

# A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm<sup>☆</sup>

Michela Fazzolari<sup>\*</sup>, Rafael Alcalá, Francisco Herrera

Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

## ARTICLE INFO

### Article history:

Received 29 July 2013

Received in revised form 10 June 2014

Accepted 17 July 2014

Available online 2 August 2014

### Keywords:

Fuzzy discretization algorithm

Granularity learning

Fuzzy rule-based classification systems

Evolutionary algorithms

Multi-objective evolutionary fuzzy systems

Complexity-accuracy trade-off

## ABSTRACT

Multi-objective evolutionary algorithms represent an effective tool to improve the accuracy-interpretability trade-off of fuzzy rule-based classification systems. To this aim, a tuning process and a rule selection process can be combined to obtain a set of solutions with different trade-offs between the accuracy and the compactness of models. Nevertheless, an initial model needs to be defined, in particular the parameters that describe the partitions and the number of fuzzy sets of each variable (i.e. the granularities) must be determined. The simplest approach is to use a previously established single granularity and a uniform fuzzy partition for each variable. A better approach consists in automatically identifying from data the appropriate granularities and fuzzy partitions, since this usually leads to more accurate models.

This contribution presents a fuzzy discretization approach, which is used to generate automatically promising granularities and their associated fuzzy partitions. This mechanism is integrated within a Multi-Objective Fuzzy Association Rule-Based Classification method, namely D-MOFARC, which concurrently performs a tuning and a rule selection process on an initial knowledge base. The aim is to obtain fuzzy rule-based classification systems with high classification performances, while preserving their complexity.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Linguistic Fuzzy Rule-Based Systems (FRBSs) are used to model complex non-linear relationships, in which the input–output variables are characterized by imprecision and vagueness. With respect to other modeling approaches, they attempt to describe phenomena in a more human-like way of thinking. Linguistic FRBSs have been extensively used in several fields and applications, in particular they have been usefully applied for classification tasks. In this case, they are often referred to as Fuzzy Rule-Based Classification Systems (FRBCSs) [1].

<sup>☆</sup> This work was supported in part by the European Project 238819 under FP7-PEOPLE-ITN-2008 Marie Curie Action (Networks for Initial Training), by the Spanish Ministry of Education and Science under grant no. TIN2011-28488 and by the Andalusian Government under grant no. P10-TIC-6858.

<sup>\*</sup> Corresponding author. Tel: +34 685643937.

E-mail addresses: [fazzolari@decsai.ugr.es](mailto:fazzolari@decsai.ugr.es) (M. Fazzolari), [alcala@decsai.ugr.es](mailto:alcala@decsai.ugr.es) (R. Alcalá), [herrera@decsai.ugr.es](mailto:herrera@decsai.ugr.es) (F. Herrera).

A FRBS can be designed by an expert by hand or can be generated automatically according to a set of data that describes a certain phenomenon. Since the automatic definition of a FRBS can be considered an optimization problem, in the last decades the use of Evolutionary and in particular Genetic Algorithms (EAs/GAs) has been proposed to generate FRBSs, exploiting their ability to find near optimal solutions in complex search spaces and to incorporate a priori knowledge. The hybridization between FRBSs and GAs is currently known as Genetic Fuzzy System (GFSs) [2,3].

One of the main advantages of using linguistic FRBSs is their interpretability, that is the possibility for a human being to understand the fuzzy model [4]. In order to consider not only the accuracy, but also the interpretability during the design or optimization process, GFSs have been extended to a multi-objective approach: Multi-Objective Evolutionary Algorithms (MOEAs) have been used to design FRBSs with different trade-offs between two or more objectives, since they are able to find a set of compromise solutions when dealing with multi-criteria optimization problems [5,6]. Nowadays, the hybridization between MOEAs and FRBSs is

often referred to as Multi-Objective Evolutionary Fuzzy Systems (MOEFSs) [7].

In this proposal we focus on how to obtain highly accurate classification models, without neglecting their complexity. When building a FRBCS, several factors affect these two properties and among them one important factor is the definition of the initial Data Base (DB). When defining a suitable DB, one of the problems that arise is how to determine the number of fuzzy sets associated to the variables of the DB, i.e. the granularity of each variable, as pointed out in [8]. Thus, the first objective of this paper is to propose a novel granularity learning process to generate suitable granularities for defining the initial fuzzy partitions of the DB. Since this approach usually leads to an increased complexity of the generated models, we address this additional issue by integrating the granularity learning process within a multi-objective approach: in fact, although the main objective is the improvement of the model's precision, it has been demonstrated that a well-designed MOEFS manages to both improve precision and reduce complexity in comparison to a single-objective approach, due to the reduction of the overfitting and to a better exploration ability [9].

In summary, the objective of this proposal is twofold: in a first stage we use an algorithm based on fuzzy discretization to extract suitable granularities and construct a DB, so that the initial DB model highly fits the considered dataset, thus fostering the accuracy. Then, we integrate this mechanism within a MOEFS, which evolves the initial model and takes care of maintaining or even decreasing the complexity while improving the accuracy. In particular, the proposed method, namely Multi-Objective Fuzzy Association Rule-Based Classification Model with granularity learning based on Discretization (D-MOFARC), comprehends the following steps:

- Step 1. A Fuzzy Discretization Algorithm is designed, in order to learn automatically suitable granularities for each variable and to generate the correspondent fuzzy partitions. This approach is based on the concept of discretization [10,11], which represents the process of transforming the range of values of a continuous attribute in a set of intervals and assigning a discrete value to each interval. We extend this approach to the case of fuzzy partitions, considering attributes interdependencies. Therefore, after obtaining a set of intervals for each variable, a fuzzy set is assigned to each interval, instead of a discrete value, to obtain the fuzzy labels associated to each variable. This process has been integrated with a tree-based generation mechanism that considers attribute partitioning interdependencies.
- Step 2. An initial Rule Base (RB) associated to the previous fuzzy partitions is created by extracting candidate fuzzy association rules. To this aim, the first two steps of the Fuzzy Association Rule-Based Classification Model for High-Dimensional Problems (FARC-HD) method proposed in [12] have been used. Since the extracted rules do not use all the labels generated in the initial step, the initial fuzzy partitions are finally refined by removing the unused labels.
- Step 3. A new specific MOEA is designed to concurrently perform the tuning of MFs in the DB and the selection of rules in the RB. This algorithm is a modified version of the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [13], and aims to improve the accuracy while maintaining the complexity of the initial model, at the same time. As said, the multi-objective approach has been preferred because it has been demonstrated that the use of the accuracy as first objective enhances models' performance, while the use of the number of rules as second objective can limit models' complexity [9].

The proposed method has been tested over 35 datasets, including small size datasets and high dimensional and large scale

datasets. The obtained results show that the proposed method statistically improves the precision, with respect to the results obtained when using a single-objective approach.

This proposal is arranged as follows. Section 2 introduces some preliminary concepts about the problem of learning granularities and about discretization methods in general and the CAIM discretization algorithm [14] in particular. Section 3 illustrates the characteristics of the fuzzy discretization approach. In Section 4 the features of the proposed method are introduced and described in detail. Section 5 illustrates the experimental framework, in which the experimental setup is described and the obtained results are presented and discussed. Finally, in Section 6, some concluding remarks are pointed out.

## 2. Related work: granularity learning and discretization approaches

In this section some preliminary concepts are briefly introduced. The first part deals with the problem of specifying suitable granularities. In the second part the concept of discretization is described and a specific discretization algorithm, i.e. CAIM [14], is presented.

### 2.1. Multi-objective evolutionary fuzzy systems: the problem of learning granularities

A FRBS is mainly composed of a Knowledge Base (KB) and an inference engine system. The KB includes the fuzzy rules in the RB and the linguistic variable descriptions in the DB. The inference engine system is composed of a fuzzification interface to transform crisp data into fuzzy sets, an inference system that uses them together with the KB to produce the output, a defuzzification interface to transform the fuzzy output into a crisp value.

Several approaches have been proposed to automatically design the KB of FRBSs, by using numerical data that provide a description of a problem. For example in [15], a method to automatically learn the data base of a Mamdani fuzzy system for regression problems is described. Regarding classification problems, most of the proposed approaches have focused on learning a RB, using a predefined DB. In this case, the main issue is the identification of suitable fuzzy partitions for the variables involved. In particular, one of the problems concerns the identification of the granularities associated with the fuzzy partitions in the DB. Granularities can affect both the accuracy and the interpretability of FRBCSs and the choice of suitable granularities is crucial, as discussed in [8]. In this contribution, the authors have pointed out that there does not exist a unique optimal granularity, but the choice depends on the type of the problem and also on the RB learning method.

The easiest strategy consists in fixing a single a-priori granularity and creating uniform fuzzy partitions for all the variables [16,17]. Despite its simplicity, this approach can not be the most appropriate since it does not consider at all the available knowledge of a problem and usually leads to models with a bad trade-off between accuracy and complexity.

Another possibility is to use multiple granularities for the same variable in the DB. For example, in [18] the authors have used a simple method to generate an initial set of candidate fuzzy association rules for classification, then they have applied a pre-screening approach to reduce the number of these rules. The associated DB has been created by considering multiple granularities. In a further stage, the initial RB is reduced by applying a genetic algorithm, therefore the selection of suitable granularities is promoted by the elimination of unnecessary rules, so that the granularities that never appear in any rule are discarded.

Also in [19], the authors have used multiple granularities in the DB. The method they have proposed aims at learning linguistic rules

to generate regression models, by means of a hierarchical system. To this aim, the structure of the KB of a FRBS has been extended in a hierarchical way, so that linguistic rules can be defined over linguistic partitions with different granularity levels. This approach improves the modeling of problems in certain subspaces.

Two further proposals dealing with the problem of choosing suitable granularities can be found in [20,21], where two MOEAs have been proposed to learn FRBSs to generate regression models. In this case, the granularity learning process has been integrated in the evolutionary algorithm. To this end, the concepts of virtual and concrete fuzzy partitions have been proposed: the former has been defined by partitioning in an uniform way each linguistic variable with a fixed maximum granularity; the latter takes into account, for each variable, the granularities determined by the evolutionary process. Since RB and DB parameters have been defined on the virtual partitions, whenever a fitness evaluation is required, the virtual partitions are mapped to the concrete partitions by employing appropriate mapping strategies.

Another interesting contribution to this topic can be found in [22]. In this work, the authors have used a two-levels MOEA to obtain a set of FRBSs with different trade-offs between accuracy and complexity. The presented approach uses a high level MOEA to determine the number of fuzzy sets for each variable and multiple distributed low-level evolutionary heuristics to tune the membership functions of the candidate variable partitions.

In a recent proposal [23], a method has been presented to generate single granularity-based fuzzy classification rules for multi-objective genetic fuzzy rule selection. This method is made of four steps: in the first one, a set of candidate fuzzy rules is generated by a heuristic procedure using multiple granularities. At the end of this phase, a single granularity for each variable is chosen, according to the frequency of employed partitions and the importance of the multiple granularity-based extracted rules. Then, these granularities are used to extract again a set of fuzzy rules, which will be evolved by an MOEA to perform a multi-objective genetic fuzzy rule selection process.

The concept of granularities applied to interval type-2 fuzzy models has been considered in [24]. In this contribution, the authors have proposed a method to design type-2 fuzzy models, exploiting a new performance index to drive the construction of the fuzzy models. The proposed index is formulated in a such a way that promotes the generation of fuzzy intervals that “cover” the experimental data, being at the same time as narrow (i.e. specific) as possible. The underlying idea is that an optimal granularity allocation throughout the membership functions used in the fuzzy model leads to the best design. The design process is performed by means of a genetic algorithm that helps in further improving the results.

The problem of designing optimal linguistic terms is also addressed in [25], where the authors have proposed an approach to design FRBCSs that includes the genetic design of optimal linguistic terms. A genetic simulated annealing algorithm has been used to design the optimal linguistic terms and their fuzzy-set-based semantics (including the granularity) for each variable in the dataset.

An interesting analysis has been presented in [26] about the intrinsic implementation of the principle of justifiable granularity [27] in clustering algorithms. In fact, clustering techniques represent another possibility to construct fuzzy models [28]. This proposal describes a first approach in the analysis of the relationships between the size of the cluster found by a clustering algorithm and the intrinsic implementation of the principle of justifiable granularity.

Nevertheless, the methods described so far present some limitations when dealing with high dimensional problems, due to the increase of the search space and of the time required by the fitness evaluation in the MOEA.

## 2.2. Discretization algorithms and CAIM

A possible approach for learning granularities is represented by discretization algorithms, which transform continuous attribute's values into a set of intervals and assign to each interval a numerical and discrete value. The aim is to minimize the number of discrete intervals while maximizing the interdependency between class labels and discrete attribute values, to prevent an excessive information loss during the discretization. A discretization scheme  $D$  on a continuous attribute  $A$  divides the continuous domain of  $A$  into  $n$  discrete intervals and can be represented as follows:

$$D : \{[d_0, d_1], (d_1, d_2], \dots, (d_{n-1}, d_n]\}, \quad (1)$$

where  $d_0$  and  $d_n$  are the minimum and the maximum values of the attribute  $A$ , respectively, and the values  $d_i$  identify the *cut points* for the discretization  $D$ .

The process consists of two steps: first, the number of intervals for each attribute is found and then the boundaries of the intervals are determined. Usually the first task is not performed automatically and the number of intervals must be specified by the user. Discretization algorithms can be grouped into two categories:

- Unsupervised (or class-blind) algorithms generate intervals without considering the class labels of each pattern.
- Supervised algorithms discretize attributes by considering the interdependence between class labels and the attribute values.

We will focus on this latter type. In [14] a discretization algorithm is proposed, that automatically determines the number of intervals for data partitioning and concurrently finds the boundaries of each interval. The method is named CAIM, from the name of the criterion optimized to measure the dependency between the class  $C$  and the discretization  $D$  for the attribute  $A$ . The criterion is defined for a given attribute by means of a matrix, called *quanta matrix*. Table 1 shows the quanta matrix for a generic attribute  $A$ , where  $S$  is the number of classes,  $q_{ir}$  is the total number of patterns that belong to the  $i$ th class and have the value of the attribute  $A$  in the interval  $(d_{r-1}, d_r]$ ,  $M_{i+}$  is the total number of patterns belonging to the  $i$ th class,  $M_{+r}$  is the total number of values of the attribute  $A$  that are in the interval  $(d_{r-1}, d_r]$ , for  $i = 1, \dots, S$  and  $r = 1, \dots, n$  and  $M$  is the total number of patterns.

The CAIM criterion is expressed as follows:

$$CAIM(C, D|A) = \frac{\sum_{r=1}^n (\max_r^2) / (M_{+r})}{n} \quad (2)$$

where  $n$  is the number of intervals,  $\max_r$  is the maximum value among all  $q_{ir}$  values and  $i = 1, \dots, S$ .

The space of all the possible discretization schemes is large and the optimal scheme cannot be identified using exact algorithms, due to the time necessary to perform this process. Thus, the CAIM algorithm searches for an approximation of the optimal scheme by finding local maximum values of the CAIM criterion. This approach has demonstrated to be computationally not expensive and to lead to good approximations of the optimal discretization scheme. This algorithm is executed in two steps:

- For each continuous attribute  $A$ , a set of candidate interval boundaries  $B$  (i.e. the candidate *cut points*) and an initial discretization scheme are created: first, the minimum ( $d_0$ ) and maximum ( $d_n$ ) values of the attribute's domain are found and added to the set of candidate cut points. Then, all the distinct values of  $A$  in the dataset are considered and for each adjacent pair of values, a midpoint is calculated. All the midpoints are added to the set of candidate cut-points  $B$ . The algorithm starts by considering the a discretization scheme ( $D : [d_0, d_n]$ ) that covers all the possible

**Table 1**  
Quanta matrix for attribute *A* and discretization scheme *D*.

| Class          | Interval     |     |                  |     |                  | Class Total |
|----------------|--------------|-----|------------------|-----|------------------|-------------|
|                | $[d_0, d_1]$ | ... | $(d_{r-1}, d_r)$ | ... | $(d_{n-1}, d_n)$ |             |
| $C_1$          | $q_{11}$     | ... | $q_{1r}$         | ... | $q_{1n}$         | $M_{1+}$    |
| $\vdots$       | $\vdots$     | ... | $\vdots$         | ... | $\vdots$         | $\vdots$    |
| $C_i$          | $q_{i1}$     | ... | $q_{ir}$         | ... | $q_{in}$         | $M_{i+}$    |
| $\vdots$       | $\vdots$     | ... | $\vdots$         | ... | $\vdots$         | $\vdots$    |
| $C_S$          | $q_{S1}$     | ... | $q_{Sr}$         | ... | $q_{Sn}$         | $M_{S+}$    |
| Interval Total | $M_{+1}$     | ... | $M_{+r}$         | ... | $M_{+n}$         | $M$         |

values of a continuous attribute and with the criterion *GlobalCAIM* = 0.

- These new cut-points are added to the initial discretization according to the maximization of the CAIM criterion. From all candidate cut-points, the algorithm selects the point that gives the highest value of the CAIM criterion, then this point is added to the discretization if  $CAIM > GlobalCAIM$  or if there are less than *S* intervals in the discretization. In fact, it is assumed that each discretized attribute needs at least the number of discretization intervals to be equal to the number of classes, since this guarantees that the discretized attribute can improve the subsequent classification. After the addition of a cut-point to the discretization, the *GlobalCAIM* criterion is updated and assigned with the CAIM value of the added cut-point.

A detailed description of the algorithm can be found in the original proposal [14]. The CAIM algorithm generates separately a single discretization *D* for each attribute. However, when considering FRBSs, there is an interdependency among attributes that reflects in the extracted rules, thus a discretization approach should take into account this aspect. Therefore, in the following section we propose a new discretization algorithm that tries to exploit the interdependence among attributes to generate fuzzy discretizations, i.e. the initial fuzzy partitions.

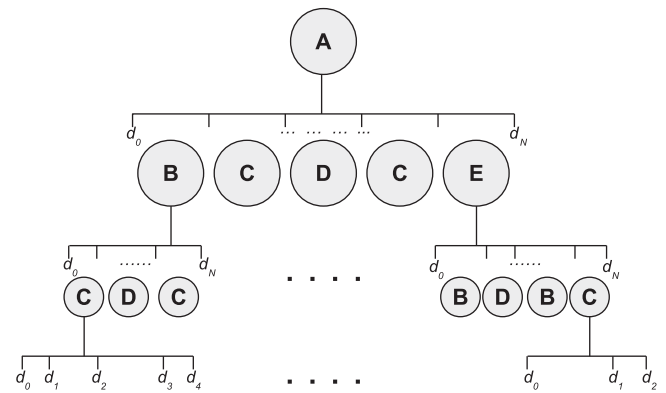
**3. Data base extraction: a proposal for fuzzy discretization on interdependent attribute scenarios**

In order to extract an initial set of fuzzy rules from a dataset, it is necessary to build an associated DB and in particular to choose a granularity for each variable [8,23]. We already highlighted how an approach in which the granularity is fixed by hand may not be the best choice, since it leads to models with a bad trade-off between accuracy and complexity.

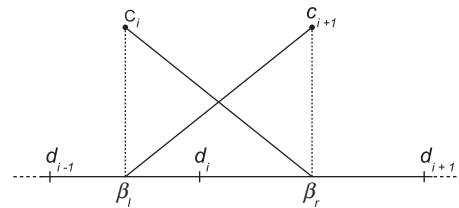
To this end, we present a fuzzy discretization method to generate automatically a promising set of labels for each attribute, without specifying any apriori granularity. For each attribute, the proposed approach generates a set of promising fuzzy partitions rather than a single partition, so that a wider number of labels will be available for the rule extraction process. The number of fuzzy partitions is fixed to a maximum of 3 for each attribute.

As explained in the previous section, the CAIM algorithm can be used to discretize the input space of an attribute, but it does not consider possible dependencies among attributes to generate this discretization. Therefore, we have extended the CAIM algorithm and we have designed a novel method that takes into account attributes interdependences by using dependency trees: a tree is constructed for each attribute by setting this attribute as root node and the branches are used to assess the level of dependency between the root attribute and the other ones.

In addition, we are interested not only in determining granularities (i.e. the number of fuzzy sets), but also in defining the initial fuzzy partitions (i.e. the shapes of fuzzy sets) for each attribute. Thus, once the discretizations are obtained, a further process is



**Fig. 1.** An example of generated tree for an attribute.



**Fig. 2.** Fuzzification of a discretization according to the extended fuzzy CAIM criterion.

applied to *fuzzify* these discretizations, i.e. to transform them into fuzzy partitions.

The proposed procedure, named Fuzzy Discretization Algorithm, comprehends the following three steps:

- Ordering the attributes according to a fitness, based on the CAIM criterion.
- For each attribute, generating a set of discretizations.
- Transforming each discretization into a fuzzy partition to obtain the initial DB.

**3.1. Step 1: ordering the attributes**

When dealing with classification datasets, attributes are supposed to be more or less important for the classification task. To establish an order among them, the CAIM algorithm is initially executed and a discretization for each attribute is obtained. The discretizations are evaluated with respect to the CAIM criterion (2), thus the corresponding attributes can be ordered from best to worst. Therefore, the CAIM algorithm itself is used to establish an order among attributes. The construction of the dependency trees will follow this order, so that more important attributes will have more influence in determining the resulting fuzzy partitions.

**Table 2**  
List of the datasets used in the study.

| DATASET       | # INST   | # ATTR(NUM/NOM) | # CLS |
|---------------|----------|-----------------|-------|
| iris          | 150      | 4(4/0)          | 3     |
| tae           | 151      | 5(5/0)          | 3     |
| hepatitis     | 155      | 19(19/0)        | 2     |
| wine          | 178      | 13(13/0)        | 3     |
| automobile    | 205      | 25(15/10)       | 6     |
| glass         | 214      | 9(9/0)          | 7     |
| newthyroid    | 215      | 5(5/0)          | 3     |
| heart         | 270      | 13(13/0)        | 2     |
| cleveland     | 297(303) | 13(13/0)        | 5     |
| haberman      | 306      | 3(3/0)          | 2     |
| ecoli         | 336      | 7(7/0)          | 8     |
| bupa          | 345      | 6(6/0)          | 2     |
| balance       | 625      | 4(4/0)          | 3     |
| crx           | 653(690) | 15(6/9)         | 2     |
| australian    | 690      | 14(8/6)         | 2     |
| wisconsin     | 699      | 9(9/0)          | 2     |
| pima          | 768      | 8(8/0)          | 2     |
| vehicle       | 846      | 18(18/0)        | 4     |
| german        | 1000     | 20(7/13)        | 2     |
| contraceptive | 1473     | 9(9/0)          | 3     |
| titanic       | 2201     | 3(3/0)          | 2     |
| segment       | 2310     | 19(19/0)        | 7     |
| spambase      | 4597     | 57(57/0)        | 2     |
| banana        | 5300     | 2(2/0)          | 2     |
| phoneme       | 5404     | 5(5/0)          | 2     |
| page-blocks   | 5472     | 10(10/0)        | 5     |
| texture       | 5500     | 40(40/0)        | 11    |
| optdigits     | 5620     | 64(64/0)        | 10    |
| satimage      | 6435     | 36(36/0)        | 7     |
| thyroid       | 7200     | 21(21/0)        | 3     |
| ring          | 7400     | 20(20/0)        | 2     |
| twonorm       | 7400     | 20(20/0)        | 2     |
| coil2000      | 9822     | 85(85/0)        | 2     |
| penbased      | 10992    | 16(16/0)        | 10    |
| magic         | 19020    | 10(10/0)        | 2     |

Source: <http://sci2s.ugr.es/keel/datasets.php>.

### 3.2. Step 2: generating the discretizations

In this step a set of discretizations is generated for each attribute by using dependency trees. A dependency tree is a tree in which an attribute is set as the root node and then other attributes are added as children, by choosing the ones that present the highest interdependency with the father. The details of the process are explained in the following.

At each step, to each node of the tree a specific discretization is associated. For the root nodes, in the previous step a discretization has already been created by the CAIM algorithm, but this discretization is discarded, since the CAIM algorithm usually promotes discretizations with a small number of cut points, especially when a dataset presents few classes. On the contrary, the first discretization for each attribute is generated by considering equidistributed intervals and by fixing the number of cut points to a higher value, in order to promote better granularities. In this proposal, the number of cut points for the initial discretizations has been fixed to 4, to generate at least 5 fuzzy labels for each attribute.

Based on the discretization of each parent node, new attributes are selected as children in a recursive manner. Let's consider a parent node attribute  $A$  and its corresponding discretization  $D_A$ . The input patterns are sorted with respect to the values of attribute  $A$  and they are grouped according to the intervals  $i = 1, \dots, I$  defined by the discretization  $D_A$ , where  $I$  is the number of intervals of discretization  $D_A$ . Let's now consider the group of patterns corresponding to an interval  $i$ . The CAIM algorithm is applied on these patterns, considering all the attributes  $T_j \neq A$  ( $j = 1, \dots, S$ ) and for each of them a discretization is obtained. These discretizations are evaluated according to the CAIM criterion and the attribute

corresponding to the discretization that maximize the criterion is chosen as child node  $B$  for the interval  $i$  in the tree.

This process is repeated for each interval in the discretization  $D_A$ . After generating all the children of a node (i.e. a child for each interval) and their corresponding discretizations, the entire step is repeated recursively for each child (Fig. 1) and it stops when one of the following conditions occurs:

- only two patterns of the dataset fall in an interval.
- the depth of the tree reaches  $Depth_{max}$ , fixed to 3.

After this step, a dependency tree for each attribute has been built and for each node of the tree a discretization has been generated. However, for each attribute a maximum of three discretizations are stored, in particular the first encountered in the trees' generation process, that are supposed to be the best ones. Thus, the order considered for the construction of the dependency trees influences the decision about the selected fuzzy partition.

### 3.3. Step 3: generating the data base

In this step, the discretizations corresponding to each attributes are transformed into fuzzy partitions, by applying the concept of strong fuzzy partition [2], which guarantees the fulfillment of many interpretability constraints. Thus, a triangular fuzzy set is generated across each interval in the following manner: the centroid  $c_i$  is set in correspondence with the middle point between the boundaries of the interval  $(d_{i-1}, d_i]$ , while the left and right parameters  $\beta_l$  and  $\beta_r$  of the labels between two adjacent centroids  $c_i$  and  $c_{i+1}$  are set in correspondence with the adjacent centroids themselves (Fig. 2). This mechanism ensures the creation of strong fuzzy partitions.

## 4. Multi-objective fuzzy association rule-based classification algorithm with granularity learning based on discretization (D-MOFARC)

The discretization approach presented so far has been integrated within a new Multi-Objective Evolutionary method, which evolves an initial KB by performing concurrently a rule selection process and a tuning process. The aim is to improve the precision of the initial fuzzy model through the tuning of the MFs, while maintaining or decreasing the complexity by means of the selection of rules. This proposed method can be summarized in three stages:

- Data base extraction. The Fuzzy Discretization Algorithm described in Section 3 is used to construct a set of initial fuzzy partitions.
- Rule base extraction. An initial RB associated to the previous fuzzy partitions is created. To this aim, the first two steps of the Fuzzy Association Rule-Based Classification model for High Dimensional problem (FARC-HD) proposed in [12] have been used. This method exploits the Apriori algorithm for mining fuzzy association rules. The synergy between the Apriori algorithm and fuzzy association rules leads to the construction of accurate classification models, in fact this method is able to extract a set of high quality classification rules from huge amounts of data in a reasonable time. Furthermore, unlike Wang and Mendel's method, this approach generates more interpretable rules, since in each rule just a subset of variables is used. After generating the initial RB, some of the labels belonging to the initial fuzzy partitions are never used in any of the extracted rules, therefore the initial partitions are refined by removing the unused labels and an initial DB is generated.

**Table 3**  
Comparison of the average results obtained by applying the D-MOFARC method and the FARC-HD algorithm.

| 10 folds<br>Dataset | D-MOFARC    |     |             |             | FARC-HD |            |       |             |
|---------------------|-------------|-----|-------------|-------------|---------|------------|-------|-------------|
|                     | #R          | #C  | Tra         | Tst         | #R      | #C         | Tra   | Tst         |
| iris                | 5.6         | 1.1 | 98.1        | <b>96.0</b> | 4.4     | 1.1        | 98.6  | 95.3        |
| tae                 | 20.2        | 2.7 | 82.1        | <b>59.3</b> | 19.9    | 2.3        | 71.7  | 59.0        |
| hepatitis           | 11.4        | 1.9 | 100.0       | <b>90.0</b> | 10.4    | 2.1        | 99.4  | 88.7        |
| wine                | 8.6         | 1.8 | 100.0       | <b>95.8</b> | 8.3     | 1.5        | 100.0 | 95.5        |
| automobile          | 38.9        | 2.7 | 99.3        | <b>81.0</b> | 34.1    | 2.7        | 96.6  | 76.8        |
| glass               | 27.4        | 2.6 | 95.2        | <b>70.6</b> | 18.2    | 2.4        | 79.0  | 69.0        |
| newthyroid          | 9.5         | 1.7 | 99.8        | <b>95.5</b> | 9.6     | 1.8        | 99.2  | 94.4        |
| heart               | 18.7        | 2.7 | 94.4        | <b>84.4</b> | 27.8    | 2.6        | 93.1  | 83.7        |
| cleveland           | 45.6        | 2.9 | 90.9        | 52.9        | 42.1    | 2.8        | 82.2  | <b>58.3</b> |
| haberman            | 9.2         | 2.0 | 81.7        | 69.4        | 5.7     | 1.2        | 79.2  | <b>73.5</b> |
| ecoli               | 26.2        | 2.5 | 94.0        | <b>82.7</b> | 32.2    | 2.4        | 91.6  | 81.2        |
| bupa                | 7.7         | 2.6 | 82.8        | <b>70.1</b> | 10.6    | 1.9        | 78.2  | 66.4        |
| balance             | 20.1        | 2.3 | 89.4        | 85.6        | 18.8    | 1.0        | 92.2  | <b>91.2</b> |
| crx                 | 22.6        | 2.6 | 91.6        | <b>84.9</b> | 24.4    | 2.4        | 91.0  | 84.7        |
| australian          | 19.8        | 2.7 | 91.3        | <b>86.0</b> | 25.9    | 2.5        | 90.8  | <b>86.4</b> |
| wisconsin           | 9.0         | 1.4 | 98.6        | <b>96.8</b> | 13.6    | 1.1        | 98.3  | 96.2        |
| pima                | 10.4        | 2.5 | 82.3        | 75.5        | 20.2    | 2.3        | 82.3  | <b>76.2</b> |
| vehicle             | 22.4        | 2.8 | 84.5        | <b>70.6</b> | 31.6    | 2.5        | 77.2  | 68.0        |
| german              | 55.4        | 2.8 | 86.8        | <b>72.4</b> | 63.3    | 2.7        | 83.4  | 71.7        |
| contraceptive       | 49.0        | 2.9 | 63.6        | 53.6        | 68.5    | 2.6        | 61.2  | <b>54.1</b> |
| titanic             | 10.4        | 1.8 | 78.9        | 78.7        | 4.1     | 1.3        | 79.1  | <b>78.8</b> |
| segment             | 26.2        | 2.5 | 98.0        | <b>96.6</b> | 41.1    | 2.7        | 94.8  | 93.3        |
| spambase            | 24.3        | 2.2 | 91.7        | 90.5        | 30.5    | 2.3        | 92.4  | <b>91.6</b> |
| banana              | 8.7         | 2.0 | 90.3        | <b>89.0</b> | 12.9    | 1.5        | 86.0  | 85.5        |
| phoneme             | 9.3         | 2.4 | 84.8        | <b>83.5</b> | 17.2    | 2.2        | 83.9  | 82.4        |
| page-blocks         | 21.5        | 2.4 | 97.8        | <b>97.0</b> | 18.4    | 2.3        | 95.5  | 95.0        |
| texture             | 60.5        | 2.8 | 97.5        | <b>95.2</b> | 50.9    | 2.7        | 93.7  | 93.0        |
| optdigits           | 135.4       | 3.0 | 97.0        | 93.4        | 204.0   | 3.0        | 97.0  | <b>93.5</b> |
| satimage            | 56.0        | 2.8 | 90.8        | <b>87.5</b> | 30.2    | 2.8        | 84.4  | 83.8        |
| thyroid             | 5.9         | 2.4 | 99.3        | <b>99.1</b> | 4.9     | 1.8        | 94.3  | 94.1        |
| ring                | 15.3        | 1.7 | 94.2        | 93.3        | 24.9    | 1.9        | 95.1  | <b>94.0</b> |
| twonorm             | 10.2        | 2.7 | 94.5        | 93.1        | 60.4    | 2.6        | 96.6  | <b>95.1</b> |
| coil2000            | 89.0        | 1.0 | 94.0        | <b>94.0</b> | 2.6     | 1.0        | 94.0  | <b>94.0</b> |
| penbased            | 119.2       | 2.9 | 97.4        | <b>96.2</b> | 152.7   | 2.8        | 97.0  | 96.0        |
| magic               | 32.2        | 2.6 | 86.3        | <b>85.4</b> | 43.8    | 2.5        | 85.4  | 84.8        |
| Mean                | <b>30.3</b> | 2.4 | <b>91.4</b> | <b>84.2</b> | 33.9    | <b>2.2</b> | 89.0  | 83.6        |

- Data base tuning and rule selection. A new specifically designed MOEA is proposed, which concurrently performs the MFs tuning of the DB and the selection of rules from the RB. This algorithm is a modified version of the Strength Pareto Evolutionary Algorithm 2 (SPEA2)[13], and aims to improve the accuracy while reducing the complexity of the initial model at the same time.

Hereinafter, the stage 2 and 3 of the method are described in detail.

#### 4.1. Rule base extraction

The Fuzzy Discretization Algorithm generates a maximum of three fuzzy partitions for each attribute, which are used to derive the initial RB. To this aim, the first two steps of the approach presented in [12] have been applied. This method uses fuzzy association rules to codify the information extracted from a dataset, that are an extension of association rules, used to represent dependencies between itemsets in a database [29,30]. Fuzzy association rules can consider not only binary or discrete values, but also quantitative values, and can be used as classification rules if their consequent is expressed with a class label. The generation of the RB is performed in two consecutive steps. In the following we briefly describe these two steps (for a more detailed explanation see [12]).

##### 4.1.1. Step 1: rule extraction

In this first step, a set candidate association rules is extracted: for each class a search tree is built, in order to list all the possible frequent itemsets of a class. Itemsets are constructed using a search tree. The root level of each tree (level 0) is generated as an empty set and all the one-itemsets constitute the first level of the

**Table 4**  
Wilcoxon's statistics on test accuracy and number of rules for D-MOFARC ( $R^+$ ) vs FARC-HD ( $R^-$ ), considering 35 datasets (10 folds).

| Comparison (Tst Acc) | $R^+$ | $R^-$ | p-value | Hypothesis   |
|----------------------|-------|-------|---------|--------------|
| D-MOFARC vs FARC-HD  | 414.5 | 180.5 | 0.04544 | Rejected     |
| Comparison (#Rules)  | $R^+$ | $R^-$ | p-value | Hypothesis   |
| D-MOFARC vs FARC-HD  | 428.0 | 202.0 | 0.0649  | Not Rejected |

search tree (level 1). The further level (level 2) for an attribute  $A$  is constructed by considering all the two-itemsets that combine the one-itemset of attribute  $A$  with all the one-itemsets for the other attributes. The same procedure is used to construct the following levels of the tree. No repeated itemsets appear in the tree. Once all the frequent itemsets have been listed, a candidate fuzzy association rule is constructed for each itemset, by setting the itemset itself in the antecedent and the corresponding class in the consequent. The maximum depth of the trees is fixed to three, in order to generate rules with few antecedent conditions, thus keeping the model simple.

##### 4.1.2. Step 2: rule prescreening

The rule extraction process generates a large number of rules, which can cause a problem of rule redundancy. To decrease this number by selecting only the best rules, a subgroup discovery technique is used, in particular the pattern weighting scheme described in [31]. Each pattern is associated to a weight  $w(i) = 1/(i + 1)$ , where  $i$  stores how many times the pattern has been covered by a rule. Initially, all the weights assume the same value  $w(0) = 1$ .

**Table 5**  
Average results obtained for the FIRST, MEDIAN and LAST solutions.

| DATASET       | FIRST |       |      |      |      |        |      |      | MEDIAN |       |      |      |      |       |      |       | LAST  |      |      |      |      |       |       |       |       |
|---------------|-------|-------|------|------|------|--------|------|------|--------|-------|------|------|------|-------|------|-------|-------|------|------|------|------|-------|-------|-------|-------|
|               | #R    |       | #C   |      | #Tra |        | #Tst |      | #R     |       | #C   |      | #Tra |       | #Tst |       | #R    |      | #C   |      | #Tra |       | #Tst  |       |       |
|               | CARD  | MEAN  | SD   | MEAN | SD   | MEAN   | SD   | MEAN | SD     | MEAN  | SD   | MEAN | SD   | MEAN  | SD   | MEAN  | SD    | MEAN | SD   | MEAN | SD   | MEAN  | SD    |       |       |
| iris          | 2.73  | 5.6   | 1.1  | 1.1  | 0.2  | 98.11  | 0.38 | 96.0 | 4.19   | 2.6   | 0.5  | 1.2  | 0.3  | 98.05 | 0.45 | 95.33 | 4.91  | 2.0  | 0.0  | 1.0  | 0.0  | 97.63 | 0.52  | 95.56 | 4.97  |
| tae           | 13.40 | 20.2  | 1.8  | 2.7  | 0.1  | 82.14  | 1.94 | 59.3 | 11.79  | 10.2  | 1.3  | 2.7  | 0.1  | 76.16 | 1.93 | 58.04 | 14.76 | 4.0  | 1.4  | 2.7  | 0.3  | 59.11 | 7.50  | 50.24 | 14.20 |
| hepatitis     | 3.00  | 11.4  | 0.4  | 1.9  | 0.3  | 100.00 | 0.00 | 90.0 | 11.96  | 3.9   | 0.3  | 2.3  | 0.5  | 98.33 | 0.56 | 90.00 | 9.90  | 1.0  | 0.0  | 3.0  | 0.0  | 96.34 | 0.67  | 90.42 | 8.34  |
| wine          | 3.43  | 8.6   | 0.6  | 1.8  | 0.1  | 100.00 | 0.00 | 95.8 | 3.88   | 3.0   | 0.2  | 1.3  | 0.2  | 97.86 | 0.77 | 92.30 | 4.54  | 2.0  | 0.0  | 1.4  | 0.3  | 91.86 | 1.07  | 87.09 | 5.59  |
| automobile    | 7.20  | 38.9  | 2.3  | 2.7  | 0.1  | 99.35  | 0.60 | 81.0 | 8.56   | 18.5  | 2.1  | 2.7  | 0.1  | 96.88 | 0.97 | 77.85 | 8.26  | 15.5 | 2.2  | 2.7  | 0.2  | 90.01 | 7.14  | 71.36 | 13.58 |
| glass         | 15.13 | 27.4  | 3.1  | 2.6  | 0.1  | 95.19  | 2.30 | 70.6 | 13.24  | 15.7  | 2.7  | 2.6  | 0.1  | 87.62 | 2.68 | 67.04 | 11.33 | 8.5  | 2.5  | 2.6  | 0.2  | 75.17 | 5.71  | 61.79 | 12.20 |
| new-thyroid   | 5.17  | 9.5   | 1.3  | 1.7  | 0.2  | 98.83  | 0.24 | 95.5 | 4.55   | 3.9   | 0.5  | 1.3  | 0.4  | 97.97 | 0.99 | 95.40 | 3.93  | 2.0  | 0.0  | 1.2  | 0.4  | 86.84 | 3.13  | 86.54 | 6.21  |
| heart         | 13.17 | 18.7  | 2.5  | 2.7  | 0.1  | 94.44  | 0.78 | 84.4 | 8.75   | 7.3   | 1.2  | 2.8  | 0.2  | 91.19 | 0.95 | 81.23 | 8.81  | 1.6  | 0.8  | 2.6  | 0.4  | 74.28 | 6.64  | 72.22 | 8.91  |
| cleveland     | 18.33 | 45.6  | 5.2  | 2.9  | 0.0  | 90.92  | 1.35 | 52.9 | 6.65   | 31.5  | 4.3  | 2.9  | 0.0  | 85.02 | 3.19 | 53.60 | 6.94  | 20.2 | 4.7  | 2.9  | 0.0  | 76.16 | 4.10  | 54.34 | 6.82  |
| haberman      | 7.10  | 9.2   | 1.8  | 2.0  | 0.3  | 81.72  | 0.92 | 69.4 | 6.06   | 4.1   | 1.0  | 1.8  | 0.5  | 79.86 | 0.69 | 71.39 | 7.37  | 1.1  | 0.3  | 2.2  | 0.6  | 76.72 | 1.31  | 73.92 | 5.51  |
| ecoli         | 13.87 | 26.2  | 2.2  | 2.5  | 0.1  | 94.03  | 0.65 | 82.8 | 4.84   | 19.8  | 2.2  | 2.5  | 0.1  | 91.39 | 1.88 | 81.65 | 3.92  | 13.1 | 2.7  | 2.5  | 0.1  | 83.06 | 4.06  | 76.82 | 6.29  |
| bupa          | 7.33  | 7.7   | 1.5  | 2.6  | 0.1  | 82.78  | 1.20 | 70.2 | 10.51  | 4.1   | 0.8  | 2.5  | 0.2  | 76.92 | 1.59 | 66.71 | 9.67  | 1.1  | 0.4  | 2.7  | 0.4  | 61.41 | 3.84  | 57.45 | 9.83  |
| balance       | 1.17  | 19.8  | 6.5  | 2.3  | 0.1  | 89.47  | 0.44 | 85.6 | 3.26   | 18.5  | 4.8  | 2.3  | 0.1  | 89.46 | 0.48 | 85.93 | 3.37  | 18.4 | 4.9  | 2.3  | 0.1  | 89.43 | 0.50  | 85.88 | 3.41  |
| crx           | 10.57 | 22.6  | 2.8  | 2.6  | 0.1  | 91.62  | 0.56 | 84.9 | 4.60   | 15.2  | 3.1  | 2.5  | 0.1  | 90.87 | 0.85 | 84.89 | 4.82  | 10.8 | 3.7  | 2.5  | 0.2  | 83.45 | 7.94  | 80.02 | 9.60  |
| australian    | 11.90 | 19.8  | 3.6  | 2.7  | 0.1  | 91.30  | 0.42 | 86.0 | 4.40   | 10.3  | 2.3  | 2.6  | 0.1  | 89.97 | 0.55 | 86.23 | 3.76  | 4.9  | 2.5  | 2.5  | 0.3  | 82.38 | 9.03  | 80.92 | 8.61  |
| wisconsin     | 7.57  | 9.0   | 0.9  | 1.4  | 0.2  | 98.63  | 0.22 | 96.8 | 2.38   | 4.0   | 0.4  | 1.3  | 0.2  | 97.74 | 0.28 | 96.21 | 2.26  | 1.0  | 0.0  | 1.0  | 0.0  | 92.90 | 0.49  | 92.75 | 3.54  |
| pima          | 10.77 | 10.4  | 2.0  | 2.5  | 0.1  | 82.36  | 0.91 | 75.5 | 4.47   | 6.3   | 1.2  | 2.4  | 0.2  | 80.22 | 1.03 | 73.93 | 3.98  | 1.5  | 0.8  | 2.1  | 0.6  | 71.36 | 4.51  | 70.54 | 6.26  |
| vehicle       | 17.13 | 22.4  | 2.7  | 2.8  | 0.1  | 84.50  | 1.40 | 70.6 | 4.68   | 13.3  | 1.9  | 2.7  | 0.1  | 78.83 | 1.87 | 70.25 | 3.20  | 5.4  | 2.0  | 2.7  | 0.2  | 59.48 | 8.17  | 56.60 | 7.90  |
| german        | 18.77 | 55.4  | 6.8  | 2.8  | 0.0  | 86.78  | 0.75 | 72.4 | 4.17   | 41.6  | 6.2  | 2.8  | 0.1  | 84.84 | 1.14 | 72.00 | 3.46  | 31.6 | 6.2  | 2.8  | 0.1  | 81.09 | 2.03  | 72.23 | 3.35  |
| contraceptive | 11.20 | 49.0  | 7.1  | 2.9  | 0.0  | 63.61  | 0.78 | 53.6 | 3.07   | 40.4  | 7.9  | 2.9  | 0.0  | 62.37 | 1.13 | 53.72 | 2.97  | 35.0 | 9.1  | 2.9  | 0.1  | 60.39 | 1.95  | 53.27 | 2.99  |
| titanic       | 4.27  | 10.4  | 1.0  | 1.8  | 0.1  | 78.95  | 0.30 | 78.7 | 1.67   | 2.4   | 0.6  | 1.8  | 0.3  | 78.00 | 0.38 | 77.80 | 2.25  | 1.0  | 0.0  | 1.9  | 0.3  | 77.48 | 0.33  | 77.53 | 1.85  |
| segment       | 13.73 | 26.2  | 2.9  | 2.5  | 0.1  | 97.97  | 0.31 | 96.6 | 0.89   | 18.0  | 2.9  | 2.4  | 0.1  | 96.69 | 0.90 | 95.64 | 1.44  | 11.8 | 3.5  | 2.3  | 0.2  | 83.55 | 8.74  | 82.58 | 9.14  |
| spambase      | 8.63  | 24.3  | 2.5  | 2.2  | 0.1  | 91.74  | 0.44 | 90.6 | 1.31   | 19.3  | 2.5  | 2.2  | 0.1  | 91.26 | 0.49 | 90.31 | 1.37  | 15.7 | 2.8  | 2.2  | 0.2  | 90.08 | 1.09  | 89.53 | 1.55  |
| banana        | 7.83  | 8.7   | 1.4  | 2.0  | 0.1  | 90.33  | 0.67 | 89.0 | 1.20   | 5.0   | 1.1  | 2.0  | 0.1  | 84.33 | 2.56 | 83.55 | 3.11  | 1.6  | 0.8  | 2.0  | 0.0  | 65.36 | 6.70  | 64.81 | 6.72  |
| phoneme       | 10.30 | 9.3   | 2.3  | 2.4  | 0.1  | 84.38  | 0.62 | 83.5 | 1.75   | 8.6   | 2.6  | 2.3  | 0.2  | 84.24 | 0.97 | 83.54 | 1.77  | 5.1  | 3.4  | 2.2  | 0.3  | 78.73 | 5.11  | 78.26 | 5.05  |
| page-blocks   | 14.30 | 21.5  | 3.0  | 2.4  | 0.1  | 97.84  | 0.20 | 97.0 | 0.58   | 13.6  | 2.4  | 2.2  | 0.1  | 97.37 | 0.26 | 96.69 | 0.60  | 7.1  | 2.9  | 2.1  | 0.3  | 94.89 | 2.40  | 94.63 | 2.99  |
| texture       | 22.13 | 60.5  | 9.0  | 2.8  | 0.0  | 97.50  | 0.23 | 95.2 | 1.05   | 41.8  | 3.9  | 2.8  | 0.1  | 95.58 | 1.65 | 93.84 | 1.69  | 28.0 | 7.5  | 2.7  | 0.1  | 85.22 | 9.05  | 84.04 | 8.89  |
| optdigits     | 22.60 | 135.3 | 28.7 | 3.0  | 0.0  | 97.02  | 0.57 | 93.4 | 1.02   | 95.2  | 16.9 | 3.0  | 0.0  | 94.21 | 2.23 | 91.42 | 2.17  | 68.7 | 18.0 | 3.0  | 0.0  | 87.26 | 6.30  | 85.34 | 6.08  |
| satimage      | 23.73 | 56.0  | 7.3  | 2.8  | 0.1  | 90.84  | 0.30 | 87.5 | 0.79   | 36.2  | 7.0  | 2.8  | 0.1  | 89.76 | 0.62 | 87.04 | 1.04  | 24.5 | 9.4  | 2.7  | 0.1  | 86.88 | 2.15  | 84.96 | 2.31  |
| thyroid       | 4.67  | 5.9   | 1.7  | 2.4  | 0.4  | 99.32  | 0.13 | 99.2 | 0.35   | 3.5   | 1.0  | 2.6  | 0.4  | 99.17 | 0.14 | 99.04 | 0.44  | 2.0  | 0.0  | 2.7  | 0.4  | 98.39 | 0.88  | 98.30 | 0.97  |
| ring          | 17.47 | 15.3  | 2.0  | 1.7  | 0.1  | 94.26  | 0.45 | 93.4 | 1.08   | 11.5  | 2.9  | 1.9  | 0.1  | 93.74 | 0.99 | 92.57 | 1.53  | 9.4  | 5.1  | 1.9  | 0.2  | 85.07 | 7.25  | 84.18 | 6.60  |
| twonorm       | 9.60  | 10.2  | 1.4  | 2.7  | 0.1  | 94.48  | 0.80 | 93.1 | 1.61   | 5.4   | 1.1  | 2.9  | 0.1  | 90.62 | 1.16 | 89.71 | 1.86  | 1.4  | 0.8  | 3.0  | 0.2  | 64.84 | 11.44 | 64.51 | 11.45 |
| coil2000      | 1.00  | 89.0  | 4.2  | 1.0  | 0.0  | 94.03  | 0.01 | 94.0 | 0.05   | 89.0  | 4.2  | 1.0  | 0.0  | 94.03 | 0.01 | 94.02 | 0.05  | 89.0 | 4.2  | 1.0  | 0.0  | 94.03 | 0.01  | 94.02 | 0.05  |
| penbased      | 15.72 | 119.2 | 20.7 | 2.9  | 0.0  | 97.42  | 0.92 | 96.2 | 0.56   | 105.7 | 17.2 | 2.9  | 0.0  | 94.63 | 2.53 | 93.67 | 2.13  | 84.9 | 20.9 | 2.9  | 0.0  | 89.09 | 6.69  | 88.35 | 6.51  |
| magic         | 8.30  | 32.2  | 3.7  | 2.6  | 0.1  | 86.35  | 0.40 | 85.4 | 0.55   | 26.7  | 5.2  | 2.6  | 0.1  | 86.09 | 0.43 | 85.28 | 0.59  | 23.3 | 6.9  | 2.6  | 0.1  | 85.04 | 2.06  | 84.56 | 2.04  |

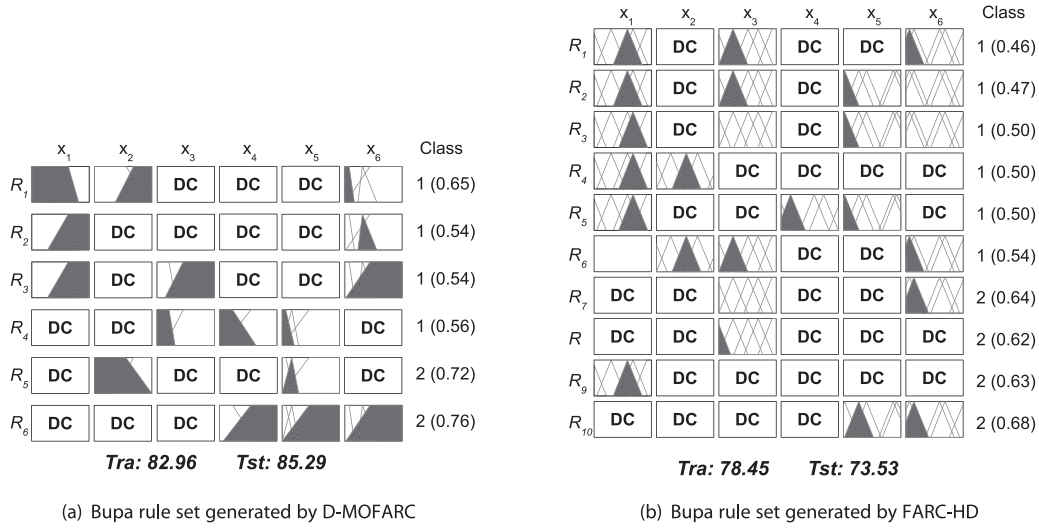


Fig. 3. Example rule sets obtained by the D-MOFARC (a) and the FARC-HD (b) approaches, respectively, when considering the third fold of the bupa dataset.

For each class, the algorithm selects the best rule, then the weights related to the patterns covered by this rule are decreased. In this way, the patterns that are still uncovered will have a greater possibility of being covered in the following iterations. When the  $i$  counter reach a threshold  $k_t$ , the correspondent pattern is deleted. The remaining rules are sorted again and the procedure is repeated until either all patterns have been deleted, or there is no rule left in the rule base. To evaluate the quality of fuzzy rules, a modification of the  $wWRAcc'$  measure described in [31] has been used. The  $wWRAcc'$  measure has been modified in order to handle fuzzy rules. The new measure is defined as follows:

$$wWRAcc''(A \rightarrow C_j) = \frac{n''(A \cdot C_j)}{n'(C_j)} \cdot \left( \frac{n''(A \cdot C_j)}{n''(A)} - \frac{n(C_j)}{N} \right) \quad (3)$$

where  $n''(A)$  is the sum of the products of the weights of all covered patterns by their matching degrees with the antecedent part of the rule,  $n''(A \cdot C_j)$  is the sum of the products of the weights of all correctly covered patterns by their matching degrees with the antecedent part of the rules,  $n(C_j)$  is the number of patterns of class  $C_j$  and  $n'(C_j)$  is the sum of the weights of patterns of class  $C_j$ .

Since the rule prescreening process can remove a large number of rules, after this step it is possible that some of the labels that belong to the original fuzzy partitions do not appear anymore in any of the remaining rules. Therefore, the initial DB needs to be refined by removing all the unnecessary labels, thus contributing to decrease the complexity of fuzzy partitions.

#### 4.2. Evolutionary multi-objective data base tuning and rule selection

In this step, the knowledge base previously obtained is improved by concurrently applying a tuning of the DB parameters and a rule selection process. A modification of the SPEA2 algorithm[13] is designed, with the aim improving the accuracy-complexity trade-off. In the following subsections the main features of this algorithm are presented. They are:

- Objectives.
- Coding scheme and initial gene pool.
- Crossover and mutation operators.

A further subsection is included at the end, which describes the main modifications introduced with respect to the classic SPEA2 algorithm.

#### 4.2.1. Objectives

Each chromosome is associated with a bi-dimensional vector, whose elements express the fulfillment degree of the following two objectives, respectively:

- Classification error minimization. It is represented by the complement of the number of the classification rate, i.e. the error rate. To compute this classification error, the following function has been used:

$$Fitness(C) = 1 - \frac{\#Hits}{N} \quad (4)$$

where  $\#Hits$  is the number of patterns correctly classified and  $N$  is the total number of patterns.

- Complexity minimization. It is represented by the number of selected rules.

#### 4.2.2. Coding scheme and initial gene pool

A double coding scheme for both rule selection ( $C_S$ ) and tuning ( $C_T$ ) is used:  $C^p = C_S^p C_T^p$ , where  $C^p$  is the chromosome representing the individual  $p$ . The  $C_S^p = (c_{S1}, \dots, c_{Sm})$  part is represented by a binary-coded string with  $m$  genes, where  $m$  is the number of initial rules. Each gene contains a value of “1” if the corresponding rule is selected, “0” otherwise. The  $C_T^p$  part uses a real coding scheme and codifies three definition parameters for each triangular MF.

$$C_T^p = C_1 C_2 \dots C_n$$

$C_i = (a_i^i, b_i^i, c_i^i, \dots, a_{m_i}^i, b_{m_i}^i, c_{m_i}^i)$   $i = 1, \dots, n$  where  $m_i$  is the number of labels in the database for each of the  $n$  variables.

Each triangular MF is represented as  $MF_j = (a_j, b_j, c_j)$ , where  $j = (1, \dots, k)$  and  $k$  is the granularity of the considered fuzzy partition. Each MF parameter can assume the values within the following variation intervals:

$$\begin{aligned} [l_{a_j}^i, r_{a_j}^i] &= [a_j - (b_j - a_j)/2, a_j + (b_j - a_j)/2] \\ [l_{b_j}^i, r_{b_j}^i] &= [b_j - (b_j - a_j)/2, b_j + (c_j - b_j)/2] \\ [l_{c_j}^i, r_{c_j}^i] &= [c_j - (c_j - b_j)/2, c_j + (c_j - b_j)/2] \end{aligned} \quad (5)$$

The first individual of the first population codifies the KB obtained by the previous step. The remaining individuals of the first population are generated randomly, with each value within the corresponding variation intervals.



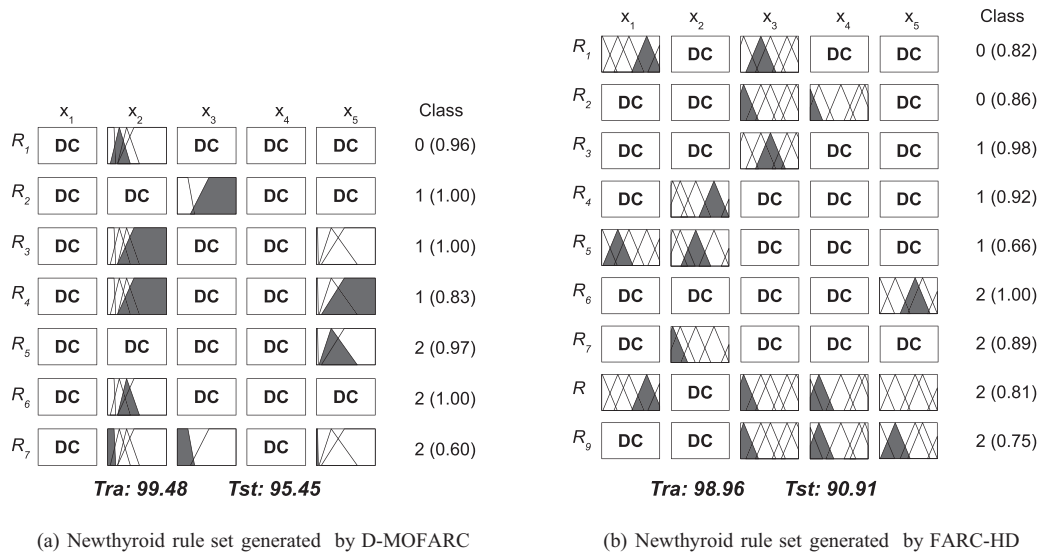


Fig. 4. Example rule sets obtained by the D-MOFARC (a) and the FARC-HD (b) approaches, respectively, when considering the third fold of the newthyroid dataset.

4.2.3. Crossover and mutation

Crossover and mutation operators have been specifically designed. Each offspring is obtained in the following way.

- First, the  $C_T$  part of the offspring is obtained by applying blend crossover (BLX)-0.5 [32] to the  $C_T$  part of the parents.
- Then, the binary part  $C_S$  is generated depending on the  $C_T$  parts of parents and offspring. For each gene in the  $C_S$  part, the following steps are performed:
  - Each gene of the  $C_T$  part which represents the corresponding MFs of the rule, is considered for both parents and offspring. The MFs of these three rules are extracted.
  - Between the offspring rule and each parent rule, euclidean normalized distances are computed by considering the center points of the MFs involved in these rules. The differences between each pair of centers are normalized by the amplitudes of their respective variation intervals.
  - The parent's rule closer to the offspring's rule is selected and its value is duplicated in the  $C_S$  part of the offspring.

This process is repeated until each gene in the  $C_S$  part of the offspring is obtained. In each step four offsprings are generated, although after applying mutation only the two best offspring are maintained. This type of crossover prevents the recovery of a bad rule already discarded, while permits the recovery of a rule that can be still considered good due to its MFs configuration.

The crossover operator performs a better exploration in the  $C_S$  part, therefore the mutation operator does not need to add rules. It simply changes randomly a gene value in the  $C_T$  part and sets to zero a random gene in the  $C_S$  part, with probability  $P_m$ .

The application of these operators brings some advantages: the crossover between individuals with very different rules allows the algorithm to explore different parts of the search space, while the mutation promotes rule extraction, since it is used to remove unnecessary rules.

4.2.4. Modifications of the classic SPEA2 algorithm

Some changes have been introduced to the original selection mechanism of SPEA2, to improve the algorithm's search ability.

- A mechanism to prevent incest has been included, based on the concepts of CHC presented in [33]. This avoid premature

convergence in the real coding ( $C_T$ ) part, which has a greater influence on the algorithm convergence and represents a wider search space than the binary coding part ( $C_S$ ). In the CHC approach, parents are crossed only if their Hamming distance divided by 4 exceeds a threshold. To follow this approach, the real coding scheme needs to be converted in a binary one, thus each gene is transformed using a gray code with a fixed number of bits per gene (BGene). The threshold value is initially set to  $L = (\#CT \times BGene) / 4$ , where  $\#CT$  is the number of genes in the  $C_T$  part of the chromosome. This value is decreased by 1 at each generation of the algorithm, therefore in further generations closer solutions can be crossed.

- A restart operator has been introduced to renew the external population when we detect that all the crossover are allowed. Actually, to prevent premature convergence, the first restart is applied if 50% of crossovers are detected at any generation (the required ratio can be defined as  $\%_{required} = 0.5$ ). Each time the restart is performed, the required ratio is updated as follows:  $\%_{required} = (1 + \%_{required}) / 2$ .

The external population after the restart includes the individuals with the best value in each objective, and the remaining individuals are initialized as follows: the  $C_S$  part is copied from the most accurate individual, while the values in the  $C_T$  part are generated randomly. In this way, the most accurate and interpretable solutions obtained so far are preserved.

Some constraints to the application of restart have been introduced: (a) a new restart cannot be applied if the most accurate solution has not been improved; (b) the restart is not applied at the end, when the approximation of the Pareto front is well formed and needs to be preserved; (c) restart is disabled if the midpoint of the total evaluations number is reached and it has been never applied before.

- A mechanism to promote the most accurate solutions has been introduced. At each stage of the algorithm, between restarting points, the number of solutions in the external population ( $\bar{P}_{t+1}$ ) that can be used to constitute the mating pool is reduced progressively and the most accurate solutions are preferred. To this end, solutions are sorted according their accuracy and the number of eligible solutions is reduced progressively from 100% at the beginning to 50% at the end of each stage. This mechanism is disabled in the last evaluations (when restart is disabled too), in order to obtain a wide and well-formed Pareto front.

## 5. Experimental framework

The method proposed and described in the previous sections has been evaluated by comparing its results with the results obtained by applying a recent and well-performing GFS for classification problems, named a Fuzzy Association Rule-Based Classification model for High Dimensional problems proposed in [12]. This algorithm currently belongs to the state-of-the-art of classification algorithms, since it has been demonstrated to outperform in accuracy some of the most widespread classification algorithms. In particular, the results presented in [12] show that it performs better than three other GFSs (FH-GBML, 2SLAVE, SGERD), two approaches to obtain fuzzy associative classifiers (LAFAR and CFAR) and five classic approaches for associative classification (C4.5, CBA, CBA2, CMAR, CPAR).

This algorithm has been demonstrated to generate good fuzzy rule-based models, in particular when dealing with high dimensional problems.

### 5.1. Experimental setup

The experiments have been performed by considering 35 real-world datasets, whose characteristics are described in Table 2. The number of instances (**#Inst**), the number of attributes (**#Attr**) (numerical and nominal attributes **Num/Nom** are highlighted, respectively) and the number of classes (**#Cls**) of each dataset are shown. The web link to the Knowledge Extraction based on Evolutionary Learning (KEEL)-data set repository is also reported [34], from which the datasets can be downloaded. The instances that presented missing values have been removed from the datasets (in particular, from cleveland and crx datasets).

To carry the different experiments out, a ten-fold cross-validation model has been applied: each dataset has been randomly split into ten folds, each containing 10% of the patterns of the dataset. Then, a single fold has been used for testing and the remaining folds for training. The cross-validation process has been repeated ten times, with each fold used exactly once for testing. For each of the ten partitions, three trials of the algorithm have been executed and finally the results have been averaged out over 30 runs. Due to the multi-objective nature of the evolutionary algorithm included in the D-MOFARC method, for the comparison with the single-objective FARC-HD approach we have selected the most accurate solution by considering the training error and averaging it on all the Pareto fronts obtained by each run of the algorithm.

To evaluate the results, statistical analysis has been adopted [35,36], in particular non-parametric tests, following the recommendations presented in [37], where a set of simple and robust non-parametric tests for statistical comparisons of FRBCSs has been described. The Wilcoxon's signed-ranks test [38,39] for pairwise comparison has been used, considering a confidence level of  $\alpha = 0.05$ . A wider description of this test and a software to perform it can be found on the web site available at: <http://sci2s.ugr.es/sicidm/>.

### 5.2. Results and analysis for the best solutions

This section shows the results of the experiments described in the previous section. Table 3 summarizes the average number of rules/conditions (**#R/#C**) and classification percentages in training (**Tra**) and test (**Tst**) of the most accurate solution from each of the obtained Pareto fronts, for the D-MOFARC approach and of the best solution for FARC-HD [12]. The overall mean values for each method are highlighted in the last row.

The two methods have been compared by applying the Wilcoxon's signed-ranks test, in order to understand if they are

statistically equivalent (null-hypothesis). When considering the accuracy, the Wilcoxon's test is based on computing the differences between the average errors on the test set, whereas when considering the complexity the test is computed by taking into account the average number of rules that are obtained by a pair of algorithms. A normalized difference *DIFF* has been adopted when considering the number of rules, which is defined as

$$DIFF = \frac{MeanRules(x) - MeanRules(RA)}{MeanRules(x)} \quad (6)$$

where *MeanRules(x)* represents the number of rules obtained on average by the *x* algorithm (D-MOFARC) and *RA* is the reference algorithm (FARC-HD). This difference expresses the improvement in percentage with respect to the reference algorithm.

Table 4 shows the statistics obtained by applying the Wilcoxon's signed-rank test on the accuracy achieved on the test set and on the number of rules, comparing the results obtained by applying the D-MOFARC approach with the results achieved by using the FARC-HD algorithm. The ranks  $R^+$  and  $R^-$  and the *p*-value are shown and a further column is added to highlight if the null-hypothesis is rejected or not, with a significance level  $\alpha = 0.05$ .

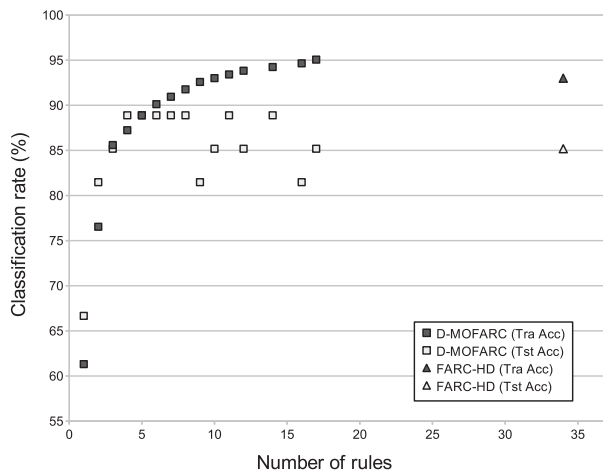
When considering the accuracy on test, the *p* value  $> \alpha$ , therefore the test rejects the null-hypothesis of equivalence. The  $R^+$  and  $R^-$  values highlight that the results of the proposed method outperform the results obtained by the FARC-HD algorithm. A further comparison has been drawn between the two methods with respect to the average number of rules of the most accurate solutions. In this case, the null-hypothesis is not rejected but the two approaches can be considered statistically different with a significance level of 0.065, i.e., a 93.5% confidence. Nevertheless, by looking at the average values presented in Table 3 we can notice that the average number of rules obtained by applying the D-MOFARC method is slightly lower than the average value obtained by applying the FARC-HD algorithm.

Fig. 3 presents a representative example of the rule sets obtained by the D-MOFARC (a) and the FARC-HD (b) approaches, respectively, when considering the third fold of the bupa dataset. As well, Fig. 4 represents the rule sets obtained by the D-MOFARC (a) and the FARC-HD (b) approaches, respectively, when considering the third fold of the newthyroid dataset. In these specific cases, the figures highlight that the proposed method induces the generation of more compact rule sets and smaller granularities with respect to the FARC-HD algorithm.

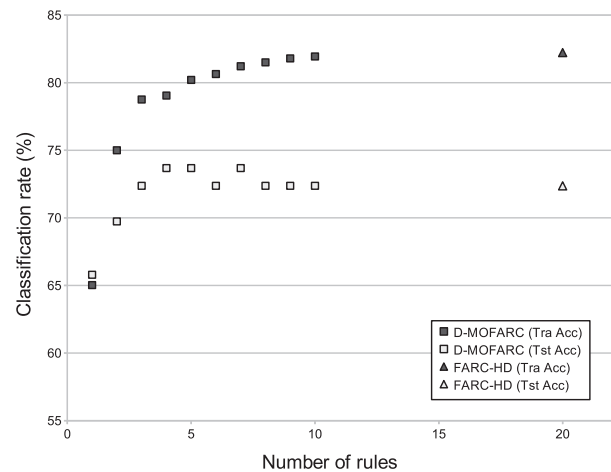
### 5.3. Results and analysis for the most representative solutions of the pareto front approximations

Although the results presented so far only consider the best solution (considering the training error) obtained by the D-MOFARC method, it actually generates a set of Pareto-optimal solutions, each of them representing a trade-off between accuracy and complexity. Even though it was not our main purpose, it is possible for a decision maker to choose the solution that better satisfies the required trade-off for a certain problem.

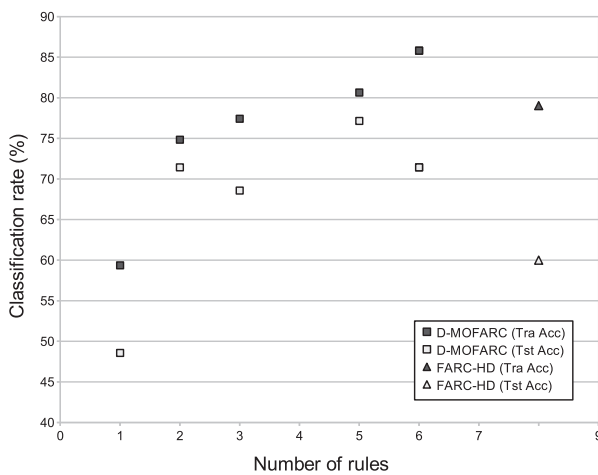
To this aim, we have used a procedure already adopted in [40,41]. This procedure is based on the analysis of three representative solutions of the Pareto front approximations, i.e. the most accurate solution, called FIRST, the least complex solution, denoted as LAST, and the median solution between the FIRST and the LAST. To compute these solutions, for each of the 30 trials we have generated the Pareto front approximations and have ordered the solutions in each approximation from the most to the least accurate. Then, for each approximation, we have selected the most accurate solution (FIRST), the intermediate solution in the Pareto (MEDIAN) and the least complex solution (LAST). Finally, for the FIRST, MEDIAN and LAST solutions, we have computed



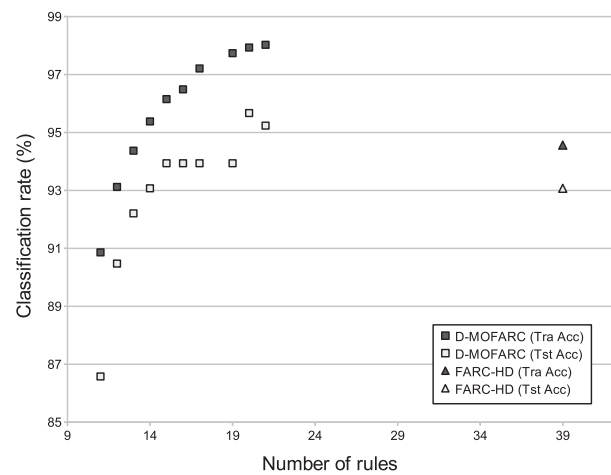
(a) Heart dataset (fold 8)



(b) Pima dataset (fold 8)



(c) Bupa dataset (fold 1)



(d) Segment dataset (fold 7)

**Fig. 5.** Pareto solutions obtained by the D-MOFARC method when considering representative folds of four datasets, namely heart (a), pima (b), bupa (c) and segment (d).

the mean values and the standard deviation values over the 30 trials, considering the number of rules, the number of conditions, the accuracy of the training set and the accuracy of the test set. In Table 5 we have reported the results described so far. In the second column, we have also reported the cardinality values of the obtained Pareto front approximations, averaged over 30 runs.

In addition, we have represented some Pareto-optimal sets in Fig. 5, in particular for heart, bupa, pima and segment datasets, respectively. Each Pareto front has been drawn by choosing a representative fold of the considered dataset and reporting the number of rules in the x-axis and the correct classification rate on training in the y-axis. The Pareto-optimal solutions are represented with dark square symbols in the graphs. In addition, the correspondent values for the correct classification rate on test are depicted with light square symbols. We have also depicted the correct classification rate on training (dark triangle symbols) and test (light triangle symbols) of the solution obtained by the application of the FARC-HD algorithm on the same folds. The observation of the Pareto fronts points out that in some cases there exist solutions with a test accuracy level almost equivalent to the best solution's test accuracy, while their complexity is considerably lower than the best solution's complexity.

## 6. Conclusion

In this proposal we have presented a multi-objective evolutionary method which performs concurrently a tuning process and a rule selection process on an initial KB of a FRBCS. A Fuzzy Discretization Algorithm has been designed to extract suitable granularities from data and to generate the fuzzy partitions that constitute the initial DB. The associated RB has been generated by extracting a set of fuzzy association rules, according to the first two steps of the FARC-HD method presented in [12]. The proposed MOEA is a modification of the SPEA2 algorithm and aims at generating compact and precise FRBCSs. To this end, two objectives have been considered at the same time in the evolutionary process, namely the correct classification rate and the number of rules of the obtained models.

The experiments have been performed on 35 real-world datasets and the obtained results have been compared with the results achieved by the execution of a recent and well performing accuracy-driven GFS, namely FARC-HD. The Wilcoxon's test applied on the average results of test accuracy has highlighted that the proposed method clearly outperforms the FARC-HD, whereas less significant statistical difference has been observed when considering the complexity of the obtained models. Nevertheless, the mean results show that the number of rules is also reduced on average.

## References

- [1] H. Ishibuchi, T. Nakashima, M. Nii, Classification and Modeling with Linguistic Information Granules: Advanced Approaches Advanced Approaches to Linguistic Data Mining, Springer, London, 2005.
- [2] O. Cordon, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific Publishing, Singapore, 2001.
- [3] F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, *Evol. Intell.* 1 (2008) 27–46.
- [4] J. Casillas, F. Herrera, O. Cordon, L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing, Springer, Secaucus, NJ, USA, 2003.
- [5] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Inc., New York, USA, 2001.
- [6] C.A.C. Coello, G.B. Lamont, D.A.V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer-Verlag, Secaucus, NJ, USA, 2007.
- [7] M. Fazzolari, R. Alcalá, Y. Nojima, H. Ishibuchi, F. Herrera, A review of the application of multiobjective evolutionary fuzzy systems: current status and further directions, *IEEE Trans. Fuzzy Syst.* 21 (1) (2013) 45–65.
- [8] O. Cordon, F. Herrera, P. Villar, Analysis and guidelines to obtain a good uniform fuzzy partition granularity for Fuzzy Rule-Based Systems using simulated annealing, *Int. J. Approx. Reason.* 25 (2000) 187–215.
- [9] M.J. Gacto, R. Alcalá, F. Herrera, Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems, *Soft Comput.* 13 (5) (2009) 419–436.
- [10] U.M. Fayyad, K.B. Irani, On the handling of continuous-valued attributes in decision tree generation, *Mach. Learn.* 8 (1992) 87–102.
- [11] J. Catlett, On changing continuous attributes into ordered discrete attributes, in: Y. Kodratoff (Ed.), Machine Learning – EWSL91, Lecture Notes in Computer Science, vol. 482, Springer, Berlin / Heidelberg, 1991, pp. 164–178.
- [12] J. Alcalá-Fdez, R. Alcalá, F. Herrera, A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning, *IEEE Trans. Fuzzy Syst.* 19 (5) (2011) 857–872.
- [13] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength pareto evolutionary algorithm, *Tech. Rep.* 103 (2001).
- [14] L.A. Kurgan, K.J. Cios, Caim discretization algorithm, *IEEE Trans. Knowl. Data Eng.* 16 (2) (2004) 145–153.
- [15] O. Cordon, F. Herrera, P. Villar, Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base, *IEEE Trans. Fuzzy Syst.* 9 (4) (2001) 667–674.
- [16] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms, *IEEE Trans. Fuzzy Syst.* 3 (3) (1995) 260–270.
- [17] H. Ishibuchi, T. Murata, I.B. Türksen, Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems, *Fuzzy Sets Syst.* 89 (2) (1997) 135–150.
- [18] H. Ishibuchi, T. Yamamoto, Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, *Fuzzy Sets Syst.* 141 (1) (2004) 59–88.
- [19] R. Alcalá, J.R. Cano, O. Cordon, F. Herrera, P. Villar, I. Zwir, Linguistic modeling with hierarchical systems of weighted linguistic rules, *Int. J. Approx. Reason.* 32 (2-3) (2003) 187–215.
- [20] M. Antonelli, P. Ducange, B. Lazzarini, F. Marcelloni, Learning concurrently partition granularities and rule bases of Mamdani fuzzy systems in a multi-objective evolutionary framework, *Int. J. Approx. Reason.* 50 (7) (2009) 1066–1080.
- [21] M. Antonelli, P. Ducange, B. Lazzarini, F. Marcelloni, Multi-objective evolutionary learning of granularity, membership function parameters and rules of Mamdani fuzzy systems, *Evol. Intell.* 2 (1-2) (2009) 21–37.
- [22] M. De Vega, J. Bardallo, F. Marquez, A. Peregrin, Parallel distributed two-level evolutionary multiobjective methodology for granularity learning and membership functions tuning in linguistic fuzzy systems, in: International Conference on Intelligent Systems Design and Applications, 2009, pp. 134–139.
- [23] R. Alcalá, Y. Nojima, F. Herrera, H. Ishibuchi, Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions, *Soft Comput.* 15 (2011) 2303–2318.
- [24] O. Castillo, P. Melin, W. Pedrycz, Design of interval type-2 fuzzy models through optimal granularity allocation, *Appl. Soft Comput.* 11 (8) (2011) 5590–5601.
- [25] C.H. Nguyen, W. Pedrycz, T.L. Duong, T.S. Tran, A genetic design of linguistic terms for fuzzy rule based classifiers, *Int. J. Approx. Reason.* 54 (1) (2013) 1–21.
- [26] M.A. Sanchez, O. Castillo, J.R. Castro, An analysis on the intrinsic implementation of the principle of justifiable granularity in clustering algorithms, in: O. Castillo, P. Melin, J. Kacprzyk (Eds.), Recent Advances on Hybrid Intelligent Systems, Studies in Computational Intelligence, vol. 451, Springer, Berlin/Heidelberg, 2013, pp. 121–134.
- [27] W. Pedrycz, The principle of justifiable granularity and an optimization of information granularity allocation as fundamentals of granular computing, *J. Inf. Process. Syst.* 7 (3) (2011) 397–412.
- [28] Y. Yoshinari, W. Pedrycz, K. Hirota, Construction of fuzzy models through clustering techniques, *Fuzzy Sets Syst.* 54 (2) (1993) 157–165.
- [29] C. Zhang, S. Zhang, Association Rule Mining: Models and Algorithms, Springer-Verlag, Berlin, 2002.
- [30] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2006.
- [31] B. Kavšek, N. Lavrač, V. Jovanoski, APRIORI-SD: adapting association rule learning to subgroup discovery, in: M.R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, C. Borgelt (Eds.), Advances in Intelligent Data Analysis V, Lecture Notes in Computer Science, vol. 2810, Springer, Berlin/Heidelberg, 2003, pp. 230–241.
- [32] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: Foundation of Genetic Algorithms 2, 1993, pp. 187–202.
- [33] L.J. Eshelman, The chc adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, in: FOGA'90, 1990, pp. 265–283.
- [34] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, *J. Mult. Valued Log. Soft Comput.* 17 (2-3) (2011) 255–287.
- [35] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, *J. Mach. Learn. Res.* 9 (2008) 2677–2694.
- [36] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, *Soft Comput.* 13 (2009) 959–977.
- [37] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (2006) 1–30.
- [38] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 3rd ed., Taylor & Francis Group, Boca Raton, 2004.
- [39] F. Wilcoxon, Individual comparisons by ranking methods, *Biom. Bull.* 1 (6) (1945) 80–83.
- [40] R. Alcalá, P. Ducange, F. Herrera, B. Lazzarini, F. Marcelloni, A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems, *IEEE Trans. Fuzzy Syst.* 17 (5) (2009) 1106–1122.
- [41] M. Antonelli, P. Ducange, F. Marcelloni, An efficient multi-objective evolutionary fuzzy system for regression problems, *Int. J. Approx. Reason.* 54 (9) (2013) 1434–1451.