR rules generated above. Therefore, the size of the RB pool is:

$$\sum_{k=1}^{NR} \binom{NR}{k} = 2^{NR} - 1. \tag{2}$$

2) Heuristic Selection of the RB: In order to explore the vast number of possible RB, in [11] the authors use genetic algorithms where each chromosome is composed of NR binary genes $g_i \in \{0, 1\}, i = 1, ..., NR$. With this coding scheme, the individual rule R_i is kept in the RB *iff* $g_i = 1$, or it is discarded otherwise.

Therefore, each chromosome represents a possible RB from the RB pool, which is evaluated on the set of training examples. Following a stochastic iterative procedure, individual chromosomes undertake gene mutations and are then combined with each other until a stable point is reached. Finally, the chromosome with the lowest fitness function, i.e. highest accuracy, gives the final RB of the system. Although genetic algorithms can be used to fine-tune the membership functions of the system variables as well (process commonly known as *lateral tuning* [20], [21]), in this paper we will confine ourselves to the case where only the RB is automatically generated.

While genetic and evolutionary techniques are most often used in the literature for RB selection (see [22] for a review), they are not the only choice available. Hence, several authors propose to replace them with other heuristics such as: simulated annealing [12], local search [23], ant colony

optimization [24]-[25] or modified versions of it [26], bee colony optimization [27], bacterial algorithms [28] or even cuckoo search methods [29].

Moreover, selecting the best RB of a fuzzy system can also be seen as a multi-objective problem, where a proper balance between accuracy and interpretability is sought. Therefore, several evolutionary techniques have been proposed to concurrently optimize the accuracy and the complexity (generally measured as the total number of rules or total number of conditions in a rule) of a FRBS. Among the multi-objective evolutionary algorithms used in the literature for this purpose, we acknowledge: genetic algorithms approaches [30], the Pareto archived evolution strategy [31], the non-dominated sorting genetic algorithms [32] or the continuous ant colony optimization algorithm proposed in [33]. Please see [34] for a review on this subject. Since from a PFFM point of view we are mainly concerned with accuracy and training speed, we advocate that RB complexity should be addressed separately, as we will show in Section V.

Given enough time, heuristic methods surpass non-iterative ones in terms of system accuracy. However they need to perform a large number of evaluations in order to converge towards the optimal RB of the system. This affects the speed at which new models can be learned, which is a critical condition for self-adaptive systems used in real-time environments. This is why in the following section we present a family of RB learning methods designed to optimize the accuracy of the system using the smallest possible number of evaluations. Our approach is based on a *greedy search* of the space of possible RBs, followed by a rule removal procedure.

III. The *Selection-Reduction* Approach for Fast and Accurate RB Learning

It is clear that when RB learning has to be both accurate and fast, a compromise between the speed of non-iterative methods and the accuracy of heuristic procedures must be reached. This compromise, which is at the core of the Precise and Fast Fuzzy Modeling approach, can only be obtained by efficiently exploring the space of candidate RBs. Moreover, when the fuzzy system has to adapt to new data and self-evolve, it is important to know beforehand the total number of evaluations needed to learn a new RB, i.e. we need a time-deterministic approach to RB learning.

Therefore, in [35] the *Selection-Reduction* RB generation method was introduced with the aim of providing an efficient trade-off between accuracy and complexity, with the additional advantage of predictability. The *Selection-Reduction* (SR) method uses an iterative procedure to

learn the system RB, but unlike the COR approaches, iterations are performed in a *greedy* way, yielding a low complexity and time-deterministic method.

The rest of this section is structured as follows: subsection III-A offers a brief reminder of the original *Selection-Reduction* method for RB learning, while subsection III-B provides a comprehensive theoretical analysis of the method highlighting its strengths and limitations. Based on this evidence, subsection III-C introduces a parametric version of the SR method, which takes into account the order in which the fuzzy subspaces are parsed.

A. The Selection-Reduction RB Learning Method

As the name suggests, the SR method is composed of two separate stages: the individual *selection* of the most relevant rule from each input subspace, followed by a rule *reduction* stage, which can be seen as a parametrized pruning stage. Therefore, the steps performed in the rule selection stage are listed below.

`Algorithm 1a - Selection Stage:`

S-1) Let ORB be the original RB as obtained with the WM method detailed in Section II-A, and let N be the cardinal of ORB, i.e. its number of rules. Next, consider a *multi-consequent* rule base $MCRB$ where each pair of antecedents can have up to c different consequents (c-consequent rules). $MCRB$ is obtained from ORB as presented in Section II-C1 and shown in Figure 2. Please note that for the sake of simplicity Figure 2 illustrates the particular case of double-consequent rule bases, i.e. where $c = 2$. However, extensions for $c > 2$ are straightforward to implement.

Moreover, if we let NR denote the number of rules of $MCRB$, we obtain that the relation between N, NR and c is governed by the set of inequalities in (1).

S-2) Define the final RB of the system (FRB) and initialize it to ORB:

$$FRB = ORB$$

S-3) Compute the relative complement of ORB in $MCRB$ using the set-theoretic difference between the two rule bases:

$$RC = \{MCRB\} \setminus \{ORB\}$$

While ORB contains the rules with the best importance degree $D(R)$, RC can be seen as the set of rules whose importance degree is non-dominant in their corresponding input

space, i.e. rules with the second-best, third-best,..., *c*th-best importance degree. The division of $MCRB$ into these two sets is also illustrated in Figure 3.

S-4) Afterwards, for the current input subspace I, do:

- One by one, insert each non-dominant rule from RC in FRB *in lieu* of its corresponding best degree rule. For example, in the input subspace illustrated in Figure 3, consider that the best degree rule ($D(R) = 0.58$) is removed from FRB and that, one at a time, the non-dominant rules from RC are inserted in place. Please note that for each input subspace there are at most $(c-1)$ replacements to be done, corresponding to the $(c-1)$ non-dominant rules in RC.
- After each replacement, the system is evaluated on a dataset of numerical examples using FRB as rule base. The original best degree rule is evaluated on the same dataset. Hence, for each input subspace there are up to c different rules to be compared, each with a different consequent. The final rule is the one that minimizes the global error of the system.
- FRB is updated to use the best rule as determined above for the current input subspace. All subsequent input subspaces will use this *updated* version of FRB.

S-5) Repeat step S-4) for each input subspace.

Once the above steps are completed, we obtain the optimized version of FRB. Since each evaluation is performed with replacement, i.e. one candidate rule is replaced by another candidate rule with a different consequent, the number of rules of FRB remains constant, and is equal to N. In the next stage however, we will perform a simple pruning procedure with the aim of removing the "malignant" rules from FRB whose presence reduces system accuracy.

`Algorithm 1b - Reduction Stage:`

R-1) Define an auxiliary rule-base ARB and initialize it: $ARB = FRB$.

R-2) For each rule $R_i \in ARB$ do:

- Remove R_i from the auxiliary rule-base:

$$ARB = \{ARB\} \setminus R_i;$$

- Test the system with ARB on a set of numerical examples. If the global error is reduced below a predefined threshold $S[\%]$, then definitively discard R_i from FRB:

$$FRB = \{FRB\} \setminus R_i;$$

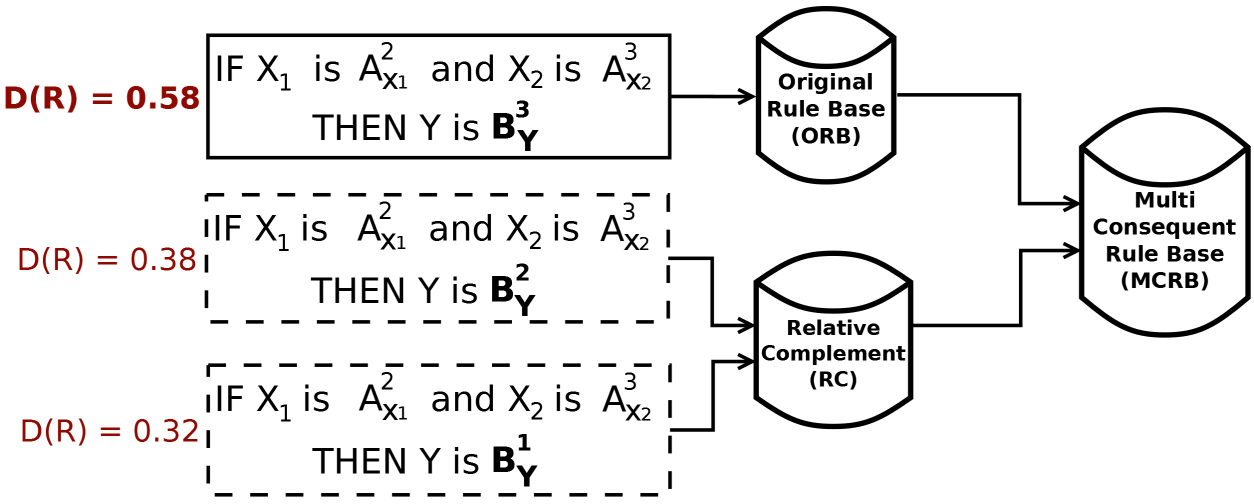

Fig. 3. Composition of the Multi-Consequent Rule Base ($MCRB$) with rules from the Original Rule Base (ORB) and rules from the Relative Complement (RC) for a given input subspace. ORB contains all the dominant rules, i.e. rules with the best importance degree $D(R)$, while RC contains only non-dominant rules.

While ensuring that each removed rule increases the accuracy of the system by at least $S[\%]$, we can also efficiently reduce the size of the RB, thus increasing the overall interpretability of the system.

B. Theoretical Considerations, Properties and Limitations of the SR *Method*

1) On the optimality of the proposed approach: Therefore, through its independent selection and reduction stages, the SR method performs a fast exploration of the space of possible rule-bases. Unlike the COR techniques, the exploration performed by the SR method is said to be *greedy*, because at each step of the algorithm, the decision about which rule to chose or which rule to remove, is taken solely based on the then-current state of the system, without considering any future states that the system may be in. Moreover, in order to guarantee maximum convergence speed, the decisions made by the algorithm are never reconsidered. We note that greedy-based approaches were traditionally used to learn decision trees (e.g., CART [36] or ID3 [37]) and have lately been investigated to learn deep neural networks [38].

Due to their non-reconsideration of previous decisions, greedy techniques generally fail to find the global optimum of a problem. However, when the problem in question exhibits "optimal

substructure"[2], greedy algorithms are guaranteed to find its global optimum. In the case of RB learning, determining the best consequent for a given pair of antecedents can be thought of as a subproblem. In this case, optimal substructure is achieved if for every input variable, the intersection of any two fuzzy subsets is empty:

$$A_i^I \cap A_j^I = \varnothing; \quad \forall I; \quad \forall i, j, i \neq j \tag{3}$$

where A_i^I and A_j^I are fuzzy subsets of the input variable I. It is worth noticing that this limitation applies to the input fuzzy subsets *only*. Hence, no constraint on the output fuzzy subsets of the system is required to obtain a problem with optimal substructure.

This can easily be proven by showing that if (3) is satisfied, then each subproblem is independent. Therefore, optimal solutions for subproblems, (i.e. picking the best consequent for a pair of antecedents), yield the optimal solution for the whole problem (i.e. the best global rule-base). Figure 4 illustrates a system with two input variables whose fuzzy subsets satisfy (3). We can easily see that each rule is independent and its optimal consequent depends solely on the datapoints that activate it. Please note that a direct implication of the above observation is that non-fuzzy (crisp) rule-base systems can be learned optimally by using the *Selection-Reduction* method.

Unequivocally, most fuzzy systems will disobey the overly-restrictive equation (3) and allow their fuzzy subsets a certain degree of overlapping. In this case, the problem loses its optimal substructure property, and greedy techniques are no longer guaranteed to find the optimal rule-base of the system. This happens because the rules are no longer independent, as one datapoint can simultaneously activate more rules. However, we advocate that if the degree of overlapping is carefully chosen, a certain degree of rule independence can be obtained.

For example, if we impose that *at most* two fuzzy subsets can be activated simultaneously, we can restrict the number of rules which have an influence on any given rule. In this case only the neighboring rules of a rule R_i can have an influence on it. Figure 5 illustrates this behavior, where only the neighboring rules R_2, R_4 and R_5 can influence rule R_1, while the other rules (R_3, R_7, R_6, R_8, R_9) have no influence on it. Please note that reducing the common

[2]The optimal substructure property of a problem implies that the optimal solution to that problem can be obtained by picking optimal solutions to all of its subproblems.

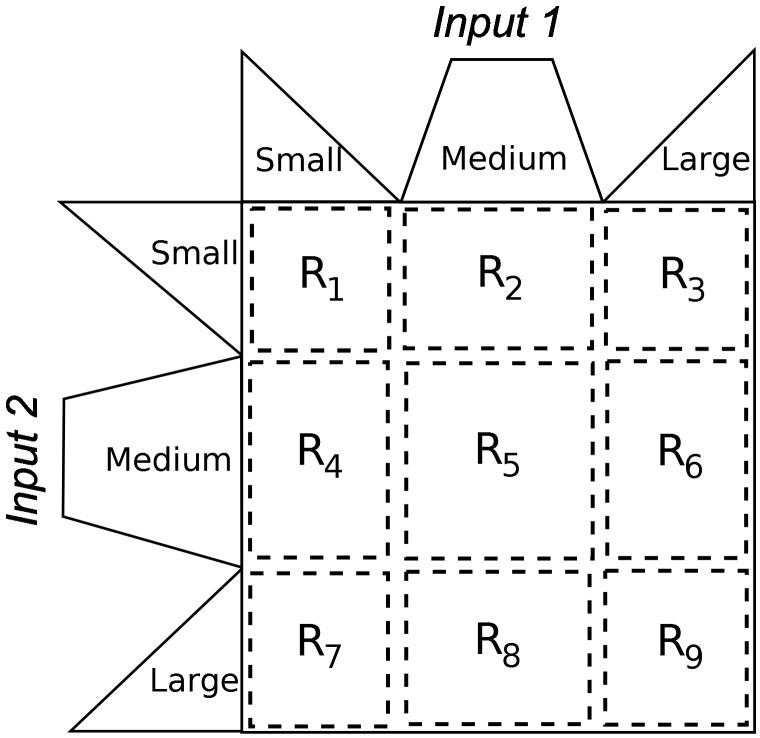

Fig. 4. System composed of two input variables, each with three fuzzy subsets. Since for both input variables the fuzzy subsets satisfy (3), every datapoint activates one and only one rule. Hence, determining the best rule for each input subspace is independent of the other rules of the system.

influence of the rules amounts to increasing their relative independence. Therefore, as we will show in Section V, the greedy-based Selection-Reduction method can successfully exploit the approximate independence property of a fuzzy system and provide near-optimal rule-bases while performing only a tiny fraction of the number of evaluations needed by the COR methods.

2) Computational complexity and other properties: As required by the PFFM paradigm, the speed at which new RBs can be learned is, along with its accuracy, a critical parameter of the system. In our case, the learning speed is a direct consequence of the complexity of the method used. This complexity can be assessed by the total number of evaluations needed to reach the final and optimized RB. In the following, *a system evaluation* will indicate the process of testing the system with a candidate RB over a predefined set of numerical examples.

Therefore, let TE indicate the total number of evaluations performed in order to learn a new RB. In the case of the SR method, one can easily show that TE is the sum of the evaluations

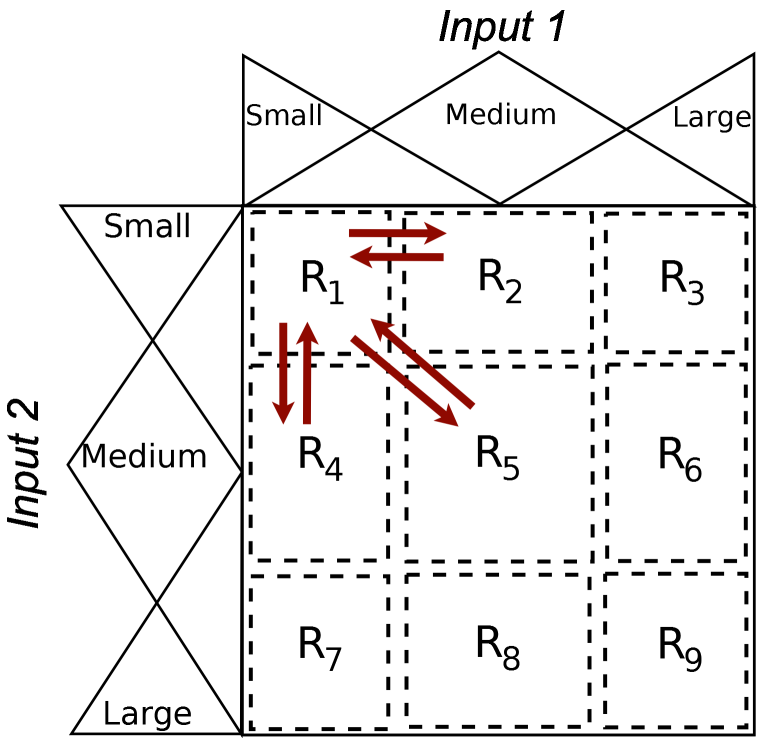

Fig. 5. Fuzzy system with two input variables and 9 fuzzy subsets empowered by Ruspini partitions. In these conditions, the only rules which can potentially have an influence on rule R_1 are its neighboring rules: R_2, R_4 and R_5.

performed in the two separate stages. More precisely:

$$\begin{aligned} TE &= |\{RC\}| + |\{FRB\}| = |\{MCRB\}| \\ &\Rightarrow TE = |\{MCRB\}| = NR \end{aligned} \tag{4}$$

where $|\{\cdot\}|$ stands for set cardinal, i.e. the number of rules; RC, FRB and $MCRB$ are the sets of rules as defined in Section III-A; and NR is the number of rules of $MCRB$;

Recalling the relationship between NR and N (the number of rules obtained when we have one rule for each input subspace populated by at least one datapoint), we obtain:

$$N \leq TE = NR \leq cN \tag{5}$$

The above inequalities show that the number of evaluations performed by the Selection-Reduction method scales linearly with N, i.e. its computational complexity is $O(N)$. This stands because in practice $c \ll N$, i.e. the number of candidate consequents to be tested in each input subspace is much smaller than the actual number of subspaces. Please note that TE is several orders of magnitude smaller than the theoretical size of the RB pool depicted in (2).

Moreover, a closer look at (4) reveals that the number of evaluations that need to be performed is known *a priori*, since it is the cardinal of the Multi-Consequent Rule-Base ($MCRB$). Therefore, once the $MCRB$ is built we can know the exact number of evaluations needed to learn the system RB. This important property of the SR method makes it the ideal choice for an autonomous adaptive system, since we can precisely estimate the time needed to learn a new RB and decide whether or not it is feasible to update the RB based on new data.

Therefore, we now have a fast and time-deterministic approach to efficiently learn the RB of fuzzy systems. Moreover, by exploiting the relatively low level of inter-dependence of fuzzy rules, our greedy exploration technique can achieve near-optimal solutions. Besides, our approach exhibits other practically appealing properties such as:

- an upper bound on the error of the system: the SR method will always have an error rate that is less than or equal to the one produced by the WM method (which was used as the original RB of the system).
- an upper bound on the number of rules in the final RB: once again, the SR method will produce RBs with fewer rules than the RBs produced by the WM method. This helps to improve the interpretability of the system.
- the complete separation of the *Selection* and *Reduction* stages, which allows them to be carried out at different times, e.g. the *Reduction* stage can be performed once every Q learning cycles, in order to speed up the RB learning process.

3) Limitations: The greedy nature of the SR method, while helping us to achieve a fast and time-deterministic way to learn RBs, might also be viewed as its biggest drawback. Although we have shown that greedy search approaches can take advantage of the low inter-dependence level of fuzzy rules and provide accurate solutions, these techniques are also known to be very sensitive to the search starting position.

In our case, the *starting position* for the SR method is the order in which the rules, or fuzzy subspaces, are being evaluated by the algorithm. Of course, since the fuzzy rules are not completely independent, picking a different way to span through the input subspaces in step S-4) of the algorithm may result in a slightly different final RB. The same remark applies for step R-2) where different rule reduction orderings may yield different outcomes.

This is why in the following section we will present an enhanced variant of the SR algorithm, where we will try to eliminate the ordering effect described above while maintaining a reasonably

low number of evaluations. Based on this, we will also present a way to parametrize the accuracy-complexity trade-off of our method.

C. Selection-Reduction Method with Near-Optimal Ordering

When the subproblems are not independent, the *order* in which they are tackled by greedy approaches (which do not reconsider the solutions found) can influence the final result. This order can be seen as a random permutation of the input subspaces, which associates to each subspace a position in a queue. Then the input subspaces are evaluated in the order in which they appear in the queue. For the system in Figure 5 a possible ordering would be: $R_7 - R_2 - R_1 - R_3 - R_5 - R_8 - R_4 - R_9 - R_6$.

Now, let P be the set of all random permutations (or orderings) of the input subspaces, and $Err(P(i))$ be the error of the system measured by running the SR method with the i-th ordering. Let the optimal ordering of system be the one which minimizes $Err(P(i))$. Therefore, by running the SR method on the optimal ordering, one can reach the global optimum on the system's error surface[3].

However, finding the optimal ordering is not trivial, as there are $N!$ possible permutations for the N input subspaces. Since we cannot afford such a complex search, in the following we will use an approximative, near-optimal ordering, whose search requires $O(N)$ iterations.

`Algorithm 2a - Selection Stage with Near-Optimal Order:`

S-1) Compute ORB (the original rule base of the system), $MCRB$ (the multi-consequent rule base of the system) and initialize FRB (the final rule base of the system) as described in step S-1) of `Algorithm 1a`. Compute RC (the relative complement of ORB in $MCRB$) as presented in step S-2) of the same algorithm.

S-2) Next, we determine the near-optimal order in which to explore the input subspaces. Thus, for each input subspace I (parsed in no particular order) do:

- One by one, evaluate all the non-dominant consequents available for that input subspace by inserting each non-dominant rule from RC in FRB as explained in step S-4) of `Algorithm 1a`. Let $Err(I_j), \forall j = 1...c$, be the error of the system associated with the j-th consequent from the I-th input subspace (where $j = 1$ corresponds to

[3]Please note that the global optimum can also be obtained through a direct evaluation of every candidate RB from the pool.

the original dominant consequent and $Err(I_1)$ is the error associated with using ORB as rule-base).

- Store the error associated with each consequent. We recall that the only thing that changes between these inner iterations is the index of the consequent, and that all the other aspects of the system are the same. Moreover, besides the rule associated with input subspace I, all the other rules are taken directly from ORB; hence, at this stage we *do not update* the RB.
- Sort the above errors ($Err(I_j), \forall j = 1...c$) in ascending order and determine the *best consequent* (BC) for subspace I as the consequent attaining the lowest error among the c possible consequents for subspace I:

$$BC(I) = \underset{j=1...c}{\operatorname{argmin}}\{Err(I_j)\}.$$

The error corresponding to the best consequent, denoted *best error* for subspace I is also stored:

$$BErr(I) = \min_{j=1...c} Err(I_j) = Err(I_{BC(I)})$$

S-3) Once we have determined the best consequent and the best error for each input subspace, we *sort the list of subspaces* in ascending order of their corresponding *best error* ($Err(I_{BC(I)})$). Figure 6 illustrates this step for a system with 9 input subspaces. The order of the subspaces in the sorted table is the order in which they will be later explored.

Therefore, we want to explore first those subspaces whose adjustment (replacement of the dominant consequent by the non-dominant ones) gives the lowest error. This is somehow similar to the steepest descent algorithm. The rationale for this is that a change in those subspaces is likely to have a great influence on the neighboring subspaces.

Generally, the particular permutation produced by the above algorithm is different from the optimal one. Nonetheless, since we explored all the subspaces before generating this ordering, we consider it to be near-optimal, and as will be shown in Section V, this ordering constantly outperformed the random ones obtained with the original SR method.

S-4) Based on the exploration order defined above, the input subspaces are once again evaluated as explained in step S-4) of `Algorithm 1a`, and if the global error of the system is improved, the RB is updated with the best consequent for that input subspace.

For example, in Figure 6 the first input subspace to be explored is subspace 3.

Subspace Index	Best Error In Subspace
1	0.53
2	0.74
3	0.24
4	0.45
5	0.35
6	0.64
7	0.77
8	0.91
9	0.82

Exploration Order of the Subspaces	Sorted Best Error
3	0.24
5	0.35
4	0.45
1	0.53
6	0.64
2	0.74
7	0.77
9	0.82
8	0.91

Fig. 6. The process of sorting the list of available input subspaces based on their corresponding best error measure. The final order in which the subspaces are explored is shown in the *Right* table.

S-5) *Optional:* Once we finish evaluating a given subspace, the ordering can be refreshed by performing step S-3) using the updated RB. The subspaces that were already evaluated are excluded from future orderings, thus reducing the iterations performed. We will refer to this operation as *ordering refresh* and let P_S denote the number of ordering refreshes performed in the selection stage. By convention, when the order is computed only once (without being refreshed), $P_S = 1$. For the original SR method, since no ordering is computed (and a random one is used instead), we have $P_S = 0$.

The above algorithm is completed when all the subspaces are evaluated. Therefore, the number of ordering refreshes to perform, P_S, can be seen as a trade-off parameter between accuracy and complexity.

Once the selection stage is completed, the same principle of near-optimal ordering can also be applied to the reduction stage.

`Algorithm 2b - Reduction Stage with Near-Optimal Order:`

R-1) Define and initialize the auxiliary rule-base $ARB = FRB$.

R-2) Determine the near-optimal order in which to remove the rules:

- One at a time, remove each rule R_i from ARB and store $Err(R_i)$, which is the error of the system if $\{ARB\} \setminus R_i$ is used as rule-base, i.e. if only rule R_i is removed.
- Just as in the selection stage, sort the above errors (in ascending order) and store the associative ordering of the rules. This will be the order in which the rules will be visited in the following step.

R-3) Given the above-mentioned exploration order, we redo step R-2) of `Algorithm 1b`, and discard from FRB the rules which negatively affect the performance of the system.

R-4) *Optional*: Once a rule R_i has been definitively removed from FRB, we can refresh the above-derived exploration ordering. Analogous to the P_S parameter from the selection stage, we denote with P_R the number of ordering refreshes performed in the reduction stage ($P_R = 1$ if the order is computed just once).

A closer look at `Algorithm 2` shows that it is time-deterministic and that the number of evaluations performed is a function of both P_S and P_R. The interested reader can refer to Appendix A for a complete analysis on the complexity of the above method.

While the ***Reduction*** stage described in `Algorithm 2b` aims to remove those "noisy" rules which negatively affect the accuracy of the system, in the next section we will present a somewhat complementary approach to further prune the rule-base, by finding and removing the most redundant rules in the rule-base.

IV. A Redundancy-Based Analysis for Highly Interpretable Rule-Bases

Following the Precise and Fast Fuzzy Modeling approach, once an accuracy-oriented RB is obtained, the next step aims to improve its interpretability. According to [39], [7] interpretability can be defined as "the possibility to estimate the behavior of the fuzzy system simply by reading and understanding its rule-base" [7]. Although many other factors account for the overall interpretability of a system (see [40] for a detailed analysis), in this section we will restrict ourselves to the study of fuzzy rule-bases and their effect on interpretability.

It has long been known ([41]) that automatically induced fuzzy rules generally fail to match the high semantic level and interpretability of expert rules. One of the reasons is that, unlike human experts, a RB learning method tries to fit the distribution of the data while ignoring the semantic dependencies between the rules. This often leads to *redundant* RBs, where two or more

rules encode approximately the same knowledge. These rules are generally superfluous in that they do not contribute significantly to the accuracy of the system. Therefore, more compact and interpretable rule-bases can be obtained by simply removing these rules from the RB. Moreover, removing the highly redundant rules lightens the RB and facilitates the dissemination of information toward a non-specialist audience. Here we propose a new index to measure the redundancy of the fuzzy rules based on how these rules influence the numerical examples around them. This index is then used to detect and then suppress the most redundant rules of the rule-base, thus increasing the interpretability of the system.

According to [42] we can distinguish between two types of rule redundancy: overlap redundancy and interpolation redundancy. The first type is generally addressed either by defining some similarity measure between the fuzzy subsets, and then fusing the most similar subsets [43], [44], or by means of clustering algorithms where less influential clusters are iteratively removed [45]. On the other hand, interpolation redundancy allows the removal of fuzzy rules from those "plateau areas", where other rules provide more or less the same information (see [46]).

It is worth noticing that other methods based on orthogonal transforms, such as singular value decomposition, exists in the literature [47]. However, due to the transformed input space, these methods offer poor interpretability since the produced rules are hardly comprehensible for human users [41].

In this paper only the interpolation redundancy is treated, as we consider that choosing a uniformly partitioned universe of discourse (as the one illustrated in Figure 5) does not lead to high levels of overlapping between the fuzzy rules. Unlike other approaches presented in the literature which are based only on some relative distances between the rules and on the structure of the rules [46], [48] our proposed index also considers the concept of *"joint influence"* of two rules R_i and R_j on the set of numerical in-out data-pairs.

Let us first state that a similar mechanism, based on the *"co-firing"* of two fuzzy rules was proposed in [49] with the aim to facilitate a visual analysis of the rule-bases which should help spot the inconsistencies between the rules. However, the *"co-firing"* of fuzzy rules is based solely on the number of samples simultaneously activating both rules, without taking their degree of activation into account. Moreover, in [49] rule-base reduction is done manually by a human operator based on three indicators: the co-firing of the rule, its covering degree and the goodness of the rule (a measure of how well the rule classifies the instances covered). We believe that

none of the above are clear indicators for the redundancy of a given fuzzy rule with respect to its rule-base.

Therefore, in the following we will introduce a method allowing to compute a redundancy index for each fuzzy rule in a given rule-base. Using this index, the highly redundant rules can be easily identified and then removed.

The main intuition behind our approach is that, given a rule-base and a set of numerical in-out examples, a certain rule R_i is considered *redundant* if the subset of data points activating this rule, also activates "neighboring" rules which infer an outcome similar to that of R_i. Hence, if R_i is to be removed, the data points which it "*influenced*" will be taken into account by the group of neighboring rules.

On the other hand, non-redundant rules are generally *isolated* rules sharing little data with the other rules of the system. Based on this, we can distinguish between two main types of isolated rules:

1) rules positioned at the extreme points of the input space, which characterize the behavior of the system for limit values of the input variables.
2) rules which model abrupt changes in the behavior of the system, i.e. behavioral discontinuities.

In order to properly quantify the fuzzy rules redundancy, the index described hereafter relies on the concepts of *distance* between fuzzy rules and also on the *influence* the rule has over the dataset.

A. A Rule-base Redundancy Index

Let RB be a rule-base with N fuzzy rules. Let $s_j \in \Re^{M+1}, \forall j = 1, ..., N_S$ (M input variables and one output) be a set of numerical input-output samples. Next, let us define the activation degree of the rule R_i by the sample s_j as follows:

$$\delta_{s_j}^{R_i} = \left(\prod_{m=1}^{M} \mu_{s_j^{Im}} \right) \mu_{s_j^O} \tag{6}$$

where $\mu_{s_j^{Im}}$ is the membership degree of the sample s_j for the fuzzy subset of the m-th input variable of rule R_i (e.g. for R_i : IF A is A_1 THEN B is B_1, $\mu_{s_j^{I_1}}$ represents the membership degree of the sample s_j to the fuzzy subset A_1 of input variable A); $\mu_{s_j^O}$ represents the membership degree for the output variable.

Based on the above, let us further define $P_{R_i} = \{s_j | \delta_{s_j}^{R_i} > 0\}$ as the subset of samples which activate R_j with a positive activation degree $\delta_{s_j}^{R_i}$.

Afterwards, let $D(R_i, R_j)$ be the *distance* between rules R_i and R_j, where the rules are seen as points in a $(M+1)$-dimensional space. When the universes of discourse of the input and output variables are uniformly partitioned, this distance can be easily defined using the Manhattan distance (cityblock). For example, the Manhattan distance between the following rules:

$$\begin{aligned} R_i &: \text{IF } A \text{ is } A_1 \text{ and } B \text{ is } B_2 \text{ THEN } C \text{ is } C_1 \\ R_j &: \text{IF } A \text{ is } A_3 \text{ and } B \text{ is } B_1 \text{ THEN } C \text{ is } C_2 \end{aligned} \tag{7}$$

is $D_M(R_i, R_j) = |1-3| + |2-1| + |1-2| = 4$.

Please note that the above uses an unweighted version of the Manhattan distance. However, one can choose to give different weights to the different input variables, or to weight differently the input and the output variables.

Next, given two fuzzy rules R_i and R_j, and a certain sample s, let $C_s(R_i, R_j)$ denote the *common influence* of rules R_i and R_j on s:

$$C_s(R_i, R_j) = \frac{\min(\delta_s^{R_i}, \delta_s^{R_j})}{\delta_s^{R_i}}, \forall s \in P_{R_i} \tag{8}$$

Please note that $C_s(R_i, R_j)$ is *not* symmetrical, as it is defined with respect to the rule R_i. Expressed in words, $C_s(R_i, R_j)$ represents "the percentage of the activation of rule R_i by the sample s, which is taken into account, or compensated, by the rule R_j". As shown in Figure 7, when sample s activates R_j with a higher degree than R_i, the common influence $C_s(R_i, R_j) = 1$, indicating that rule R_j characterizes s better than R_i.

Next, the distance $D(R_i, R_j)$ between rules R_i and R_j is used to normalize the common influence:

$$C_s^{Norm}(R_i, R_j) = \frac{C_s(R_i, R_j)}{D(R_i, R_j)}, \forall s \in P_{R_i} \tag{9}$$

The *global* normalized common influence for rule R_i and sample s is computed by simply summing (9) over the whole rule-base:

$$C_s^{Norm}(R_i, \{RB\}) = \sum_{r=1}^{N} C_s^{Norm}(R_i, R_r), \quad r \neq i \tag{10}$$

The equation above indicates the extent to which the influence of rule R_i on the sample s is compensated by the ensemble of fuzzy rules in the rule-base RB. Thus, high values for

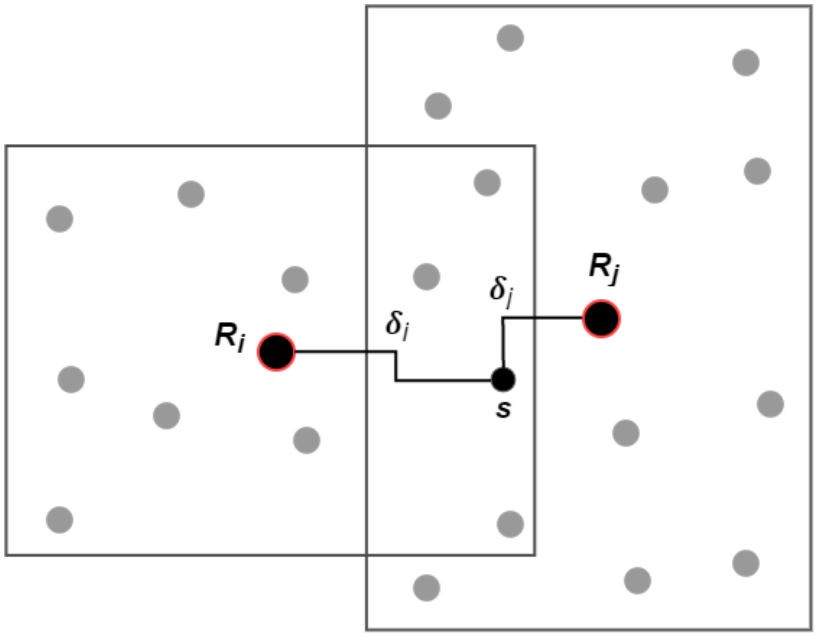

Fig. 7. Illustration of the common influence of rules R_i and R_j on sample s. Here $\delta_s^{R_j} > \delta_s^{R_j}$, which geometrically amounts to s being "closer" to R_j. Hence $C_s(R_i, R_j) = 1$, meaning that R_j compensates R_i with respect to sample s.

$C_s^{Norm}(R_i, \{RB\})$ suggest that R_i is redundant in $\{RB\}$ with respect to sample s. It should be noticed that if R_i does not share samples with the other rules (no other rule co-fires with R_i) then $C_s^{Norm}(R_i, \{RB\}) = 0$, meaning that R_i represents an isolated rule which should never be removed.

Considering the above, the complete algorithm to compute the redundancy index for a given rule-base is presented below.

`Algorithm 3 - Redundancy Index of Fuzzy Rules:`

RI-1) For each fuzzy rule R_i, compute the activation degrees $\delta_{s_j}^{R_i}, \forall s_j \in P_{R_i}$ using equation (6).

RI-2) For a given rule R_i, the global normalized common influence $C_{s_j}^{Norm}(R_i, \{RB\})$ is computed using equation (10) for *all* samples $s_j \in P_{R_i}$.

RI-3) Next, the redundancy index of the fuzzy rule R_i in the rule-base RB is computed over the whole set of samples $s_j \in P_{R_i}$ as the arithmetic mean of the individual $C_{s_j}^{Norm}(R_i, \{RB\})$

values:

$$RI\{R_i\} = \frac{1}{N_S^{R_i}} \sum_{j=1}^{N_S^{R_i}} C_{s_j}^{Norm}(R_i, \{RB\}) \tag{11}$$

where $N_S^{R_i} = |P_{R_i}|$, i.e. the number of samples activating rule R_i with an activation degree $\delta_{s_j}^{R_i} > 0$.

Iterating the above steps for all the rules $R_i \in \{RB\}$ allows us to associate a redundancy index to each rule. By using this index, we can easily identify the highly redundant rules in a given rule-base (according to the proposed index, the most redundant rule is the one which maximizes RI) and then remove them in order to improve the readability of the rule-base and the overall interpretability of the system.

Once the most redundant rule is removed from the rule-base, the redundancy index for the remaining rules must be recomputed using `Algorithm 3`. This is needed because once a rule is suppressed, the common influences of its neighboring rules must be recalculated.

Please note that the above algorithm is limited to the case of rule-bases, and does not alter the membership functions of the fuzzy subsets. The possibility of combining and harmonizing the proposed algorithm with a similarity-based fuzzy sets fusing technique is still to be investigated as it could optimize several interpretability criteria at once.

Therefore, rule-base simplification based on the above redundancy index leads to smaller rule-bases consisting only of those rules which are essential for the phenomenon to be modeled (the "pillars" of the system). Furthermore, as we will show in practical examples in Section V, these rule-bases, although smaller, are still very accurate. Thus, the general knowledge underlying a given phenomenon can now be expressed using only $N' < N$ rules, which helps to disseminate and better understand the information.

In the following section the proposed algorithms for RB learning and simplification are tested on several real-world and synthetic problems and assessed according to the three fundamental indicators of the Precise and Fast Fuzzy Modeling methodology: accuracy, speed and interpretability.

V. Experimental Results

In the first part of this section we will start by evaluating the RB learning behavior of the proposed *Selection-Reduction* methods and compare their performances with the state-of-the-

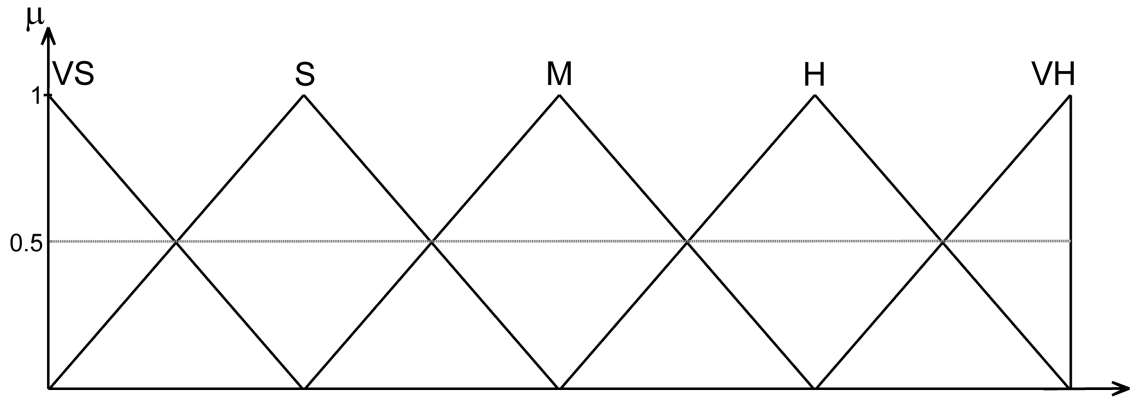

Fig. 8. A uniformly partitioned linguistic variable with five fuzzy subsets defined by triangular membership functions.

art methods presented in Section II. Afterwards, the efficiency of our redundancy-based index is tested on the previously-obtained rule-bases by showing that we can remove a significant percentage of rules and still obtain good performances from the system.

A. Experiments in RB Learning

Since the tests below only deal with the rule-base of the system, we will consider that each variable (input *and* output) is uniformly partitioned on its universe of discourse by triangular-shaped membership functions, as illustrated in Figure 8. The granularity of the variables, i.e. number of linguistic labels, will be explicitly specified for each test.

In all cases, the following inference operators were used: the *arithmetic product T-norm* to model the *AND* connections between input variables[4], and the Mamdani *min T-norm* and *max T-conorm* as implication and aggregation operators respectively. Defuzzification is carried out by the method of the *centroid*.

The RB generation methods evaluated in this section and their acronyms are: the *Selection-Reduction* method (SR), the *Selection-Reduction* method boosted by rule ordering (SRO), the original Wang and Mendel method (WM), the WM completed method (WMC) and the GA-driven COR method (COR-GA). Since the three iterative methods rely on the multi-consequent rule-base ($MCRB$) we have chosen a common value for the number of candidate consequents,

[4]Please note that all the RB generation methods assessed therein produce only *AND* rules, and therefore there is no need to model the *OR* connections between input variables.

$c = 2$. While higher values for parameter c generally lead to increased numerical performances, they also increase the number of evaluations needed to harvest the best rule in each subspace. Unless otherwise stated, for the SRO method, only one ordering refresh is performed in both the *Selection* and *Reduction* stages, i.e. $P_S = P_R = 1$. The default value of the removal threshold is set to $S[\%] = 0$ for both SR and SRO methods (unless explicitly stated otherwise).

In the case of the COR-GA method, we used the following parameters: a population size of 50 binary-coded individuals, a mutation probability of 0.01 per gene, a cross-over probability of 0.8 and a stall criterion of 15 generations (the algorithm stops if no better chromosome is found after 15 generations).

Finally, the three fundamental criteria of the PFFM methodology will be quantitatively assessed by the following performance indicators:

- *accuracy* - measured by the well-known Mean Square Error metric:
$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\hat{y}_i - y_i)^2$$
where $\hat{y}_i$ is the prediction of the system for the i-th data-point, and y represents its true output value.
- *speed* - for iterative methods, *speed* is measured as the total number of evaluations needed to learn a new rule-base; it is therefore a metric of how fast the learning procedure is.
- *interpretability* - in the absence of human evaluators, in this paper RB interpretability is evaluated as the final number of rules obtained; therefore, a RB with fewer rules will be considered more interpretable (compactness criterion for rule-bases).

1) Function Modeling: Let us consider the highly non-linear function defined by the equation below and illustrated in Figure 9.

$$F(x,y) = \exp\left(-\frac{x^2}{4}\right) + \exp\left(-\frac{y^2}{4}\right)$$
$$\forall x, y \in [-5,5]; \quad F(x,y) \in [0,2].$$

To test the approximation behavior of the above-mentioned RB generation methods, we designed a Mamdani fuzzy system with 2 input variables and 1 output variable. The training set consists of 6400 datapoints collected by uniformly sampling the 2D input space and evaluating the function at those locations. The test set is composed of 2500 datapoints, i.e. 28% of the total (train+test) set, obtained by randomly sampling the input space and evaluating the function.

For simplicity, the universe of discourse of the input and output variables was partitioned into an equal number of fuzzy subsets and we conducted two experiments using *three* and then *seven* subsets. For a given condition, each method was learned on the train set and tested on the test set (the same sets were used for all methods). This procedure was repeated a number of $K = 20$ times and the average results are collected in Table I and Table II, where for each of the above indicators the best result is shown in bold. For some indicators, the standard deviation around the mean is also marked in parentheses.

As can be seen, when the complexity of the space to be searched is low, as is the case of the *three* fuzzy subsets condition, all three iterative methods (SR, SRO and COR-GA) yield the same solution. This solution is most probably the optimal one within the space of all possible solutions. This behavior, which was first observed in [35], indicates that for low complexity problems the heavy computation of the GA does not bring any additional performances to the system and that more subtle methods (such as SR or SRO) converge much more quickly towards the desired optimum.

In the high-complexity search space of the *seven* subsets condition, the exhaustive exploration of the COR-GA method yields the rule-bases with the lowest error rates of all five methods. However, this comes with two major drawbacks, highlighted in Table II: a) the slow convergence speed of the method, which is a critical deficiency for self-adaptive FRBS, and b) the high number of rules which limits the interpretabiliity of the system.

With regard to the second criterion, it should be noticed that the COR-GA method constantly reaches a higher number of rules than the $7 \times 7 = 49$ fuzzy subspaces explored. This means that the corresponding rule-bases contain a small number of "contradictory rules" (two or more rules sharing the same antecedents but with a different consequent). Please note that this area of the solution space is "closed" to the SR and SRO methods, since they are forced to pick the best rule from each subspace. Therefore, we did another test where COR-GA was forced to limit the number of rules to the number of input subsets (49 in this case). This was possible by adding a proportional penalty factor to those candidate rule-bases for whom $NR > 49$. We call this new method FIX-COR-GA, and its results can be found in Table II.

Therefore we observe that limiting the solution space to that of the SR and SRO methods has a huge impact on the performance of the COR-GA method, whose error increased by more than 15% on the training set and more than 10% on the test set. This shows that, *over the*

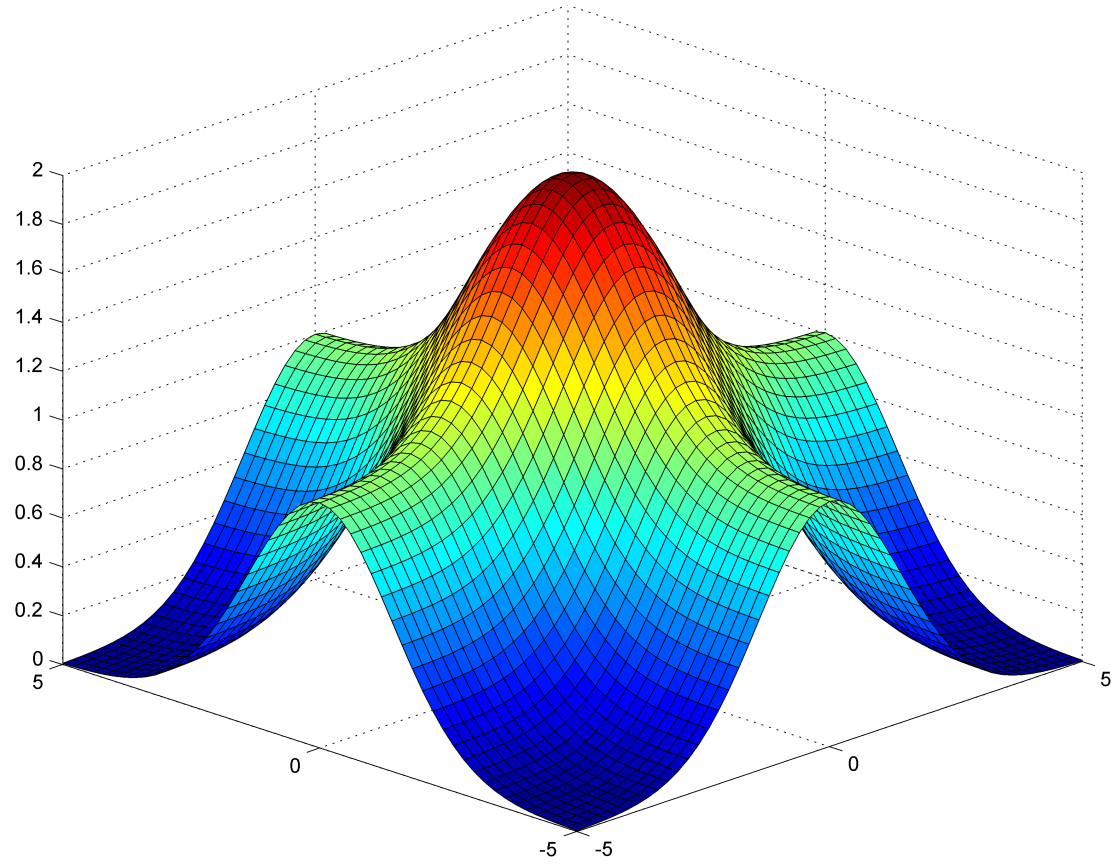

Fig. 9. Illustration of the bivariate sum of exponentials function.

same solutions space, the SRO method achieves performances comparable to those of the COR methods, and needs to perform over one order of magnitude fewer evaluations. Nevertheless, it should be noticed that, as illustrated in Table II, the proposed SR and SRO methods obtained the lowest number of rules among all the methods tested therein.

Next, for the *seven* fuzzy subsets condition only, we performed an experiment where we varied the number of ordering refreshes in the *Selection* and *Reduction* stages for the SRO method, i.e. P_S and P_R. For each parameter, seven possible values were tested: $\{0, 1, 2, 3, 4, 5, N\}$, where a value of 0 means that no pre-ordering is performed (in this case the ordering is performed randomly as in the SR method), while a value of N means that the order is refreshed after the evaluation of each input subspace in the *Selection* stage or after the removal of each rule, in the *Reduction* stage. This corresponds to steps S-5) and R-4), respectively, from `Algorithm 2`.

TABLE I

EXPERIMENTAL RESULTS FOR THE FUNCTION MODELING PROBLEM USING *three* FUZZY SUBSETS FOR EACH VARIABLE.

	MSE_{train}	MSE_{test}	NR	TE
WM	0.1986	0.2051 (0.0239)	**9**	–
WM-C	0.2278	0.2300 (0.0142)	**9**	–
COR-GA	**0.1089**	**0.1101** (0.0086)	**9**	1050 (48.4)
SR	**0.1089**	**0.1101** (0.0086)	**9**	**14** (0.0)
SRO	**0.1089**	**0.1101** (0.0086)	**9**	27 (0.0)

TABLE II

EXPERIMENTAL RESULTS FOR THE FUNCTION MODELING PROBLEM USING *seven* FUZZY SUBSETS FOR EACH VARIABLE.

	MSE_{train}	MSE_{test}	NR	TE
WM	0.0265	0.0272 (0.0019)	49	–
WM-C	0.0265	0.0272 (0.0019)	49	–
COR-GA	**0.0103**	**0.0107** (0.0010)	53.5 (1.1)	3760 (510.31)
FIX-COR-GA	**0.0119**	**0.0118** (0.0019)	48.15 (1.04)	4830 (1327.98)
SR	0.0155	0.0161 (0.0021)	**46.7** (1.59)	**90** (0.0)
SRO	0.0120	0.0122 (0.0015)	**46.7** (1.97)	178.3 (0.47)

Please note that we are allowed to stop the algorithm if no improvement is observed from one ordering to the next. Then, the subspaces in the *Selection* stage and the rules to be removed in the *Reduction* stage are evaluated in the order provided by this last ordering.

The average results after $K = 20$ repetitions are collected in Table III, where for convenience we also added the results of the special case defined by $(P_S = 0; P_R = 0)$ which corresponds to the SR method without rule ordering. We recall that, by definition, when $P_S = 0$ or $P_R = 0$, a random rule evaluation order is used in the Selection or Reduction stage, respectively. To improve readability, standard deviations were omitted.

We notice that by varying P_S from 0 to N the error on both train and test sets decreases monotonically with P_S. Moreover, starting from $P_S = 5$ we obtain a *lower error than the FIX-COR-GA method* (see Table II) on both sets. While the monotonically decreasing nature is not guaranteed to occur for each case (due to the greedy character of the method), it validates the principle that greedy rule selection, preceded by a carefully chosen rule ordering, achieves state-of-the-art results which compare favorably with the sophisticated COR methods.

We also notice that starting from $P_S = 5$ the error saturates and stops decreasing, showing the limitations of our greedy approach, and therefore the limitations of the SRO method. However, this also means that the optimal value for P_S is rather small and that a lower error could be attained using a relatively small number of evaluations. Further experiments are needed to link the optimal value of P_S to the complexity of the problem.

On the other hand, varying the number of orderings over the *Reduction* stage, i.e. P_R, does not significantly alter the error of the system. This could be due to the clean nature of this dataset which contains very few, if any, "malignant" rules. However, $P_R = 1$ (the ordering is performed just once) should always be preferred to $P_R = 0$ (random ordering). This viewpoint is also confirmed by Table III.

2) Vibrotactile Perception Evaluation: The Vibrotactile Perception problem [50] consists of predicting the comfort level for a series of $N_S = 48$ vibrotactile stimuli characterized by three physical attributes: energy (*E*), velocity (*V*) and spectral complexity or entropy (*S*). The comfort values for the 48 stimuli were obtained by asking a panel of 16 users three times, and averaging the results. The results of this experiment will be later exploited by automotive experts to design haptic or vibrotactile effects for automobile touch-screen displays.

As described in [50], the granularity of the variables is 5 fuzzy subsets for *E*, 3 for *V* and *S*

TABLE III

THE EFFECT OF THE ORDERING REFRESH PARAMETERS OF THE SYSTEM (P_S AND P_R) ON THE PERFORMANCE INDICATORS.

P_S	P_R	MSE$_{train}$	MSE$_{test}$	NR	TE
0	0	0.0162	0.0157	45.95	90
0	1	0.0159	0.0154	46.95	138.2
1	1	0.0120	0.0120	46.80	178.3
2	1	0.0119	0.0119	46.9	217.2
3	1	0.0117	0.0116	47.5	255.4
4	1	0.0113	0.0112	48.4	292.7
5	1	**0.0110**	**0.0111**	49	329
N	1	**0.0110**	**0.0111**	49	840

P_S	P_R	MSE$_{train}$	MSE$_{test}$	NR	TE
0	0	0.0162	0.0157	45.95	90
1	0	0.0122	0.0122	46.20	130
1	1	0.0120	0.0120	46.80	178.3
1	2	0.0122	0.0121	46.2	211.2
1	3	0.0125	0.0123	45.7	257.5
1	4	0.0122	0.0121	46.3	275.7
1	5	0.0122	0.0122	46.2	293.7
1	N	0.0122	0.0120	46.4	297.7

and 5 again for the output variable (the predicted comfort value). Given the small size of the dataset, both SR and SRO methods used a rule removal threshold $S[\%] = 5$. This means that a given rule is removed only if the error decreases by at least 5%.

The $N_S = 48$ vibrotactile stimuli were randomly divided into a train set of 36 samples (75%) and a test set of 12 samples (25%). All methods were trained and tested on the same sets, and the procedure was repeated for $K = 50$ random splits of the 48 vibrotactile stimuli. The mean values along with the standard deviations are collected in Table V.

We first notice that the three iterative methods (SR, SRO and COR-GA) are superior to the non-iterative ones (WM and WM-C), and that although COR-GA achieves the best results on the training set, it fails to maintain its performance on the test set, where the SRO method obtains the best result. This clearly indicates an over-fitting behavior for the COR-GA method when the amount of available data is sparse.

On the other hand, the SR and SRO methods provide comparable performances on both training and test sets, with only a marginal advantage for the SRO method. We believe that this behavior has something to do with the sparsity of the data and with the fact that certain zones on the input space grid are inaccessible (e.g. no vibrotactile stimulus can simultaneously have high energy and high velocity [50]). This leads to a considerable amount of "holes" within the input space which reduces the mutual dependency of the rules. For example, if in Figure 5 rule

R_1 is omitted and replaced by a hole, the relative independence of neighboring rules R_2, R_4 and R_5 is increased. Moreover, if rules R_2, R_5 and R_6 are replaced by holes, then rule R_3 becomes completely independent and can be optimally solved by greedy methods. This leads to the following conclusions: a) greedy based methods like SR and SRO are well suited for sparse data problems or problems with constraints on the input distribution; b) for those particular problems, the orderings performed by the SRO method have very little influence on the performances of the system (indeed, permuting the set of independent subproblems of a given problem has no effect on the outcome of the problem).

Table IV illustrates a particular RB for this problem obtained by the SRO method, and chosen as the one with the lowest generalization error. It should be noticed that this RB efficiently encodes the knowledge beneath the vibrotactile perception phenomenon in a way that is easy to read and diagnose by the human user. These rule-bases, which establish links between the physical properties of the vibrotactile stimuli and their perceived comfort, are then used by automotive engineers and ergonomics experts during the process of designing pleasant haptic effects. This is why, in this case, providing compact and interpretable rule-bases, which are able to accurately synthesize users' subjective evaluations, is of crucial importance.

3) Airfoil Noise Prediction Problem: The problem consists of predicting the self-noise level (in dB) for a series of airfoil blades undergoing aerodynamics tests with varying parameters: frequency, angle of attack, chord length, velocity and side displacement [51]. A total number of $N_S = 1503$ input-output pairs are available. Once again, we tested the RB selection methods on two initial conditions: using *three* and *seven* uniformly partitioned fuzzy subsets for each input and output variable. In both conditions, we used 75% of the available samples for training and the remaining 25% for testing. The mean results after $K = 20$ repetitions were collected in Table VI and Table VII, respectively.

A quick look at the *three* fuzzy subsets condition confirms the conclusion of the Vibrotactile Perception experiment, with the COR-GA method achieving the best error rate on the training set, but not on the test set, where the best result is achieved by the SRO method. The SRO method also yields the most compact rule-bases of the five methods tested and requires less than 4% of the evaluations performed by the COR-GA method.

In the extremely large space of the $seven$ fuzzy subsets condition, the COR-GA outperforms the SR and SRO method, but only by a negligible margin. However, for a self-adaptive system

TABLE IV

RULE-BASE OBTAINED WITH THE SRO METHOD FOR THE VIBROTACTILE PERCEPTION PROBLEM. THE VIBROTACTILE STIMULI ARE DEFINED BY THE INPUT VARIABLES: ENERGY (E), VELOCITY (V) AND SPECTRAL COMPLEXITY (S), AND BY THE OUTPUT VARIABLE: ESTIMATED COMFORT. ($MSE_{train} = 0.0164$; $MSE_{test} = 0.0066$).

E	V	S	**Comfort**
Very Small	Small	High	*Unacceptable*
Very Small	Medium	High	*Intolerable*
Very Small	High	High	*Unacceptable*
Small	Small	Medium	*Comfortable*
Small	Small	High	*V. Comfortable*
Small	Medium	Medium	*Acceptable*
Small	Medium	High	*Acceptable*
Small	High	High	*Acceptable*
Medium	Small	Small	*Intolerable*
Medium	Small	Medium	*V. Comfortable*
Medium	Small	Small	*Acceptable*
High	Small	Medium	*Acceptable*
Very High	Small	Medium	*Unacceptable*
Very High	Medium	Medium	*Intolerable*

TABLE V

EXPERIMENTAL RESULTS FOR THE VIBROTACTILE PERCEPTION PROBLEM.

	MSE_{train}	MSE_{test}	NR	TE
WM	0.0197 (0.0035)	0.0343 (0.0162)	13.9 (0.94)	–
WM-C	0.0207 (0.0032)	0.0355 (0.0170)	13.9 (0.94)	–
COR-GA	**0.0121** (0.0022)	0.0306 (0.0135)	**10.86** (1.25)	1444 (436.9)
SR	0.0134 (0.0027)	0.0295 (0.0146)	12.8 (1.12)	**19.6** (1.07)
SRO	0.0133 (0.0026)	**0.0292** (0.0145)	12.64 (1.22)	37.26 (2.16)

TABLE VI

EXPERIMENTAL RESULTS FOR THE AIRFOIL NOISE PREDICTION PROBLEM IN THE *Three* FUZZY SUBSETS CONDITION.

	MSE_{train}	MSE_{test}	NR	TE
WM	48.539 (1.203)	49.238 (2.198)	55.3 (1.031)	–
WM-C	50.128 (1.051)	50.774 (2.101)	55.3 (1.031)	–
COR-GA	**23.378** (1.512)	25.344 (1.963)	36.85 (3.437)	4585 (1074.5)
SR	24.848 (0.986)	26.517 (2.174)	39.55 (2.089)	**88.75** (1.77)
SRO	23.619 (1.254)	**25.299** (2.001)	**36.15** (2.059)	175.5 (3.546)

TABLE VII

EXPERIMENTAL RESULTS FOR THE AIRFOIL NOISE PREDICTION PROBLEM IN THE *Seven* FUZZY SUBSETS CONDITION.

	MSE_{train}	MSE_{test}	NR	TE
WM	14.907 (0.586)	18.439 (2.502)	314.2 (3.664)	–
WM-C	10.861 (0.395)	15.605 (1.791)	314.2 (3.664)	–
COR-GA	**9.219** (0.416)	**14.329** (1.345)	369.7 (15.748)	11612.5 (2736.2)
SR	9.386 (0.349)	14.494 (1.472)	267.15 (6.011)	**486.95** (5.924)
SRO	9.234 (0.33)	14.367 (1.518)	**266.8** (6.947)	971.9 (11.849)

it needs a prohibitively large number of evaluations and leads to a large number of rules. On the other hand, the SRO method achieves very compact rule-bases and performs less than 9% of the evaluations performed by the COR-GA method. While achieving results similar to the SRO method (both in terms of accuracy and number of rules), the unordered SR method performs around 4% of the evaluations of the COR-GA method.

The experimental results exposed before show that the proposed SR and SRO methods are very strong candidates for the task of rule selection and that in many situations they can successfully replace the combinatorial search algorithms used in the COR methodology. Regarding the compactness of the generated rule-bases, the SR and SRO methods always obtained fewer rules than the WM and WM-C methods, and in many cases they obtained fewer rules than the COR-GA method. Moreover, under the PFFM approach, where accuracy and speed are equally important, we advocate that our proposed methods are the best choice because of their high accuracy (in our experiments, at test time, the increase in accuracy of the SRO method can be as high as 122% compared to the WM method) and because of the reasonably small number of evaluations needed (between 5% and 9% of the total evaluations performed by the COR-GA method).

Now, while the SRO method generally performs better than the SR method, it does require a non-negligible number of additional evaluations, depending on the values of the P_S and P_R parameters. However, as proven in Appendix A, given the values for P_S and P_R, we can exactly compute the number of evaluations needed to learn the RB. Therefore, by knowing the time constraints in the environment where the system is to be deployed, we can let the SRO method automatically pick the values of P_S and P_R resulting in a number of evaluations TE which can be safely executed. Hence, the ordering-enhanced *Selection-Reduction* (SRO) method can be seen as a *parametrized trade-off* between accuracy and speed, where the influence of the parameters on the convergence speed can be exactly determined beforehand.

B. Experiments in Redundancy-based Rule Reduction

Once a RB is obtained by a given method, the results can be further processed to improve interpretability of the RB using the compactness criterion, i.e. with the aim to reduce its number of rules. Using the RB redundancy index described in `Algorithm 3`, we will experiment with the removal of the top $X[\%]$ most redundant rules and see how this affects the overall accuracy

of the system. Typically, we would expect the removal of these redundant rules to have a much smaller effect on the accuracy than the stochastic removal of an equivalent number of rules.

Therefore, for each of the three problems described earlier, we will consider the rule-bases obtained with the SRO method (using the default values $P_S = P_R = 1$), and choose as the final rule-base the one which achieves the lowest generalization error among the K repetitions performed in each set. Please note that, while this procedure is normally used in practice, the redundancy index described in `Algorithm 3` can be applied to any given rule-base.

The final rule-base along with the set of numerical examples are processed by `Algorithm 3`, in order to compute the redundancy index for each rule. Next, the top $X[\%]$ most redundant rules are removed from the rule-base and the accuracy of the system is evaluated *without* these rules. The accuracy is computed as the mean square error over the set of numerical examples[5]. In this paper we tested the following values for $X[\%]$: {0, 1, 5, 15, 25, 35, 50}, where the 0% case is equivalent to the unaltered rule-base. For each problem, the $\widehat{NR}$ column indicates the number of rules removed in a given iteration, i.e. $\widehat{NR} = \lfloor \frac{X \cdot NR}{100} \rfloor$.

For comparison purposes, we will also *randomly* remove an equivalent number of rules, i.e. $\widehat{NR}$ rules, from the rule-base and evaluate the accuracy on the same set of examples. This stochastic rule removal procedure will be repeated 100 times and the accuracy results will be averaged. In both the redundancy-based and stochastic rule removal, when a certain data sample does not trigger any rule, it is assigned to a default value α computed as the midpoint of the output variable's universe of discourse, i.e. $\alpha = \frac{min(Out)+max(Out)}{2}$.

1) Function fit: For the problem of function fitting, the two original conditions were evaluated: using *three* and *seven* fuzzy subsets for each input and output variable. The set of numerical examples consists of 6400 data points, uniformly sampled in the 2D space of the input variables. The results obtained are collected in Table VIII.

In both the *three* and *seven* fuzzy subsets conditions, the removal procedure based on the rule redundancy index produces very reliable systems, which are almost twice as accurate as the ones obtained by the stochastic removal procedure. It should be noted that the redundancy-based approach constantly produces better results than the stochastic one, for each condition and for

[5]Please note that in `Algorithm 3` the accuracy of the system is neither evaluated nor used in any way to determine the redundant rules. This means that we can safely use the same numerical examples to evaluate the system mean square error.

TABLE VIII

EFFECT OF REDUNDANCY-BASED RULE REMOVAL VS. STOCHASTIC RULE REMOVAL. EXPERIMENT ON THE FUNCTION MODELING PROBLEM. LEFT: THE *three* FUZZY SUBSETS CONDITION ($NR = 9$ RULES); RIGHT: THE *seven* FUZZY SUBSETS CONDITION ($NR = 49$ RULES).

$X[\%]$	$\widehat{NR}$	$\text{MSE}_{Redun.}$	$\text{MSE}_{Stochas.}$
0%	0	0.1089	0.1089
1%	1	0.1153	0.1337
5%	1	0.1153	0.1337
15%	2	0.1216	0.1710
25%	3	0.1279	0.2189
35%	4	0.1342	0.2490
50%	5	0.1808	0.3040

$X[\%]$	$\widehat{NR}$	$\text{MSE}_{Redun.}$	$\text{MSE}_{Stochas.}$
0%	0	0.0110	0.0110
1%	1	0.0112	0.0128
5%	3	0.0116	0.0159
15%	8	0.0135	0.0279
25%	13	0.0137	0.0399
35%	18	0.0202	0.0597
50%	25	0.0468	0.0977

each iteration considered.

Another important observation is that, in both conditions, the best accuracy-compactness ratio is achieved by removing 25% or 35% of the most redundant rules. Going further and removing 50% of the system's most redundant rules significantly impacts on the quality of the RB. This happens because, when all the redundant rules are removed from the RB, we are left only with the important rules, which can be seen as the pillars of the system. Forcing the method to go further with the removal, means that it must choose the most redundant rules within a group of very important rules, whose removal leads to a substantial deterioration of the accuracy. Empirical evidence presented here (and confirmed by the following two problems) suggest that a proper value for the number of redundant rules to remove ($X[\%]$) should be around 30%.

2) Vibrotactile Perception Evaluation: The same experimental procedure applied in the function fitting problem is employed for this problem also. The numerical dataset used to determine the redundancy index and later to test the pruned rule-bases, is the set of 48 vibrotactile stimuli. The original rule-base has $NR = 14$ rules and is chosen as the RB with the lowest generalization error among the rule-bases generated using the SRO method in Section V-A. As mentioned in Section V-A2, for this experiment, the interpretability of the rule-base is very important, since it will be used by automotive experts to design a series of haptic effects to be used in automobile

TABLE IX

EFFECT OF REDUNDANCY-BASED RULE REMOVAL VS. STOCHASTIC RULE REMOVAL ON SYSTEM ACCURACY. EXPERIMENT ON THE VIBROTACTILE PERCEPTION PROBLEM.

$X[\%]$	$\widehat{NR}$	$\text{MSE}_{Redun.}$	$\text{MSE}_{Stochas.}$
0%	0	0.0140	0.0140
1%	1	0.0139	0.0174
5%	1	0.0139	0.0169
15%	3	0.0139	0.0254
25%	4	0.0142	0.0314
35%	5	0.0170	0.0395
50%	7	0.0366	0.0490

touch-screen displays. Therefore the ability of the proposed index to quickly discard redundant rules will help the automotive expert focus on the relevant information in a rule-base.

Data collected in Table IX confirms the superiority of our redundancy index as an efficient proposal mechanism for rule removal. In this case, the best accuracy-compression ratio is achieved for $X = 25\%$, when we obtain a very compact rule-base which has more or less the same accuracy as the original rule-base. An interesting fact here is that the MSE *decreases* with the removal of certain rules, e.g. for $X = 1\%$. This might be due to the greedy nature of the SRO method and the limited number of orderings used ($P_S = P_R = 1$). Nevertheless, this MSE decrease reveals the complementary nature of the SRO method for rule-base generation and the redundancy index for rule-base optimization. It also shows that for noisy and uncertain data (such as users' perceptual assessments) cascading the SRO method and the redundancy index may have positive results on both the number of rules and the accuracy of the system.

3) Airfoil Noise Prediction Problem: Analogous to the function fitting problem, here again we used two initial conditions on the system variable: with *three* and *seven* fuzzy subspaces for each input and output variable. The original RB in the *three* subsets condition has $NR = 33$ rules, while in the *seven* subsets condition it has $NR = 264$ rules. In both conditions, we used the full set of 1503 numerical input-output pairs to compute the redundancy index and test the rule-bases. The results are given in Table X.

TABLE X

EFFECT OF REDUNDANCY-BASED RULE REMOVAL VS. STOCHASTIC RULE REMOVAL. EXPERIMENT ON THE AIRFOIL NOISE PREDICTION PROBLEM. LEFT: THE *three* FUZZY SUBSETS CONDITION ($NR = 33$ RULES); RIGHT: THE *seven* FUZZY SUBSETS CONDITION ($NR = 264$ RULES).

$X[\%]$	$\widehat{NR}$	$\text{MSE}_{Redun.}$	$\text{MSE}_{Stochas.}$
0%	0	22.6798	22.6798
1%	1	23.3215	23.9127
5%	2	23.4440	25.7668
15%	5	25.7415	30.0056
25%	9	28.8163	37.1680
35%	12	34.5830	43.4129
50%	17	37.0264	55.8683

$X[\%]$	$\widehat{NR}$	$\text{MSE}_{Redun.}$	$\text{MSE}_{Stochas.}$
0%	0	10.1769	10.1769
1%	3	10.2742	10.7171
5%	14	10.4959	12.9016
15%	40	11.3714	18.0818
25%	66	12.6680	22.8942
35%	93	15.1330	27.7895
50%	132	24.2040	35.9125

These results corroborate our previously presented findings, showing that, for every iteration and condition tested, the redundancy-based approach to rule removal is preferable to the stochastic one by a large margin. In both conditions, a very good compression ratio is achieved by removing around $X = 25\%$ of the most redundant rules, which increases the system MSE by 27% in the *three* fuzzy subsets condition, and by 24% in the *seven* fuzzy subsets condition (for the same $X[\%]$, the stochastic rule removal increases the MSE by 64% and 125%, respectively).

The three experiments presented above show that the rule-base of a system can be efficiently trimmed by dropping out its most redundant rules, allowing it to capture only the essence of a phenomenon. We also showed that our proposed redundancy-based rule removal approach preserves most of the accuracy of the system and largely outperforms stochastic removal procedures, for each experimental condition and iteration tested. An interesting research direction would be to study the extent to which this mechanism could be used to "regularize" the model in order to avoid overfitting.

VI. CONCLUSION

In this paper the Precise and Fast Fuzzy Modeling formalism was introduced, dealing with the generation of accurate rule-bases from numeric data followed by an efficient rule-base optimization procedure. The key feature of the Precise and Fast Fuzzy Modeling approach is its

special balance between the *accuracy* of the rules obtained, and the *speed* required to generate them.

This essential feature is achieved with the aid of the *Selection-Reduction* family of methods for RB learning, which was originally introduced in [35] and further developed therein. Of particular interest for the problem in question is the proposed parametric version of the *Selection-Reduction* method, representing the first important contribution of this paper. Under this revised learning procedure, a near-optimal exploration ordering for the input fuzzy subspaces is computed before the Selection and Reduction stages, respectively. We showed that this additional stage can improve the accuracy of the system, while also increasing the number of evaluations performed. Since the exploration order can be parametrically refreshed several times, it provides a simple, yet elegant way to fine-tune the accuracy-speed balance of the method.

We also showed that both the original and parametric versions of the method are time-deterministic, meaning that the number of evaluations needed to learn the rule-base can be precisely determined beforehand. This fact can be easily exploited for self-adaptive FRBS employed in real-time environments where time requirements are rather strict.

Another important contribution of this paper is the theoretical analysis of the *Selection-Reduction* method, providing qualitative justification for its usefulness within the fuzzy context. The justification is based on the observation that the input subspaces for most FRBSs exhibit low interdependence on one another, which in turn favors the optimal substructure property of the problem (e.g. Ruspini partitions guarantee that one data sample activates at most two fuzzy subspaces). This allows the greedy nature of the *Selection-Reduction* methods to achieve results comparable to the computationally expensive COR methods, while using only a tiny fraction of their learning time. Through three different problems and several initial conditions, we have empirically illustrated the advantages of the *Selection-Reduction* methods as a very fast and accurate solution for RB learning.

The third important contribution of this paper, which completes the Precise and Fast Fuzzy Modeling approach, is the introduction of a new index to quantitatively measure the redundancy of fuzzy rules. This index allows the identification of the superfluous rules which provide little, if any, relevant knowledge for the system. These rules can be later removed from the rule-base, and we have shown through comprehensive experiments that the resulting rule-bases, while considerably more compact, are still very accurate.

Therefore, the *Selection-Reduction* family of methods, along with the rule redundancy index, provide a complete solution for the rule-bases of FRBS. To further develop and extend the Precise and Fast Fuzzy Modeling paradigm, future research must address the optimization of the membership functions in a fast and reliable way. Another potential research direction would be a quantitative and empirical analysis of the relationship between rules interdependence and the efficiency of the *Selection-Reduction* methods.

APPENDIX A
COMPLEXITY OF THE SELECTION-REDUCTION METHOD WITH NEAR-OPTIMAL ORDERING

While computing the order in which the input subspaces are explored may increase the accuracy of the system, it also increases the number of evaluations needed to obtain the final rule-base. Nonetheless, the total number of evaluations needed can still be determined beforehand as a function of parameters P_S and P_R. Therefore, `Algorithm 2` can still be viewed as a time-deterministic one.

Let us recall that, when $P_S = 0$, i.e. a random exploration order is used in the Selection stage, the number of evaluations performed in the Selection stage is the cardinal of the relative complement of ORB in $MCRB$ (see `Algorithm 1a`):

$$TE_{Selection}(P_S = 0) = |\{RC\}| \tag{12}$$

When $P_S \geq 1$ (the order is computed at least once), the number of evaluations in the Selection stage is a function of P_S. In its simplest form, this function is:

$$TE_{Selection}(P_S) = \sum_{i=0}^{P_S} (|\{RC\}| - c \cdot i) \tag{13}$$

where c is the number of candidate (non-dominant) consequents used to determine $MCRB$[6].

Put into words, the above equation describes the following behavior: A) determine the first ordering by exploring *all* subspaces; B) once this ordering is obtained, change the original consequent of the first input subspace with respect to this ordering, for the best consequent for this subspace; once changed, this consequent becomes locked; C) compute the other $P_S - 1$ orderings, *without revisiting the subspaces* already locked by previous orderings (hence, the

[6]Please note that the number of rule consequents (c) may differ between input subspaces. Nonetheless, it is known beforehand.

$-c \cdot i$); D) visit and evaluate the remaining subspaces as described in step S-4) of `Algorithm 1a`; the exploration order is given by the *last* ordering, i.e. the P_S-th ordering.

Please note that other ordering refresh strategies can be used. For example, one may choose to refresh the exploration order of the subspaces after the evaluation of p subspaces, or when the drop in error is below a particular threshold. For the sake of simplicity, we will only use the function denoted by (13), without investigating other extensions.

Analogous to the Selection stage, the number of evaluations performed in the Reduction stage is also a function of P_R:

$$TE_{Reduction}(P_R) = \sum_{i=0}^{P_R} |\{FRB\}| - i \tag{14}$$

where $|\{FRB\}|$ is the number of rules in the final rule-base FRB. FRB is available after the Selection stage. In the above equation, the second term of the sum is $(-i)$ because we only evaluate the removal of the rule with the winning consequent, and not all candidate consequents.

Given the above, the total number of evaluations performed in the two stages is the sum of the evaluations performed in each stage:

$$TE = \sum_{i=0}^{P_S} (|\{RC\}| - c \cdot i) + \sum_{i=0}^{P_R} (|\{FRB\}| - i) \tag{15}$$

Please note that when $P_S = P_R = P_{SR}$, i.e. the same number of ordering refreshes is used in both stages, and if we neglect the second term from each sum, the above formula simplifies to[7]:

$$TE = (P_{SR} + 1) \cdot |\{MCRB\}| = (P_{SR} + 1) \cdot NR \tag{16}$$

Since $|\{RC\}|$, $|\{FRB\}|$, $|\{MCRB\}|$ are known beforehand, and since P_S and P_R are meta-parameters defined by the user, the exact number of evaluations performed by `Algorithm 2` can easily be determined in advance. This time-deterministic nature gives our approach the flexibility to adapt to new data, if time allows; or to keep its current structure, otherwise.

REFERENCES

[1] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[7]By neglecting the terms $(-c \cdot i)$ and $(-i)$, equation (16) become an upper bound for the actual TE.

[2] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *Fuzzy Systems, IEEE Transactions on*, vol. 14, no. 5, pp. 676–697, 2006.

[3] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. J. Rantz, M. Aud *et al.*, "Modeling human activity from voxel person using fuzzy logic," *Fuzzy Systems, IEEE Transactions on*, vol. 17, no. 1, pp. 39–49, 2009.

[4] S. Alayón, R. Robertson, S. K. Warfield, and J. Ruiz-Alzola, "A fuzzy system for helping medical diagnosis of malformations of cortical development," *Journal of Biomedical Informatics*, vol. 40, no. 3, pp. 221–235, 2007.

[5] L.-X. Wang, "Fuzzy systems are universal approximators," in *Fuzzy Systems, IEEE International Conference on*. IEEE, 1992, pp. 1163–1170.

[6] B. Kosko, "Fuzzy systems as universal approximators," *Computers, IEEE Transactions on*, vol. 43, no. 11, pp. 1329–1333, 1994.

[7] J. M. Alonso and L. Magdalena, "Special issue on interpretable fuzzy systems," *Information Sciences*, vol. 181, no. 20, pp. 4331–4339, 2011.

[8] R. Alcalá, J. Casillas, O. Cordón, A. González, and F. Herrera, "A genetic rule weighting and selection process for fuzzy control of heating, ventilating and air conditioning systems," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 3, pp. 279–296, 2005.

[9] R.-E. Precup and H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Computers in Industry*, vol. 62, no. 3, pp. 213–226, 2011.

[10] W. Siler and J. J. Buckley, *Fuzzy expert systems and fuzzy reasoning*. John Wiley & Sons, 2005.

[11] O. Cordón and F. Herrera, "A proposal for improving the accuracy of linguistic modeling," *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 335–344, 2000.

[12] J. Casillas, O. Cordón, and F. Herrera, "COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 32, no. 4, pp. 526–537, 2002.

[13] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 1, no. 1, pp. 116–132, 1985.

[14] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 6, pp. 1414–1427, 1992.

[15] X. Zeng and L. Koehl, "Representation of the subjective evaluation of the fabric hand using fuzzy techniques," *International Journal of Intelligent Systems*, vol. 18, no. 3, pp. 355–366, 2003.

[16] P.-C. Chang, C.-H. Liu, and R. K. Lai, "A fuzzy case-based reasoning model for sales forecasting in print circuit board industries," *Expert Systems with Applications*, vol. 34, no. 3, pp. 2049–2058, 2008.

[17] M. Blagojević, M. Šelmić, D. Macura, and D. Šarac, "Determining the number of postal units in the network–fuzzy approach, serbia case study," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4090–4095, 2013.

[18] L.-X. Wang, "The WM method completed: a flexible fuzzy system approach to data mining," *Fuzzy Systems, IEEE Transactions on*, vol. 11, no. 6, pp. 768–782, 2003.

[19] K. Nozaki, H. Ishibuchi, and H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy Sets and Systems*, vol. 86, no. 3, pp. 251–270, 1997.

[20] J. Alcala-Fdez, R. Alcala, and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 5, pp. 857–872, 2011.

[21] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, "A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 616–635, 2007.

[22] O. Cordón, "A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems," *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 894–913, 2011.

[23] J. Cózar, L. de la Ossa, and J. A. Gámez, "Learning TSK-0 linguistic fuzzy rules by means of local search algorithms," *Applied Soft Computing*, vol. 21, pp. 57–71, 2014.

[24] M. Mucientes and J. Casillas, "Quick design of fuzzy controllers with good interpretability in mobile robotics," *Fuzzy Systems, IEEE Transactions on*, vol. 15, no. 4, pp. 636–651, 2007.

[25] A. Ghanbari, S. Kazemi, F. Mehmanpazir, and M. M. Nakhostin, "A cooperative ant colony optimization-genetic algorithm approach for construction of energy demand forecasting knowledge-based expert systems," *Knowledge-Based Systems*, vol. 39, pp. 194–206, 2013.

[26] O. Castillo, E. Lizárraga, J. Soria, P. Melin, and F. Valdez, "New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system," *Information Sciences*, vol. 294, pp. 203–215, 2015.

[27] C. Caraveo, F. Valdez, and O. Castillo, "Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation," *Applied Soft Computing*, vol. 43, pp. 131–142, 2016.

[28] L. Gal, R. Lovassy, and L. T. Kóczy, "Progressive bacterial algorithm," in *Computational Intelligence and Informatics, IEEE 13th International Symposium on*. IEEE, 2012, pp. 317–322.

[29] C. I. Gonzalez, P. Melin, J. R. Castro, O. Castillo, and O. Mendoza, "Optimization of interval type-2 fuzzy systems for image edge detection," *Applied Soft Computing*, vol. 47, pp. 631–643, 2016.

[30] Y. Maldonado, O. Castillo, and P. Melin, "A multi-objective optimization of type-2 fuzzy control speed in fpgas," *Applied Soft Computing*, vol. 24, pp. 1164–1174, 2014.

[31] M. Antonelli, P. Ducange, and F. Marcelloni, "A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers," *Information Sciences*, vol. 283, pp. 36–54, 2014.

[32] M. B. Gorzałczany and F. Rudziński, "A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability," *Applied Soft Computing*, vol. 40, pp. 206–220, 2016.

[33] C. F. Juang, T. L. Jeng, and Y. C. Chang, "An interpretable fuzzy system learned through online rule generation and multiobjective aco with a mobile robot control application," *IEEE Transactions on Cybernetics*, vol. 46, pp. 2706–2718, 2016.

[34] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 45–65, 2013.

[35] L.-C. Duţu, G. Mauris, and P. Bolon, "A linear-complexity rule base generation method for fuzzy systems," in *IFSA-EUSFLAT 2015-16th World Congress of the International Fuzzy Systems Association (IFSA)-9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)*. Atlantis Press, 2015, pp. 520–527.

[36] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[37] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[38] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems*, vol. 19, p. 153, 2007.

[39] U. Bodenhofer and P. Bauer, "A formal model of interpretability of linguistic variables," in *Interpretability Issues in Fuzzy Modeling*. Springer, 2003, pp. 524–545.

[40] J. M. Alonso, C. Castiello, and C. Mencar, "Interpretability of fuzzy systems: Current research trends and prospects," in *Springer Handbook of Computational Intelligence*. Springer, 2015, pp. 219–237.

[41] S. Guillaume, "Designing fuzzy inference systems from data: an interpretability-oriented review," *Fuzzy Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 426–443, 2001.

[42] S. Galichet and L. Foulloy, "Size reduction in fuzzy rulebases," in *Systems, Man, and Cybernetics, IEEE International Conference on*, vol. 3. IEEE, 1998, pp. 2107–2112.

[43] M. Setnes, R. Babuška, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 3, pp. 376–386, 1998.

[44] C.-T. Chao, Y.-J. Chen, and C.-C. Teng, "Simplification of fuzzy-neural systems using similarity analysis," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 2, pp. 344–354, 1996.

[45] M. Setnes, "Supervised fuzzy clustering for rule extraction," *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 4, pp. 416–424, 2000.

[46] L. T. Koczy and K. Hirota, "Size reduction by interpolation in fuzzy rule bases," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 27, no. 1, pp. 14–25, 1997.

[47] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 29, no. 1, pp. 13–24, 1999.

[48] M. K. Ciliz, "Rule base reduction for knowledge-based fuzzy controllers with application to a vacuum cleaner," *Expert Systems with Applications*, vol. 28, no. 1, pp. 175–184, 2005.

[49] 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," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 6, pp. 1133–1149, 2013.

[50] L.-C. Dutu, G. Mauris, P. Bolon, S. Dabic, and J.-M. Tissot, "A fuzzy rule-based model of vibrotactile perception via an automobile haptic screen," *Instrumentation and Measurement, IEEE Transactions on*, vol. 64, no. 8, pp. 2323–2333, 2015.

[51] T. F. Brooks, D. S. Pope, and M. A. Marcolini, *Airfoil self-noise and prediction*. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Division, 1989, vol. 1218.

Liviu-Cristian Duţu recevied the M.Sc. degree in Electrical Engineering and Computer Science from the Politehnica University of Bucharest, in 2011, and his Ph.D. degree in Computer Science and Information Processing from the University of Grenoble Alpes, Polytech Annecy-Chambéry School of Engineering, in 2015.

He is currently an R&D Engineer with the Machine Learning and Computer Vision team at FotoNation in Bucharest, Romania, and also a benevolent researcher with the LISTIC research lab at the University of Savoie Mont-Blanc in France.

His current research interests include signal and image processing, fuzzy and evolutionary modeling, optimization theory, knowledge discovery and data mining and machine learning.

Gilles Mauris (M'08) graduated from the Ecole Normale Supérieure de Cachan (France) in 1985. He received a Ph.D. degree from the Université de Savoie (France) in 1992.

He is currently Associate Professor of Electrical Engineering at the Savoie Mont-Blanc University with the Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance (LISTIC).

His research interests include fuzzy logic, possibility theory for information processing in the instrumentation and measurement area.

Philippe Bolon (M'94) received the Engineer degree in electrical engineering in 1978 and the Ph.D. degree in signal processing in 1981, both from the National Polytechnic Institute of Grenoble, France.

He received the "Habilitation à Diriger des Recherches" degree from University of Savoie in 1993. From 1980 to 1983, he was Assistant Professor at the National Polytechnic Institute of Grenoble and moved to University of Savoie in 1984, as an Associate Professor. Since 1994, he has been professor at Polytech Annecy-Chambéry, school of engineering of University of Savoie. He is member of LISTIC research lab (Computer Science, Information and Knowledge Processing).

His current research interests include information fusion and image processing. Pr. Bolon is a member of the IEEE Signal Processing Society.