Received: December 17, 2019

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Linguistic Fuzzy Rule Learning through Clustering for Regression Problems

Khalid Bahani[1]* **Mohammed Moujabbir[2]** **Mohammed Ramdani[1]**

[1]*Department of Computer Science, Faculty of Sciences and Technics of Mohammedia, Hassan II University Mohammedia, Morocco*
[2]*Department of Computer Science, Polydisciplinary, Faculty of Khouribga, Sultan Moulay Slimane University, Khouribga Morocco*
* Corresponding author's Email: kbahani1@gmail.com

Abstract: The fuzzy rule-based system model is an important and active research line in the fuzzy logic community looking for compact and robust systems with a high level of accuracy – interpretability of trade-offs. On the other hand, fuzzy rules-based systems provide accurate and interpretable solutions that give the ability to handle Complex data and uncertainty. It has also been historically applied in the solution of classification and regression problems. In this paper, the authors present a new fuzzy approach for solving problems of regression based on linguistic fuzzy rule learning with subtractive clustering and linguistic modifiers. The proposed system includes two phases for getting linguistic fuzzy rules: Multi- granularity, fuzzy discretization of the linguistic variables and linguistic approximation of fuzzy rules learned. Regarding experiments, researchers used twelve real-world data sets to compare the proposed system with three of the most widely used simplified fuzzy genetic systems: $FS^{e}_{MOGFS}+TUN^{e}$, A-METSK-HD^{e} and FRULER. The results highlight the competitiveness of the model in terms of accuracy and its superiority in interpretability.

Keywords: Fuzzy rule-based systems, Fuzzy systems, Regression, Subtractive clustering, Linguistic modifiers.

1. Introduction

The fuzzy Rule-Based System (FRBS) aims to represent the knowledge of human experts in a set of fuzzy IFTHEN rules. These rules are generated through the knowledge of human experts or the use of numerical data and machine learning methods. Many approaches have been proposed for this learning task, such as space partition based methods [1, 2], neural-fuzzy techniques [3], clustering methods [4, 5], genetic algorithms [6, 7]and gradient descent learning methods [8]. The FRBS obtained model belongs to one of two different fuzzy Modeling areas [9]: Linguistic fuzzy modeling and the precise fuzzy modeling. Linguistic fuzzy modeling aims primarily at obtaining an interpretable fuzzy system using linguistic labels with acceptable accuracy. These systems are called Mamdani FRBSs [10]. The reasoning of Mamdani FRBS is based on a set of fuzzy rules, which use linguistic labels both in their antecedents and in their consequents. In the other Fuzzy Modeling area - precise fuzzy modeling - the objective is to obtain Takagi–Sugeno FRBS [11] with good accuracy. This system uses the fuzzy sets to represent the antecedents and a weighted combination of the input variables to represent the consequents. There are two criteria for evaluating FRBSs, accuracy and interpretability. In literature, the root mean square error (RMSE) is defined to be a measure of accuracy. Concerning interpretability, there are two main kinds of approaches [12]: The complexity based interpretability and the semantics-based interpretability. The complexity-based interpretability aims at reducing complexity, it is, usually, measured with number of rules, antecedents and linguistic labels. On the other hand, the semantics-based interpretability is dedicated to maintaining the semantics of membership functions (MFs). This is done by imposing a set of restrictions on coverage, distinguishability and fuzzy ordering, to name a few. Three different criterias for a good interpretability compromise are therefore required: Accuracy, complexity and semantic. These criteria can be improved respectively, but the difficulty is to do that simultaneously. This is because the automatic generation approach usually extracts a large number of rules, especially when the data set is big, which leads to a loss of interpretation in the fuzzy model [13]. This problem is overcome during the optimization process through pruning ineffective rules, by either deleting [14], merging [13], selection [15] or pruning rules [16]. In the literature, one of the most effective methods for improving accuracy, complexity, and semantic is the multiobjective evolutionary algorithms (MOEAs) [17-19]. The MOEAs algorithms allow obtaining varying degrees of accuracy and interpretation in FRBSs. In many works, FRBSs are mainly combined with MOEAs in order to take into account interpretability issues such as PAES in [20]; SPEA2ACC (TSSP2-SI) in [20] and NSGA-II in [21]. At present, FRBSs based on MOEA to solve problems of regression [22] and classification [23] because of the efficiency of this system in overcoming the challenges, such as, obtained simple and accurate models, fast learning and dealing with high number of variables and instances. Three of the most accurate genetic fuzzy systems for regression in the literature are FS^{e}_{MOGFS}+TUN^{e} [9], FRULER [24] and A-METSK-HD^{e} [25]. FRULER is a TSK-1 genetic fuzzy system for regression problems. It has three main sections: The preprocessing stage, the evolutionary learning process and the rule generation unit. The preprocessing stage consists of instance selection and multigranularity fuzzy discretization. FS^{e}_{MOGFS}+TUN^{e} is a multi-objective evolutionary algorithm that learns Mamdani fuzzy rules. This algorithm learns the granularities from uniform multi-granularity fuzzy partitions and slight displacements of fuzzy-partition. It provides a post-processing algorithm to adjust MF parameters and to select rules. In A-METSK-HD^{e} algorithm, the same steps are used to learn accurate Takagi-Sugeno-Kang fuzzy rule-based Systems. Despite the good interpretability-accuracy trade off in FS^{e}_{MOGFS}+TUN^{e} and A-METSK-HD^{e}, the semantics-based interpretability is affected. Indeed, the tuning of the MFs affect the transparency of fuzzy partition (the distinguishability, the coverage, the fuzzy ordering, etc). In this paper, authors present a new method, called FRLC-Regress, to learn linguistic fuzzy rule based on fuzzy clustering for Regression Problems. It is a Mamdani fuzzy system based on fuzzy clustering and linguistics modifiers providing a good balance between accuracy and interpretability.

This work is organized as follows: section 2 deals with FRBS. The section 3 describes FRLC-Rgress model and training algorithms. The experiments are presented in section 4, which discusses the obtained results. Finally, section 5 contains the conclusion.

2. Preliminaries

The FRBS consists of a KB and an inference system module. The inference system module contains an inference engine, fuzzification and defuzzification interfaces. The fuzzification interface transforms crisp data to fuzzy sets, the inference engine uses fuzzy sets with KB to infer through a reasoning method, and the defuzzification interface translates fuzzy rule actions with a defuzzification method into real action. The KB consists of two parts, a database (DB) and a rule base (RB): The RB is a set of fuzzy IF-THEN rules and the DB contains fuzzy sets of linguistic labels. In particular, the DB defines the number of linguistic labels for each linguistic variable and the parameters of their membership functions (MF). Fig 1 shows the FRBS model.

The automatic generation of linguistic FRBS model from data involves the learning of KB components (DB and RB) among other FRBS components. In literature, many approaches have been proposed for learning the DB and RB separately or simultaneously, among possibilities of KB learning process proposed in [6], is the embedded learning process: it is a DB generation process including RB learning. At Each DB has been obtained, the RB generation method is used to derive the rules, and an evaluation stage is applied to validate the obtained KB. The authors use embedded DB learning in FRLC-Regress which represented in Fig 2.

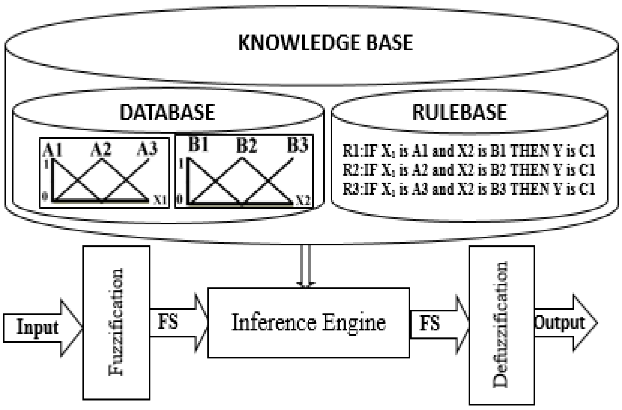

Figure.1 FRBS model

Received: December 17, 2019

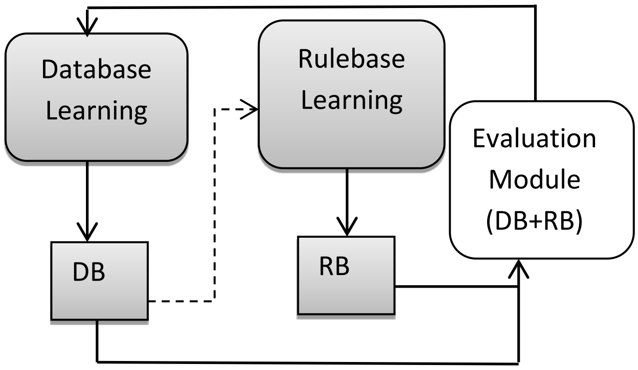

Figure.2 Embedded DB learning

3. FRLC-Rgress model

The FRLC-Rgress model is proposed in [26]. It's an automatic generation of linguistic FRBS model from data in which we integrated an embedded DB learning wrapping RB learning.The FRLC-Rgress model, showing in Fig 3, contains three components: Database learning, Rulebase learning and Evaluation module. The DB learning is based on Multi-granularity fuzzy discretization algorithm to obtain uniform fuzzy partitions with Gaussian MFs. In order to respect the complexity and the semantic constraints of interpretability in [27, 28], the number of MFs in each fuzzy partition must be between 2 and 9. The algorithm researches iteratively the final DB in which each iteration provides an intermediate DB. This DB triggers RB learning, that contains three components: Radius module, Subtractive clustering and Rules module. Radius module calculates the radius r_a using the parameters of Gaussian MFs: mean and standard deviation. r_a is a vector of scalars used in subtractive clustering to extract the clusters. The Rules module is based on these clusters to learn the linguistic fuzzy rules in two steps: The first is the linguistic approximation of the fuzzy rules and the second is the improvement of accuracy in linguistic fuzzy rules with linguistic modifiers. The third component is the Evaluation module in which the KB is evaluated in a MAMDANI fuzzy inference system (FIS) with RMSE, the number of rules and the number of conditions. The MAMDANI FIS uses the t-norm ∧ (minimum) for the logical connective "and" and "*Center of Gravity*" as defuzzification method. The DB learning process stops when the optimal knowledge base is obtained.

3.1 Database learning

DB learning is based on Multi-granularity fuzzy discretization algorithm, in which the authors suppose that the fuzzy partitions are uniform and the Gaussian MFs define the meanings of each linguistic label. In order to select the optimal database, two issues to take into account: the error produced when applying the model to the training data and its complexity. In our case, the error is obtained from RMSE and the complexity is determined with the number of rules (*NBRules*). The objective of multi-granularity fuzzy discretization algorithm is to precise the number of the linguistic labels for each linguistic variable. Formally, consider a collection of N data points $\{x_1, x_2, \ldots, x_N\}$ in an M-dimensional space Let $V=\{v_1, v_2, \ldots, v_M\}$ a set of linguistic variables, $min(v_i)$ and $max(v_i)$ are, respectively, the minimum and maximum values of universe of discourse of v_i, *NbMax* (equal to 9) and *NbMin* (equal to 2) are, respectively, the maximum and the minimum

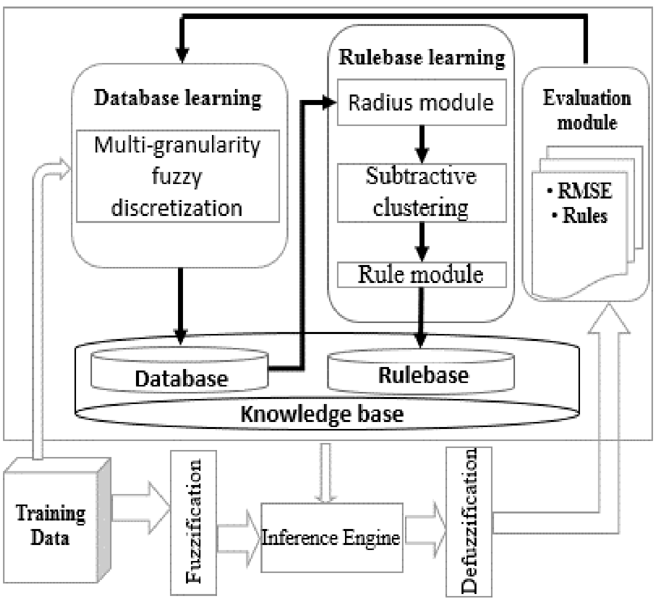

Figure.3 FRLC-Rgress architecture

 DOI: 10.22266/ijies2020.0630.08

numbers of linguistic labels per linguistic variable, and $L=\{(l_1,l_2,...,l_M) / l_i \epsilon \{2,..., NbMax\} \text{ and } i=1,...,M\}$ the set of M–tuples where l_i is the number of linguistic labels of v_i ($l_i \leq NbMax$). L define the search space, it's in order of $(NbMax\text{-}1)^M$. To deals with the complexity of L, the researchers determine the initial DB by searching in $\{(n,....,n) / n=NbMin... NbMax\}$ the optimal M–tuples ($OIDB$=(n_{opt}^1,....,n_{opt}^M)). Afterwards, the algorithm searches iteratively the final DB using *OIDB*: For example, in dimension j, the algorithm searches iteratively the optimum number (*OPTJ*) of linguistic labels (from *NbMin* to *NbMax*) by fixing the other dimensions and replaces n_{opt}^j with *OPTJ*. The algorithm deals with the other dimensions in the same manner. The obtained DBs are an intermediate DBs (IDBs). This process is repeated for each IDB until the final DB has been obtained. The following algorithm provides the main steps for Database learning. From line 1 to 8, the DB learning algorithm implements a loop from the lowest number *NBMin* of linguistic labels to the highest number *NBMax*. The goal of the loop is to determine the optimalM-tuple to find the final DB. Line 2 discretize linguistic variables according to the value of variable I, for example if I = (3; ...; 3), the universe of discourse of each linguistic variable has been divided into two interval for defining three Gaussian membership functions. Line 3 calls the RB learning algorithm to extract the linguistic rules IRB. Line 4 evaluates the obtained Database IDB and Rulebase IRB by calculating the RMSE and the number of rules extracted. Lines 5 and 6 are used to specify n_{opt} the optimum number of linguistic labels. The algorithm makes sure that the RMSE is decreasing and the number of rules does not exceed the threshold limit in line 5 (in this study, the threshold value is less than 100, that is explained in section4). If both conditions are met, the value of n_{opt} is changed to the current value of variable I. From line 9 to the end, the algorithm searches for the final DB using three nested loops: the first loop begins at line 11; its role is to continue searching for the final DB as long as there is a difference between the two variables *tupleLeval* and *tupleOpt*. The second loop begins at line 13, where it aims to change the values of *tupleLeval*. The third loop begins at line 14, where it re-executes lines 2, 3, 4 and 5 using the *tupleLeval* variable. If the previous two conditions are met, at line 19, the algorithm changes the J^{th} value of tupleLeval to i the optimal number of linguistic labels. The search ends when the *tupleLeval* and *tupleOpt* are identical.

```
1)  For each I in {(n... n) / n=NbMin... NbMax}
2)          IDB←Descritize (DB, I)
3)          IRB←RB_Lerning (IDB)
4)          [RMSE, NBRules] ←Evalute (IDB, IRB)
5)          if (RMSE is decrising and NBRules < threshold) then
6)                  n_opt←I_Current_value
7)          end if
8)  end for
9)  tuple_leval←( n_opt, ..., n_opt)
10) tupleOpt←Null
11)  while tupleLeval ≠ tupleOpt
12)         tupleOpt← tupleLeval
13)         For each dimension j
14)                 For each i in {NBMin …NBMax}
15)                         IDB←Descritize (DB, tupleLeval)
16)                         IRB←RB_Lerning (IDB)
17)                         [RMSE, NBRules] ←Evalute (IDB, IRB)
18)                         if (RMSE is decrising and NBRules < threshold) then
19)                                 tupleLeval(j)←i
20)                         end if
21)                 End for
22)         End for
23) End while
```

3.2 Rulebase learning

The RB learning is based on subtractive clustering and linguistic modifiers. The subtractive clustering belongs to the fuzzy clustering which based on the density of data point [29, 30]. Eq. (1) calculates the potential of each data point x_i.

$$P_i = \sum_{j=1}^{N} e^{-\alpha\|x_i - x_j\|^2} \quad (1)$$

Where $\alpha=4/r_a^2$ and r_a is the *cluster radius*, it is an M-dimensional vector of positive scalars which specifies the value of the radius in each dimension. The subtractive clustering algorithm uses a set of initial parameters: The cluster radius r_a, the accept ratio ($\acute{\varepsilon}$= 0.5), the reject ratio (ε= 0.15) and the neighborhood of cluster ($r_b= 1.25*r_a$). As showing in Fig 3, the radius module uses the DB parameters to calculate the radius r_a [26]. In order to illustrate task radius module in j^{th} dimension, let $\{MFun_j^k / k=1... l_j\}$ the set of Gaussian membership functions Obtained by uniform discretization of v_j, the $MFun_j^k$ parameters are: Its mean C_j^k and the standard deviation σ_j^k. The module calculates the j^{th} value r_a^j of r_a with Eq. (2).

$$r_a^j = \frac{\sigma_{jk}\sqrt{8}}{(max(v_j) - min(v_j))} \quad (2)$$

In order to obtain a set linguistic fuzzy rules, the Rule module projects the extracted cluster in all dimensions. Afterwards, the module linguistically approximates the fuzzy rule with Euclidean distance and improves the accuracy with linguistic modifiers (particularly, powered modifiers: *Very*, *Plus*, *Minus*, *More or less*, *slightly*, and *A little*) using Hamming distance [26]. Eq. (3) illustrates the linguistic approximation of the cluster x_i^C:

$$T_i^{\,j} \leftarrow \underset{k=1,\dots,l_j}{argmin} \left(\left|x_{ij}{}^C - C_j{}^k\right|\right) \quad (3)$$

With x_{ij}^C is the j^{th} value of x_i^C and C_j^k the mean of $MFun_j^k$.

Fig 4 showns an example of linguistic approximation of the fuzzy rules with three Linguistic labels and three clusters. The centers of linguistic Labels *L1*, *L2* and *L3* are closet, respectively, to center of clusters *Cl2*, *Cl3* and *Cl1*. Thus, *Label1* repalces *Cluster2* (in rule 2), *Label2* repalces *Cluster3* (in rule 3) and *Label3* repalces *Cluster1* (in rule 1). The generated fuzzy rules require an improvement of accuracy due to the uniform fuzzy partition. The following subsection presents the accuracy improvement of the fuzzy rules with linguistic modifiers.

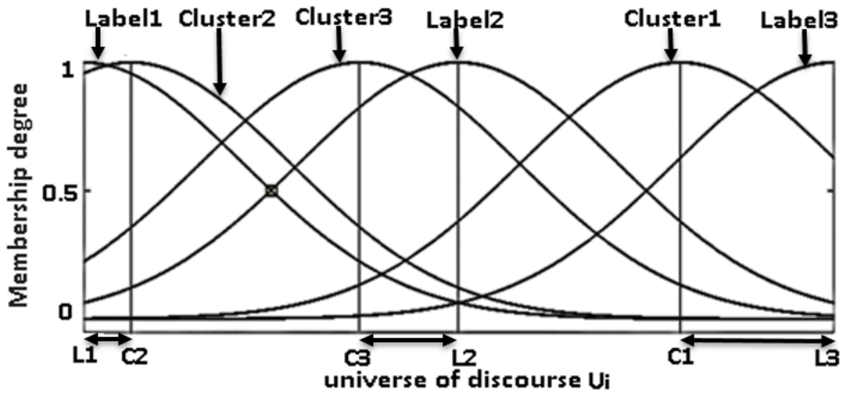

Figure.4 Linguistic approximation with centers

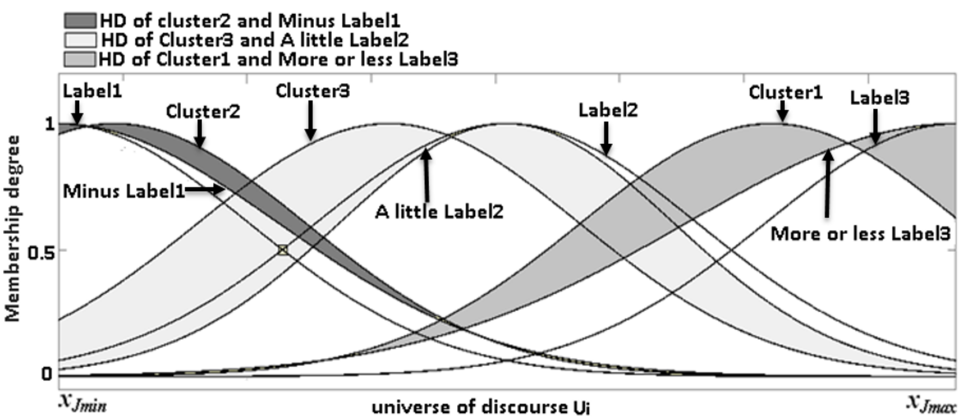

Figure.5 Improving the Accuracy in Linguistic Fuzzy rules

Eq. (4) calculates the Hamming distance between $AFun_i^j$ and all $(MFun_{ij}^C)^P$:

$$D_h = \int_{min(v_j)}^{max(v_j)} |AFun_i^j(x) - (MFun_{ij}^C)^P(x)|\, dx \quad (4)$$

Where P is the parameter of powered modifier and $AFun_i^j$ is the MF of cluster x_i^C in j^{th} dimension. Fig 5 represents the improvement of accuracy in the previous example. *Minus Label1* repalces *Label1* (in rule 2), *a little Label2* repalces *Label2* (in rule 3) and *more or less Label3* repalces *Label3* (in rule 1).

In the obtained RB, each linguistic fuzzy rule includes M conditions.To simplify the RB and take into account the improvement of accuracy simultaneously, we have reduced the number of conditions with *don't care* condition [24]. The following algorithm provides the main steps for rulebase learning, which uses a set of training data (*InputData* and *OutputData*) and contains two main stages to extract linguistic fuzzy rules based on database parameters. The first stage starts from line 2 and ends at line 4, this stage contains the first loop in which the vector r_a is calculated. Line 3 calculates the j^{th} value of the vector r_a by the function *Influence_Range* and the parameters of the j^{th} linguistic variable v_j, according to Eq. (2). In the second stage, the algorithm uses the vector r_a to extract the linguistic fuzzy rules through three nested loops. The second stage begins at line 5, the subtractive clustering function *Sub_Clr* uses the vector r_a to extract clusters and saved them in the variable *Cset*. The first loop begins at line 6, where the objective is to deal with all extracted clusters. On line 7, the second loop begins with a linguistic approximation of the current cluster x_i^C, where line 8 approximates the membership functions of the cluster x_i^C to those of v_j. Now, it is possible to use the term "linguistic fuzzy rules" after projecting clusters on all dimensions and applying the linguistic approximation. The stage for improving accuracy in linguistic fuzzy rules starts from line 9. This stage includes the third loop, from line 10 to line 15, the linguistic modifiers m are applied to the membership functions of the current rules, this is for calculating the Hamming distance. After that, the linguistic modifiers that achieves the smallest value of the Hamming distance is chosen. Finally, line 16 adds the linguistic fuzzy rule to the rulebase.

Rulebase learning algorithm

1) Data= {InputData, OutputData}
2) FOR EACH V_j in V
3) $r_a^j \leftarrow$ Influence_Range(V_j)
4) END FOR
5) Cset $\leftarrow$ Sub_Clr (InputData, OutputData , r_a)
6) FOR each Cluster x_i^C in Cset
7) FOR EACH V_j in V
8) $T_i^j \leftarrow \underset{k=1,\ldots,l_j}{\text{argmin}} \left(|x_{ij}^C - C_j^k|\right)$
9) min_area $\leftarrow \int_{min(v_j)}^{max(v_j)} MFun_{ij}^C(x)$
10) FOR each linguistic_modifier m
11) Hamming_distance $\leftarrow \int_{min(v_j)}^{max(v_j)} |AFun_i^j(x) - (MFun_{ij}^C)^P(x)|\, dx$
12) IF Hamming_distance < min_distance THEN
13) $m^C \leftarrow$ m
14) min_area $\leftarrow$ Hamming_distance
15) END FOR
16) ADD Linguistic_Term With (m^C and T_i^j)
17) END FOR
18) END FOR

4. Experiments development and obtained results

In order to analyse the performance of FRLC-Rgress, authors have used the KEEL project repository's twelve real-world regression problems [9]. Table 1 displays the dataset properties, with 337 to 20640 examples of instances and 2 to 16 input variables. The FRLC-Rgress results are compared with three fuzzy rule-based systems for regression problems: $FS^e_{MOGFS}+TUN^e$ [9], A-METSK-HD^e [25] and FRULER [24]. Researchers have adopted a cross validation model in all experiments, which involves a random division of the dataset into 5 folds, each of which contains 20% of the dataset patterns and four folds used to train and one to test. In order to reduce search space, FRLC-Rgress do not comply with situations in which there are obviously more than 100 clusters. In the other hand to handle the scalability problem in datasets, authors consider small percentage (SP) of the training data to estimate the RMSE. Experimentally, SP equal to 2000 is a good choice. The selection of the training data is based on the potential of each data point. The potentials are calculated using the optimal M –tuples OIDB presented in the previous section, then sorting the data based on the potentials and selecting those separated with *N/SP* (*N*=length of training data). This selection reduces data density and ensures an optimal representation. The Final KB is presented in Table 2, in DB, the values shown in input and output columns represent the number of Gaussian MFs in each dimension. In RB, the columns R and C shown the numbers of rules and conditions per rule respectively.

Table 1. Data sets considered in experiments

Problem	Abbr	Variables	Cases
Electrical Length	ELE1	2	495
Quake	QUA	3	2178
Friedman	FRIE	5	1200
AutoMPG6	MPG6	5	398
Daily electricity energy	DEE	6	365
Delta elevators	DELELV	6	9517
AutoMPG8	MPG8	7	398
Stock	STP	9	950
Weather Ankara	WAN	9	1609
Forest Fires	FOR	12	517
Baseball	BAS	16	337
California Housing	CAL	8	20,640

Table 2. KB of FRLC-Rgress

Problems	Database		Rulebase	
	Inputs	O	R	C
ELE1	6 7	8	7.8	1.9
QUA	4 8 2	8	11	2.3
FRIE	6 5 2 5 4	8	33.8	4.0
MPG6	7 8 8 4 5	9	9.6	3.3
DEE	4 5 5 4 3 3	8	13.2	4.0
DELELV	7 6 5 3 9 5	8	5.4	4.7
MPG8	4 4 3 9 2 4 2	8	11.6	4.2
STP	8 8 7 7 9 7 6 3 2	8	27.6	5.4
WAN	8 5 7 7 2 2 2 3 2	9	25.6	4.9
FOR	2 2 3 2 2 3 2 2 7 2 2 2	9	1.6	1.00
BAS	5 6 7 6 3 3 2 9 7 4 2 2 9 6 9 3	9	12.2	10.3
CAL	8 8 3 7 5 6 7 7	7	15.4	4.5

Table 3. Average number of rules (R) and RMSE (Tst.). Results in this table (Tst.) should be multiplied by 10^{-6} and 10^{5} for DELELV and BAS respectively

Datasets	FRULER		A-METSK-HD^{e}		FS$^{e}_{MOGFS}$+TUN^{e}		FRLC-Rgress	
	R	(Tst.)	R	(Tst.)	R	(Tst.)	R	(Tst.)
ELE1	**4.1**	2.012	11.4	2.022	8.1	1.954	7.8	**1.911**
QUA	7.8	0.0181	18.3	0.0181	**3.2**	0.0178	11	**0.0177**
FRIE	**8**	**0.731**	66	1.888	22	3.138	33.8	3.132
MPG6	13.7	**3.727**	53.6	4.478	20	4.562	**9.6**	4.452
DEE	**7.9**	0.080	50.6	0.103	18.3	0.093	13.2	**0.077**
DELELV	5.8	1.045	39.1	**1.031**	7.9	1.086	**5.4**	1.300
MPG8	12.7	**4.084**	64.2	5.391	23	4.747	**11.6**	4.675
STP	42.4	**0.353**	66.4	0.387	**23**	0.912	27.6	0.720
WAN	**5.6**	**0.888**	48	1.189	8	1.635	25.6	1.564
FOR	5.6	**2214**	40.6	5587	10	2628	**1.6**	2467
BAS	**6.2**	3.0577	59.8	3.6882	17	2.6132	12.2	**2.557**
CAL	15.4	2.11	55.8	**1.71**	**8.4**	2.95	15.4	2.72

Table 3 shows the average results of FRLC-Rgress, FS^{e}_{MOGFS}+TUN^{e}, A-METSK-HD^{e} and FRULER. For each algorithm and dataset, two different results are shown: The columns (*Tst.)* and *(R). Tst* present the average RMSE in testing data and *R* present the average number of rules. The values with the best results are marked in bold. It can be seen that the number of rules of FRLC-Rgress and FRULER is the lowest in all datasets (equal to 5) followed by FS^{e}_{MOGFS}+TUN^{e}. In the case of accuracy, FRULER achieves the best results in six problems followed by FRLC-Rgress in four problems. In order to analyze the statistical significance of these results, authors have used KEEL software tool [31] to apply the statistical tests: Friedman test [32] was used for the test error and the number of rules in order to get a ranking of the algorithms and to see whether the difference between them was statistically significant. As showing in Table 4, FRULER algorithm gets the top ranking, i.e., it has the best results in accuracy among all the algorithms. In order to compare whether the difference between FRULER and other ranked algorithms was significant, we performed a Holm's test [33]. As depicted in Table 5, Holm's test indicates that FRULER is not statistically superior to either A-METSK-HD e or FRLC-Rgress. To compare directly FRLC-Rgress with other algorithms, authors have used the Wilcoxon signed-rank test [34]. Table 6 indicates that FRLC-Rgress is statistically better than FS^{e}_{MOGFS}+TUN^{e} while it is statistically equivalent to A-METSK-HD^{e}.

To analyze the complexity of each algorithm, the same Friedman test was performed to the number of rules in Table 3 (Table 7). FRULER has the lowest ranking. The next algorithms in the ranking are FRLC-Rgress and FS^{e}_{MOGFS}+TUN^{e}, followed by the A-METSK-HD^{e} approaches with a big difference in the ranking. Researchers have used a Holm test (Table 8) in order to assess whether the difference in complexity among the most accurate proposals was significant. The hypothesis of equality with FS^{e}_{MOGFS}+TUN^{e} and FRLC-Rgress is accepted with p-value> 0.05. To compare directly FRLC-Rgress with the other competing algorithms, the Wilcoxon test is given in Table 9. the test indicates that FRLC-Rgress is statistically better than A-METSK-HD^{e}; it is also statistically equivalent to FS^{e}_{MOGFS}+TUN^{e}. Researchers note that the complexity of rule-based systems has other parameters, such as the number of features and linguistic labels used per rule, but only the number of rules is used here. Concerning interpretability based semantic, only FRLC-Rgress provides a uniform Database for a large variety of problems. while all obtained DBs by FS^{e}_{MOGFS}+TUN^{e} violates distinguishability and complementarity constraints. This is illustrated in Fig 6: That shows the obtained fuzzy patition of BAS output with FS^{e}_{MOGFS}+TUN^{e} and FRLC-Rgress. It is clear that obtained fuzzy partition with FS^{e}_{MOGFS}+TUN^{e} violates complementarity constraint. This study showed the effectiveness of FRLC-Rgress in the resolution of regression problems, and at the level of accuracy, FRLC-Rgress was able to bypass mamdani systems and rivalry TSK systems of the first degree and outperform them at the level of interpretability. FRLC-Rgress has also provided distinguished solutions in interpretability based semantic.

Table 4. Friedman test ranking results for the test error in Table 3

Algorithm	Ranking
FRULER	1.84
FRLC-Rgress	2.17
A-METSK-HD^{e}	2.84
FS^{e}_{MOGFS}+TUN^{e}	3.17

Table 5. Posthoc test with $\alpha = 0.05$ for accuracy

Control Algorithm: FRULER					
i	**Algorithm**	**z-value**	**P-value**	**α/i**	**Hypothesis**
3	FS^{e}_{MOGFS}+TUN^{e}	2.5298	0.0114	0.0166	Rejected
2	A-METSK-HD^{e}	1.8973	0.0577	0.025	Accepted
1	FRLC-Rgress	0.6324	0.5270	0.05	Accepted

Table 6. Wilcoxon test between FRLC-Rgress and the other competing algorithms for the accuracy results

Comparison	P-value
FRLC-Rgress vs FS^{e}_{MOGFS}+TUN^{e}	0.0116
FRLC-Rgress vs A-METSK-HD^{e}	0.34827

Table 7. Friedman test ranking results for the number of rules in Table 3

Algorithm	Ranking
FRULER	1.70
FRLC-Rgress	1.95
FS^{e}_{MOGFS}+TUN^{e}	2.33
A-METSK-HD^{e}	4

Table 8. Posthoc test with $\alpha = 0.05$ for complexity

Control Algorithm: FRULER					
i	**Algorithm**	**z-value**	**P-value**	**α/i**	**Hypothesis**
3	A-METSK-HD^{e}	4.3481	1.4E-5	0.0166	Rejected
2	FS^{e}_{MOGFS}+TUN^{e}	1.1858	0.23568	0.025	Accepted
1	FRLC-Rgress	0.4743	0.63525	0.05	Accepted

Table 9. Wilcoxon test between FRLC-Rgress and the other competing algorithms for the complexity results

Comparison	P-value
FRLC-Rgress vs A-METSK-HD^{e}	0.0011
FRLC-Rgress vs FS^{e}_{MOGFS}+TUN^{e}	0.5000

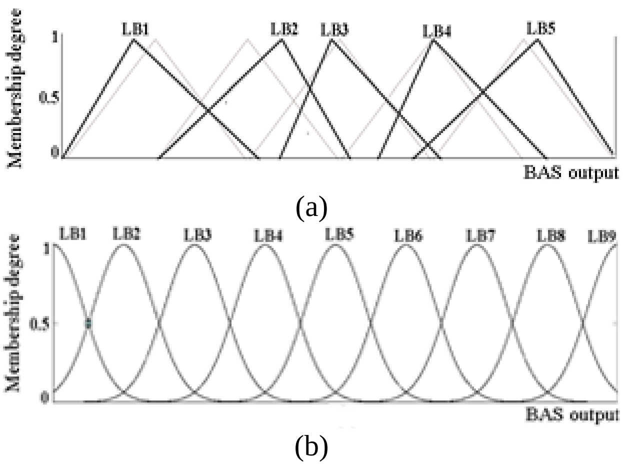

Figure.6 Fuzzy partition of BAS output with: (a) $FS^{e}_{MOGFS}+TUN^{e}$ and (b) FRLC-Rgress

5. Conclusion

In this paper, a new fuzzy rule based system called FRLC-Rgress is presented. It learns MAMDANI fuzzy rule for regression problems based on subtractive clustering and linguistic modifiers. FRLC-Rgress has been compared to the most accurate Genetic Fuzzy Systems for twelve datasets. The results have shown the potentialities of the proposed approach with respect to the state of the art in the Fuzzy Rules Based Systems area.

Future works look forward to optimizing the knowledge base of FRLC-Rgress and dealing with high-dimensional regression problems.

References

[1] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems", *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, pp. 506–515, 2001.

[2] H. Liu and M. Cocea, "Granular computing-based approach of rule learning for binary classification", *Granular Computing*, Vol. 4, No. 2, pp. 275–283, 2019.

[3] J. Kim and N. Kasabov, "Hyfis: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems", *Neural Networks*, Vol. 12, No. 9, pp. 1301–1319, 1999.

[4] N. K. Kasabov and Q. Song, "Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction", *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 2, pp. 144–154, 2002.

[5] M. Prasad, K.-P. Chou, A. Saxena, O. P. Kawrtiya, D.-L. Li, and C.-T. Lin, "Collaborative fuzzy rule learning for mamdani type fuzzy inference system with mapping of cluster centers", In: *Proc. of 2014 IEEE Symposium on Computational Intelligence in Control and Automation*, pp. 1–6, 2014.

[6] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects", *Evolutionary Intelligence*, Vol. 1, No. 1, pp. 27–46, 2008.

[7] B. Ratner, *Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data*. Chapman and Hall/CRC, 2017.

[8] H. Ishibuchi, K. Nozaki, H. Tanaka, Y. Hosaka, and M. Matsuda, "Empirical study on learning in fuzzy systems by rice taste analysis," *Fuzzy Sets and Systems*, Vol. 64, No. 2, pp. 129–144, 1994.

[9] R. Alcalá, M. J. Gacto, and F. Herrera, "A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems", *IEEE Transactions on Fuzzy Systems*, Vol. 19, No. 4, pp. 666–681, 2011.

[10] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis", In: *Proc. of the sixth international symposium on Multiple-valued logic*, pp. 196–202, 1976.

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

[12] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures", *Information Sciences*, Vol. 181, No. 20, pp. 4340–4360, 2011.

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

[14] M. Patel, "Various rule pruning techniques and accuracy measures for fuzzy rules", *International Journal of Application or Innovation in Engineering & Management*, Vol. 2, No. 12, pp. 175–178, 2013.

[15] 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.

[16] A. Batbarai and D. Naidu, "Approach for rule pruning in association rule mining for removing redundancy", *Int. J. Innov. Res. Comput. Commun. Eng*, Vol. 2, No. 5, pp. 4207–4213, 2014.

[17] 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", *IEEE Transactions on Fuzzy Systems*, Vol. 21, No. 1, pp. 45–65, 2013.

[18] M. Antonelli, P. Ducange, and F. Marcelloni, "An efficient multi-objective evolutionary fuzzy system for regression problems", *International Journal of Approximate Reasoning*, Vol. 54, No. 9, pp. 1434–1451, 2013.

[19] A. Fernandez, V. Lopez, M. J. del Jesus, and F. Herrera, "Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges", *Knowledge-Based Systems*, Vol. 80, pp. 109–121, 2015.

[20] M. J. Gacto, R. Alcalá, and F. Herrera, "Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems", *Soft Computing*, Vol. 13, No. 5, pp. 419–436, 2009.

[21] P. Pulkkinen and H. Koivisto, "A dynamically constrained multiobjective genetic fuzzy system for regression problems", *IEEE Transactions on Fuzzy Systems*, Vol. 18, No. 1, pp. 161–177, 2010.

[22] I. Rodriguez-Fdez, M. Mucientes, and A. Bugarin, "S-fruler: Scalable fuzzy rule learning through evolution for regression", *Knowledge-Based Systems*, Vol. 110, pp. 255–266, 2016.

[23] N. L. Tsakiridis, J. B. Theocharis, R. Pérez, A. González, and G. C. Zalidis, "Extensions of the deco 3 r algorithm for generating compact and cooperating fuzzy rule-based classification systems", In: *Proc. of 2016 IEEE International Conference on Fuzzy Systems*, pp. 1429–1436, 2016.

[24] I. Rodrguez-Fdez, M. Mucientes, and A. Bugarn, "Fruler: Fuzzy rule learning through evolution for regression", *Ínformation Sciences*, Vol. 354, pp. 1–18, 2016.

[25] M. J. Gacto, M. Galende, R. Alcalá, and F. Herrera, "Metsk-hd e: A multiobjective evolutionary algorithm to learn accurate tsk-fuzzy systems in high-dimensional and large-scale regression problems", *Information Sciences*, Vol. 276, pp. 63–79, 2014.

[26] K. Bahani, M. Moujabbir, and M. Ramdani, "Fuzzy rule learning with linguistic modifiers", In: *Proc. of the 12th International Conference on Intelligent Systems: Theories and Applications*, p. 33, 2018.

[27] J. V. de Oliveira, "Semantic constraints for membership function optimization", *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 29, No. 1, pp. 128–138, 1999.

[28] C. Mencar and A. M. Fanelli, "Interpretability constraints for fuzzy information granulation", *Information Sciences*, Vol. 178, No. 24, pp. 4585–4618, 2008.

[29] S. L. Chiu, "Fuzzy model identification based on cluster estimation", *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, pp. 267–278, 1994.

[30] S. Chiu, *Extracting fuzzy rules from data for function approximation and pattern classification*, Chapter 9 in Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications, ed. D. Dubois, H. Prade, and R. Yager, John Wiley, 1997.

[31] I. Triguero, S. González, J. M. Moyano, S. Garcá, J. Alcalá-Fdez, J. Luengo, A. Fernández, M. J. del Jesús, L. Sánchez, and F. Herrera, "Keel 3.0: an open source software for multi-stage analysis in data mining", *International Journal of Computational Intelligence Systems*, Vol. 10, No. 1, pp. 1238–1249, 2017.

[32] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance", *Journal of the American Statistical Association*, Vol. 32, No. 200, pp. 675–701, 1937.

[33] S. Holm, "A simple sequentially rejective multiple test procedure", *Scandinavian Journal of Statistics*, pp. 65–70, 1979.

[34] F. Wilcoxon, "Individual comparisons by ranking methods", *Breakthroughs in statistics*, pp. 196–202, 1992.