Improving Michigan-style fuzzy-rule base classification generation using a Choquet-like Copula-based aggregation function

Edward Hinojosa-Cardenas[1], Edgar Sarmiento-Calisaya[1], Heloisa A. Camargo[2] and Jose Antonio Sanz[3]

[1] *Universidad Nacional de San Agustin de Arequipa, Arequipa, Peru*
[2] *Federal University of São Carlos, São Carlos, São Paulo, Brazil*
[3] *Universidad Publica de Navarra, Pamplona, Spain*

Abstract

This paper presents a modification of a Michigan-style fuzzy rule based classifier by applying the Choquet-like Copula-based aggregation function, which is based on the minimum t-norm and satisfies all the conditions required for an aggregation function. The proposed new version of the algorithm aims at improving the accuracy in comparison to the original algorithm and involves two main modifications: replacing the fuzzy reasoning method of the winning rule by the one based on Choquet-like Copula-based aggregation function and changing the calculus of the fitness of each fuzzy rule. The modification proposed, as well as the original algorithm, uses a (1+1) evolutionary strategy for learning the fuzzy rule base and it shows promising results in terms of accuracy, compared to the original algorithm, over ten classification datasets with different sizes and different numbers of variables and classes.

Keywords

Michigan-style algorithm, fuzzy rule-based classification systems, Choquet-like Copula-based aggregation function, evolutionary strategy

1. Introduction

We face classification problems in a wide range of real-world problems and research areas. For example, cancer classification [1], text classification [2], emotion classification [3], so on.

Many researchers have proposed machine learning-based techniques to solve the classification task, for instance, decisions trees [4], neural networks [5], deep learning [6] and Fuzzy Rule-Based Classification Systems (FRBCSs) [7].

FRBCSs, a type of Fuzzy Rule-based Systems (FRBSs), have demonstrated to be an effective technique to tackle classification problems [8]. Additionally, a FRBCS contains fuzzy rules (*if-then*) with linguistic labels (represented by fuzzy sets) that model natural language and have high interpretability, offering the possibility to understand in detail how the system works [9].

Evolutionary Computation is studied since the beginning of the 1990s to automatic learn or tune all components of FRBSs and FRBCSs. This hybridization is named Evolutionary Fuzzy

WILF'21: The 13th International Workshop on Fuzzy Logic and Applications, Dec. 20–22, 2021, Vietri sul Mare, Italy
ehinojosa@unsa.edu.pe (E. Hinojosa-Cardenas); esarmientoca@unsa.edu.pe (E. Sarmiento-Calisaya); heloisa@dc.ufscar.br (H. A. Camargo); joseantonio.sanz@unavarra.es (J. A. Sanz)
0000-0003-0307-7567 (E. Hinojosa-Cardenas); 0000-0002-0956-3091 (E. Sarmiento-Calisaya); 0000-0002-5489-7306 (H. A. Camargo); 0000-0002-1427-9909 (J. A. Sanz)

Systems (EFSs) [10]. The proposed algorithm in this paper is an EFS to automatic learn the Rule Base (RB), that is, the set of fuzzy rules composing the FRBCS.

There are four main approaches to learn a RB in EFS: Pittsburgh [11], Michigan [12], iterative rule learning [13] and genetic cooperative competitive learning [14]. This paper is focused on the Michigan approach where an individual represents a single fuzzy rule, thus, a RB is defined by all individuals of the population in the evolutionary optimization process.

Another important component of a FRBCS is the Fuzzy Reasoning Method (FRM), which is responsible for performing the classification of new examples based on the RB and the Data Base (DB - that specifies the definitions of fuzzy sets for the variables).

The classification of new instances made by the FRM is based on an aggregation function. Several FRBCSs use the Winning Rule (WR) as aggregation function, where, the class assigned to a new instance is determined by the fired fuzzy rule with the maximum compatibility with the new instance, i.e., the WR uses the maximum aggregation function, considered as averaging. Thus, the information provided by the other fired fuzzy rules are ignored.

To avoid this problem, other FRBCSs use additive combination as aggregation function in the FRM, where, the information of all fired fuzzy rules for each class is taken into account in the aggregation step to determine the class for a new instance. This aggregation operator is considered as non-averaging.

In order to mix the characteristic of two aggregation functions mentioned above and to improve the results of FRMs, Barrenechea et al. [15] introduced the usage of an averaging operator named Choquet integral. Different generalizations of the Choquet integral are proposed in [16] [17]. The use and extension of the Choquet Integral-based operators applied to improve the performance FRBCs is a field of current interest for researchers and a future research direction for the aggregations operators [18].

The main contribution of this paper is to use the generalization called Choquet-like Copula-based aggregation function (CC-integral) [16] in a Michigan-style fuzzy rule generation algorithm [19] to improve the classification rate of the learned fuzzy rules.

The remainder of this paper is outlined as follows: Section 2 presents the concept of FRBCSs and details of each stage in a FRM. In section 3 the proposed algorithm are detailed step by step. Then, Section 4 shows the experimental results to evaluate the accuracy of the proposed algorithm. Finally, in Section 5, the conclusions are drawn and some future works are proposed.

2. Fuzzy Rule-Based Classification Systems

Any classification problem considers a set of examples $E = \{e_1, e_2, \ldots, e_p\}$ and a set of classes $Class = \{Class_1, Class_2, \ldots, Class_m\}$, and the objective is to assign a class $Class_j \in Class$ to each example $e_b \in E$. Each e_b is defined by a set of features $e_b = \{e_{b1}, e_{b2}, \ldots, e_{bn}\}$ and each feature is defined by a linguistic term a.

In a FRBCS, the FRM uses a set of L fuzzy rules in the RB and fuzzy sets in the DB to assign a class to each example. Usually, the fuzzy rules follow the format:

R_i: **IF** e_{b1} **IS** a_{i1} **AND** e_{b2} **IS** a_{i2} **AND** ... **AND** e_{bn} **IS** a_{in} **THEN** Class = $Class_{j_i}$ **WITH** RW_i

The FRM follow four stages [20]:

1. Matching degree (M): For each fuzzy rule i in the RB, the if-part is compared with the example to be classified e_b using a t-norm (T) as conjunction operator for all membership

degrees (μ) obtained.

$$M_i(e_b) = T(\mu_{a_{i1}}(e_{b1}), \ldots, \mu_{a_{in}}(e_{bn})) \tag{1}$$

2. Association degree (A): For each fuzzy rule i in the RB, M_i is weighted by its rule weight according the $Class_{j_i}$

$$A_i^{Class_{j_i}}(e_b) = M_i(e_b) \cdot RW_i \tag{2}$$

3. Example classification soundness degree for all classes (S): At this point, for each class, $Class_j$, the positive information, $A_i^{Class_{j_i}}(e_b) > 0$, given by the fired fuzzy rules of the previous step, is aggregated by an aggregation function $\mathbb{A}$.

$$S_{Class_{j_i}}(e_b) = \mathbb{A}\left(A_1^{Class_{j_i}}(e_b), \ldots, A_L^{Classs_{j_i}}(e_b)\right) \tag{3}$$

The key point in the FRM is how the information given by the fired fuzzy rules is aggregated. Following, three different aggregation functions are presented:

a) Winning Rule (WR): For each class, it only considers the rule having the maximum compatibility with the example.

$$S_{Class_{j_i}}(e_b) = max\{A_i^{Class_{j_i}}(e_b)\} \tag{4}$$

b) Additive Combination (AC): It aggregates all the fired rules, for each class $Class_j$, by using the normalized sum.

$$S_{Class_{j_i}}(e_b) = \frac{\sum_{i=1}^{L} A_i^{Class_{j_i}}(e_b)}{max_{j=1,\ldots,m} \sum_{i=1}^{L} A_i^{Class_{j_i}}(e_b)} \tag{5}$$

c) Choquet-like Copula-based aggregation function (CC-integral): It is an aggregation function supported by solid theory, proposed and detailed in [16].

$$S_{Class_{j_i}}(e_b) = \mathfrak{C}_{\mathfrak{m}_j}^{C}\left(A_1^{Class_{j_i}}(e_b), \ldots, A_L^{Class_{j_i}}(e_b)\right) \tag{6}$$

The $\mathfrak{C}_{\mathfrak{m}_j}^{C}$ is the constructed CC-integral for the copula C: $[0,1]^2 \to [0.1]$ and fuzzy measure $\mathfrak{m}_j$:

$$\mathfrak{C}_{\mathfrak{m}_j}^{C}(\vec{x}) = \sum_{i=1}^{L}\left(min\{x_{(i)}, \mathfrak{m}_j(A_{(i)})\} - min\{x_{(i-1)}, \mathfrak{m}_j(A_{(i)})\}\right) \tag{7}$$

$$\mathfrak{m}_j(X) = \left(\frac{|X|}{n}\right)^{q_j}, with\ q_j > 0 \tag{8}$$

where $\vec{x} = A_1^{Class_{j_i}}(e_b), \ldots, A_L^{Class_{j_i}}(e_b)$ and $X \subseteq N$. In the proposed algorithm, the CC integral is constructed using the minimum and the cardinality as the copula (C) and fuzzy measure (m_j), respectively.

4. Classification: The final decision is made in this step. To do so, a function $F : [0,1] \to \{1, \ldots, m\}$ is applied over the results obtained by example classification soundness degrees of all classes:

$$F\left(S_{Class_j}, \ldots, S_{Class_j}\right) = max_{j=1,\ldots,m}\left(S_{Class_j}\right) \tag{9}$$

An example of the behavior of three aggregation functions mentioned above is presented in [16].

3. Proposed Algorithm

The proposed modified algorithm in this paper is based on the algorithm proposed in [19] (it is called in this paper as Michigan_EE). Basically, we propose an algorithm that modifies the calculation of fitness of each fuzzy rule, the calculation of the classification rate of the RB based on the CC-integral aggregation function (explained in the previous section) and the evolutional optimization of the values of the exponents q. We call the proposed algorithm Michigan_EE_CC and it is detailed in Algorithm 1.

Algorithm 1 Proposed Algorithm Michigan_EE_CC

Output: P_{best} and q^{best}

1: Create the DB
2: Generate N_{rule} fuzzy rules by the MPB to make an initial population P_0
3: Calculate the fitness of each rule R_i in P_0
4: Generate an encoded individual q^0 and $q^{best} = q^0$
5: Calculate the Classification Rate (CR) by P_0 using q^{best} and $CR_{best} = CR$
6: **for** $t = 1$ to TQ **do**
7:     Generate a randomly q^t
8:     Calculate the CR by P_0 using q^t
9:     **if** ($CR > CR_{best}$) **then**
10:         $CR_{best} = CR$; $q^{best} = q^t$
11:     **end if**
12: **end for**
13: $P_{best} = P_0$
14: **for** $iter = 0$ to $Iter$ **do**
15:     $P_{iter} = P_{best}$
16:     Generate $N_{replace}/2$ fuzzy rules by genetic operations in P_{iter} and $N_{replace}/2$ fuzzy rules by the MPB
17:     Replace the worst $N_{replace}$ fuzzy rules in P_{iter} with the newly generated $N_{replace}$ fuzzy rules to make a new population P_{iter}
18:     Calculate the fitness of each rule R_i in P_{iter}
19:     Calculate the CR by P_{iter} using q^{best}
20:     **for** $t = 1$ to TQ **do**
21:         Generate a randomly q^t
22:         Calculate the CR by P_{iter} using q^t
23:         **if** ($CR > CR_{best}$) **then**
24:             $CR_{best} = CR$; $q^{best} = q^t$; $P_{best} = P_{iter}$
25:         **end if**
26:     **end for**
27: **end for**

In line 1, for each attribute, the minimum and maximum value are obtained. After, nFS triangular fuzzy sets are defined uniformly distributed on the attribute domain, i.e. each fuzzy set has the same support and they cover all the range between maximum and minimum values.

In line 2, N_{rule} fuzzy rules are generated based on the format mentioned in Section 2, and inserted into the population P_0. Each fuzzy rule is encoded as a chromosome with three parts: the first part represents the antecedent part where each gene represents an index of a fuzzy set (or linguistic term) for each attribute (value zero represents a *don't care* condition, what means that the respective attribute does not appear in the rule). The second part (only one gene) represents the class or consequent of the fuzzy rule. Finally, the third part (only one gene) represents the rule weight. Figure 1 illustrates the representation used in this step.

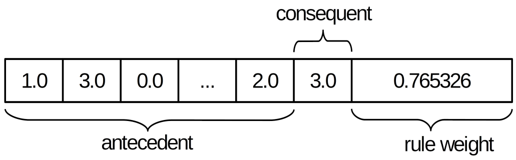

Figure 1: Encoding a Fuzzy Rule

Each fuzzy rule in P_0 is generated by Multi-Pattern-Based rule generation (MPB), where, for a single rule generation, one base example and some $(H-1)$ support examples are randomly selected from the training data of the same class as the base example. More details of MPB can be found in [19].

One of the proposed modification is performed in line 3. In Michigan_EE algorithm, the fitness of each fuzzy rule is calculated by the number of correctly classified training patterns by the fuzzy rule, which is more appropriate for using WR. In the proposed algorithm, that uses CC-integral, the fitness of each rule R_i (with a class C_{R_i}) is calculated by the difference between: the average degree of association (> 0) of all examples with a class $C+$, where $C_{R_i} = C+$, and twice the average degree of association (> 0) of all examples with a class $C-$, where $C_{R_i} \neq C-$. That difference refers to the idea of obtaining rules covering the maximum number of examples (completeness degree) with the minimum number of negative examples (consistency degree) proposed in [21]. The next equation shows that difference:

$$fitness(R_i) = \frac{\sum_{b=1}^{p} A_i^{Class_{j_i}}(e_b)+}{|A_i^{Class_{j_i}}(e_b)+|} - 2 \times \frac{\sum_{b=1}^{p} A_i^{Class_{j_i}}(e_b)-}{|A_i^{Class_{j_i}}(e_b)-|} \tag{10}$$

In line 4, an encoded individual q^0 is generated, which contains the values of the exponents for each class used in the CC-integral aggregation method (see Secction 2-3-c) and it is stored as q^{best}. The value of 1.00 is assigned to each exponent so that the classical cardinality measure is represented. Figure 2 illustrates the representation used in these steps.

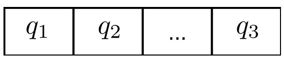

Figure 2: Encoding the values of the exponents used in CC-integral aggregation function

Another modification in the proposed algorithm is performed in line 5. In Michigan_EE algorithm the classification rate (CR) of all the fuzzy rules in the population or RB is calculated based on a FRM with WR aggregation function for each training example. In the proposed algorithm, the CR of the RB is based on a FRM with CC-integral aggregation function for each training example, using q^{best}. After that, CR is stored as CR_{best}.

In lines 6-12, new better values for each exponent are searched with a small value of TQ because the calculation of the CR is computationally expensive. In line 7, a new q^t is randomly generated. The values of the exponents q_j are generated in the range $[0.01, 1.99]$. However, according to [15], the suggested final values of the exponents are in the range $[0.01, 100]$, for that, the values used in the calculation of CR are adapted as:

$$q_j = \begin{cases} q_j & if\ 0.00 < q_j \leq 1.00 \\ \frac{1}{2-q_j} & if\ 1.00 < q_j < 2.00 \end{cases} \tag{11}$$

In line 8, q^t is used in the calculation of CR using P_0. After that, the new q^t is stored as q^{best} and CR is stored as CR_{best} if the CR is better than CR_{best}.

In lines 14-27, the Michigan approach is performed to learn evolutionarily the RB. In each iteration the worst $N_{replace}$ ($= N_{rule}/2$) fuzzy rules in the population P_{iter} (copy of $Pbest$ or population with the best CR) are replaced by fuzzy rules genetically created or MPB, in order to found a better population (or a population with better CR).

In line 16, for generating the first $N_{replace}/2$ fuzzy rules are used a parent selection operator, a crossover operator (with $crossProb$ probability) and a mutation operator (with $mutaProb$ probability and a random replacement of each membership function with $mutaProbMF$ probability) based on population P_{iter}. For the remaining $N_{replace}/2$ fuzzy rules, the MPB is performed, where, for a single rule generation, the base example is randomly selected from the misclassified examples by P_{iter}. If misclassified examples do not exist, base examples are selected from the whole training data.

In line 17, the $N_{replace}$ worst fuzzy rules are replaced in P_{iter} by the newly-generated $N_{replace}$ fuzzy rules. Then, the fitness of all fuzzy rules in P_{iter} are calculated (using the fitness function proposed) in line 18 and the CR of P_{iter} using q^{best} are calculated in line 19.

Finally, new better values of each exponent are searched in lines 20-26 (similar to lines 6-12), where, a randomly generated q^t and P_{iter} are stored as q^{best} and P_{best}, respectively, if the calculus of CR using both is better than previous one.

The final outputs P_{best} and q^{best} are the best population and the best values of exponents found during the evolution process.

4. Experiments

In this section, we present a computational experiment aimed to assess the performance of proposed Michigan_EE_CC algorithm modification when it is applied on ten datasets with varied numbers of examples, attributes and classes. Table 1 show the datasets used in this paper, which are available at KEEL dataset repository [22].

Table 1
Datasets used in this study

Dataset	Examples	Attributes	Classes	Dataset	Examples	Attributes	Classes
appendicitis	106	7	2	newthyroid	215	5	3
bupa	345	6	2	pima	768	8	2
glass	214	9	7	segment	2310	19	7
hayes-roth	160	4	3	tae	151	5	3
heart	270	13	2	wine	178	13	3

The parameters and genetic operators of the proposed Michigan_EE_CC algorithm modification used in this paper are listed in Table 2.

Table 2
Parameters and genetic operators used in this study

Parameter	Value	Parameter	Value
Population size (N_{rule})	30	Crossover probability ($crossProb$)	0.9
Number of replaced rules ($N_{replace}$)	6	Mutation probability ($mutaProb$)	$1/n$ (n: Number of attributes)
Number of Fuzzy Set (nFS)	5	Function membership mutation probability ($mutaProbFS$)	0.1
MPB (H)	2	TQ	5
Parent selection	Binary tournament selection	$Iter$	100000
Crossover	Uniform crossover	Number of runs	50

Table 3 shows the achieved results by the proposed Michigan_EE_CC algorithm modification for training and testing, each line describes the mean of the accuracy obtained after 50 runs (10-fold cross validation $\times$ five times) and the standard deviations in brackets.

In order to show the quality of the proposed Michigan_EE_CC algorithm, we compare it with the Michigan_EE algorithm. The parameters used in Michigan_EE algorithm are the same as Michigan_EE_CC algorithm, except for the TQ parameter that is not used. The results obtained by the Michigan_EE algorithm are shown in Table 3.

Table 3
Accuracy rate in training and testing for the proposed Michigan_EE_CC algorithm modification vs Michigan_EE algorithm base

Dataset	Michigan_EE		Michigan_EE_CC	
	Training	Testing	Training	Testing
appendicitis	0.9287 (0.0109)	0.8024 (0.0473)	**0.9382** (0.0104)	**0.8209** (0.0801)
bupa	**0.7307** (0.0217)	0.5943 (0.0631)	0.7264 (0.0119)	**0.6425** (0.0478)
glass	**0.7498** (0.0250)	**0.6281** (0.0662)	0.7155 (0.0149)	0.6133 (0.0899)
hayes-roth	0.8163 (0.01510)	0.7063 (0.0557)	**0.8832** (0.0062)	**0.7825** (0.0812)
heart	**0.8915** (0.0116)	0.7756 (0.0627)	0.8697 (0.0106)	**0.7763** (0.0467)
newthyroid	0.9008 (0.03180)	0.8230 (0.0376)	**0.9775** (0.0050)	**0.9370** (0.0291)
pima	0.7515 (0.0075)	0.6941 (0.0333)	**0.7795** (0.0033)	**0.7363** (0.0276)
segment	0.8555 (0.0305)	0.8428 (0.0366)	**0.8570** (0.0104)	**0.8480** (0.0173)
tae	0.6248 (0.0295)	0.5114 (0.0802)	**0.6574** (0.0123)	**0.5161** (0.1004)
wine	0.9889 (0.0106)	0.8805 (0.0368)	**0.9938** (0.0035)	**0.8895** (0.0472)
AVG	0.8239	0.7259	**0.8398**	**0.7562**

Table 3 shows that the proposed Michigan_EE_CC algorithm obtains better results than Michigan_EE algorithm in seven out of the ten training data and nine out of the ten testing data. We also consider the use of the Wilcoxon test [23] in order to perform pair-wise comparison on test results for the two algorithms.

Table 4 shows that the null hypothesis for the Wilcoxon's test has been rejected (p-value $\leqslant \alpha$) and we may conclude that proposed Michigan_EE_CC algorithm presents better results than previous version.

The source code of the proposed Michigan_EE_CC algorithm (GitHub) and our implementation of Michigan_EE (GitHub) algorithm are available on GitHub.

Table 4
Wilcoxon't Test ($\alpha = 0.05$)

Comparison	R^+	R^-	Hypothesis	p-value
Michigan_EE_CC vs. Michigan_EE	0.3187	0.0148	**Rejected**	0.025

5. Conclusions

In this paper, we proposed the Michigan_EE_CC algorithm, which is a modification of the Michigan-style fuzzy rule generation algorithm proposed in [19], using a Choquet-like Copula-based aggregation function. Michigan_EE_CC algorithm was applied to ten standard classification datasets and compared to the prominent original algorithm, named Michigan_EE, that uses winning rule aggregation function in the fuzzy reasoning method. The experimental results showed that the Michigan_EE_CC algorithm is able to increase the accuracy over the training and testing dataset.

We foresee different avenues for future work, they include: 1) using other generalizations of the Choquet integral and, 2) evaluating the performance with challenging datasets, i.e., imbalanced and high dimensional datasets.

Acknowledgments

This work was supported by the Universidad Nacional de San Agustin de Arequipa under Project IBAIB-06-2019-UNSA and in part by the Spanish Ministry of Economy and Competitiveness through the Spanish National Research (project PID2019-108392GB-I00 / AEI / 10.13039/501100011033) and by the Public University of Navarre under the project PJUPNA1926.

References

[1] S. D. Bharathi, S. Sudha, A survey on gene selection for microarray cancer classification based on soft computing techniques, in: 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018, pp. 304–309.

[2] N. Arunachalam, S. J. Sneka, G. MadhuMathi, A survey on text classification techniques for sentiment polarity detection, in: 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), 2017, pp. 1–5.

[3] H. P. Ünal, G. Gökmen, M. Yumurtacı, Emotion classification with deap dataset:survey, in: Innovations in Intelligent Systems and Applications Conference, 2020, pp. 1–6.

[4] S. Pathak, I. Mishra, A. Swetapadma, An assessment of decision tree based classification and regression algorithms, in: 2018 3rd International Conference on Inventive Computation Technologies (ICICT), 2018, pp. 92–95.

[5] G. Algan, I. Ulusoy, Image classification with deep learning in the presence of noisy labels: A survey, Knowledge-Based Systems 215 (2021) 106771.

[6] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications, Computer Science Review 40 (2021) 100379.

[7] M. Elkano, M. Galar, J. Sanz, H. Bustince, Chi-bd: A fuzzy rule-based classification system for big data classification problems, Fuzzy Sets and Systems 348 (2018) 75–101.

[8] H. Ishibuchi, T. Nakashima, M. Nii, Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing), Springer-Verlag, Berlin, Heidelberg, 2004.
[9] M. Gacto, R. Alcalá, F. Herrera, Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures, Information Sciences 181 (2011) 4340–4360. Special Issue on Interpretable Fuzzy Systems.
[10] A. Fernández, V. López, M. D. Jesús, F. Herrera, Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges, Knowl. Based Syst. 80 (2015) 109–121.
[11] C. H. Tan, K. S. Yap, S. Y. Wong, M. T. Au, C. T. Yaw, H. J. Yap, Genetic rules induction fuzzy inference system for classification and regression application in energy industry, International Journal of Engineering and Advanced Technology (IJEAT) 9 (2019) 4154–4160.
[12] A. Orriols-Puig, J. Casillas, E. Bernado-Mansilla, Fuzzy-ucs: A michigan-style learning fuzzy-classifier system for supervised learning, IEEE Transactions on Evolutionary Computation 13 (2009) 260–283.
[13] E. H. Cárdenas, H. A. Camargo, Y. J. Túpac, Imbalanced datasets in the generation of fuzzy classification systems-an investigation using a multiobjective evolutionary algorithm based on decomposition, in: IEEE International Conference on Fuzzy Systems, 2016, pp. 1445–1452.
[14] F. J. Berlanga, M. J. del Jesus, F. Herrera, A novel genetic cooperative-competitive fuzzy rule based learning method using genetic programming for high dimensional problems, in: 2008 3rd International Workshop on Genetic and Evolving Systems, 2008, pp. 101–106.
[15] E. Barrenechea, H. Bustince, J. Fernandez, D. Paternain, J. A. Sanz, Using the choquet integral in the fuzzy reasoning method of fuzzy rule-based classification systems, Axioms 2 (2013) 208–223.
[16] G. Lucca, J. A. Sanz, G. P. Dimuro, B. R. C. Bedregal, M. J. Asiain, M. Elkano, H. Bustince, Cc-integrals: Choquet-like copula-based aggregation functions and its application in fuzzy rule-based classification systems, Knowl. Based Syst. 119 (2017) 32–43.
[17] G. P. Dimuro, G. Lucca, B. R. C. Bedregal, R. Mesiar, J. A. Sanz, C. Lin, H. Bustince, Generalized cf1f2-integrals: From choquet-like aggregation to ordered directionally monotone functions, Fuzzy Sets Syst. 378 (2020) 44–67.
[18] L. Sun, H. Dong, A. X. Liu, Aggregation functions considering criteria interrelationships in fuzzy multi-criteria decision making: State-of-the-art, IEEE Access 6 (2018) 68104–68136.
[19] Y. Nojima, S. Takemura, K. Watanabe, H. Ishibuchi, Michigan-style fuzzy GBML with (1+1)-ES generation update and multi-pattern rule generation, in: Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems, IEEE, 2017, pp. 1–6.
[20] O. Cordón, M. J. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 20 (1999) 21–45.
[21] A. Gonzalez, R. Perez, Slave: a genetic learning system based on an iterative approach, IEEE Transactions on Fuzzy Systems 7 (1999) 176–191.
[22] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, F. Herrera, KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2011) 255–287.
[23] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (1945).