

# Obtaining a linguistically understandable random sets-based classifier from interval-valued data with genetic algorithms

Luciano Sánchez, Inés Couso  
Universidad de Oviedo, Spain

## Abstract

*Combining descent algorithms and a coevolutionary scheme, we have defined a new procedure that is able to obtain rule-based models from datasets with censored or interval-valued data, and can also identify the conflictive instances in the training set: those that contribute the most to the indetermination in the likelihood of the model.*

## 1 Introduction

When statistical classifiers and fuzzy rule-based classifiers are compared, we often do not know to what extent the difference in performance is intrinsic to the dataset (because the decision surface is too complex for being representable by a compact set of rules [5]) or the learning algorithm is accountable because it has not found the best knowledge base (KB). To help with this decision, in former works we have proposed a random sets rule based system (RSRBS) that, under certain conditions, is numerically equivalent to a fuzzy rule based system (FRBS) [18, 19]. RSRBSs can be estimated from data with deterministic descent algorithms. Comparing its performance to a GFS, we can find those cases where a GFS has not properly converged to the best KB.

In this paper we have extended RSRBSs to interval data. The new method can be used to learn fuzzy classifiers from data, however its primary purpose will be to establish a threshold in the performance of FRBS from imprecise data as we did with crisp datasets in [19].

There are, however, numerical difficulties when obtaining a RSRBS from interval data, because the objective function is not completely known [17]. In previous works we have proposed evolutionary schemes guided to obtain non-dominated sets of bounds of the objective function [20, 21], that could be used to solve this problem. In this work, however, we propose a different coevolutionary scheme [15] that is able to produce not only a nondominated linguistically understandable classifier, but also the list of the instances of the training set that contribute the most to the

lack of knowledge about the fitness of the classifier.

This paper is structured as follows: in Section 2 we recall and update the definition of RSRBS introduced in [19], and its similarities with a FRBS. In Section 3 we state the maximum likelihood estimate of a RSRBS from crisp data, and discuss how to extend the problem to imprecise data. In Section 4, we discuss a coevolutionary genetic algorithm that solves the problem, and in Section 5 we provide compared numerical results. The paper finishes with the concluding remarks in Section 6.

## 2 A random-set based linguistically understandable classifier

Let  $C \in \{1, \dots, N_c\}$  be the class labels,  $x = (x^1, \dots, x^n)$ , the features with which we perceive an object, and let  $X$  be the input space,  $x \in X = X^1 \times \dots \times X^n$ . Lastly, let the Bayes minimum error classifier be

$$\text{class}(x_0) = \arg \max_i P(c = i | x = x_0). \quad (1)$$

We will consider that a rule-based classifier is a parametric model of  $P(c | x)$  which has a specific human-readable form. In this section we will develop a statistical model that relates that linguistically understandable form, based on fuzzy logic, to abstract concepts of classification theory.

### 2.1 Crisp sets-based model

Let us define first a crisp parametric model as a partition  $\{A_1, \dots, A_N\}$  of the input space  $X$  and a matrix

$$M = \begin{pmatrix} p_{11} & \dots & p_{1N_c} \\ \vdots & \ddots & \vdots \\ p_{M1} & \dots & p_{MN_c} \end{pmatrix} \quad (2)$$

where

$$p_{ic} = P(c | A_i). \quad (3)$$

Given the matrix  $\mathbf{M}$  and an input  $x_0$ , we can compute

$$P(c = C_0 | x) = \quad (4)$$

$$= \sum_{i=1}^N P(c = C_0 | A_i) P(A_i | x) \quad (5)$$

$$= \sum_{i=1}^N p_{iC_0} I_{A_i}(x) \quad (6)$$

where  $I_{A_i}(x)$  is either 1 or 0 if  $x \in A_i$  or  $x \notin A_i$ , respectively.

In addition, if we impose that each element of this last partition is decomposable,

$$A_i = A_i^1 \times \dots \times A_i^n, \quad A_i^j \subset X^j \quad (7)$$

then the model is linguistically understandable, because to each element of  $\mathbf{M}$  can be assigned a linguistic rule, as follows:

$$\text{if } x_1 \text{ is } A_i^1 \text{ and } \dots \text{ and } x_n \text{ is } A_i^n \text{ then class } C_i \text{ with } p_i. \quad (8)$$

For example, let  $X^1 = [0, 1]$  be the domain of the weights of a collection of objects, and  $X^2 = [1, 2]$  the domain of their lengths, thus  $X = [0, 1] \times [1, 2]$ . Let  $\{\text{SMALL}, \text{LARGE}\}$  with  $\text{SMALL} = [0, 0.5]$ ,  $\text{LARGE} = [0.5, 1]$  a linguistic partition of  $X^1$ , and  $\{\text{SHORT}, \text{LONG}\}$  with  $\text{SHORT} = [1, 1.5]$ ,  $\text{LONG} = [1.5, 2]$  a linguistic partition of  $X^2$ . Lastly, consider the rule that follows:

$$\text{if } x_1 \text{ is SMALL and } x_2 \text{ is SHORT then class 2 with 0.8.} \quad (9)$$

The information provided by this rule is

$$P(c = 2 | x \in [0, 0.5] \times [1, 1.5]) = 0.8 \quad (10)$$

## 2.2 Random sets-based model

We will define a random sets rule-based system (RSRBS) by means of a family of crisp models, indexed by a parameter  $\theta \in \Theta$ , and a probability distribution in  $\Theta$ . Each model in this family shares the same matrix  $\mathbf{M}$  and depends on a partition  $\{A_1^\theta, \dots, A_N^\theta\}$  of the input space  $X$ . To classify an input value  $x$ , we average the outputs of all the crisp models in the family:

$$P(c = C_0 | x) = \int_{\Theta} \left( \sum_{i=1}^N p_{iC_0} I_{A_i^\theta}(x) \right) dP_\theta \quad (11)$$

$$= \sum_{i=1}^N p_{iC_0} \int_{\Theta} I_{A_i^\theta}(x) dP_\theta \quad (12)$$

$$= \sum_{i=1}^N p_{iC_0} \Phi_i(x) \quad (13)$$

where  $\Phi_i(\cdot)$  is the one point coverage function of the random set  $A_i^\theta$ , i.e.  $\Phi_i(x) = P(x \in A_i^\theta)$ .

Let  $A_i^\theta = A_i^{1\theta} \times \dots \times A_i^{n\theta}$ ; in case the random variables  $I_{A_i^{j\theta}}(x)$  are independent, then

$$\Phi_i(x) = \prod_{j=1}^n \Phi_i^j(x_j) \quad (14)$$

where

$$\Phi_i^j(x_j) = \int_{\Theta} I_{A_i^j}^\theta(x) dP_\theta \quad (15)$$

are one point coverage functions of random sets defined on the variables  $X^j$ .

## 2.3 Relationship between the random set-based model and a FRBS

According to [2], the one point coverage function of a random set can be understood as a fuzzy membership function. If the functions  $\Phi_i$  are regarded as membership functions,

$$\begin{aligned} \sum_{i=1}^M \Phi_i(x) &= \sum_{i=1}^M \int_{\Theta} I_{A_i^\theta}(x) dP_\theta \\ &= \int_{\Theta} \left( \sum_{i=1}^M I_{A_i^\theta}(x) \right) dP_\theta \\ &= \int_{\Theta} dP_\theta = 1 \end{aligned} \quad (16)$$

thus they form a Ruspini's fuzzy partition of  $X$  [16]. That is to say, the linguistic information of a random sets-based model is compatible with that of a fuzzy model. The inference procedure is similar, too. The inference in a FRBS comprising rules "if  $\tilde{A}_k$  then  $C_k$  with  $w_k$ " is:

$$\text{class}(x) = \arg \max_c \left\{ \bigvee_{i:C_i=c} \left( \bigwedge_j \tilde{A}_i^j(x) \wedge w_i \right) \right\} \quad (17)$$

and the same process in a RSRBS composed by the same linguistic rules (i.e.  $\Phi(x) = \tilde{A}(x)$ ) produces

$$\text{class}(x) = \arg \max_c \left\{ \sum_{i:C_i=c} \left( \prod_j \Phi_i^j(x) \cdot w_i \right) \right\} \quad (18)$$

that is to say, the RSRBS is a particular case of fuzzy classifier where  $\sum \tilde{A}(x) = 1$  for any  $x$ , and voting-based inference [8] and product t-norm are used. In the remainder of this paper we will use the type-III fuzzy rules; each group of  $N_c$  random set-based rules like

if  $\Phi_k$  then  $C_1$  with  $w_{k1}$   
 $\vdots$   
if  $\Phi_k$  then  $C_{N_c}$  with  $w_{kN_c}$

will be written

“if  $\tilde{A}_k$  then  $C_1$  with  $w_{k1}$  and  $\dots$  and  $C_{N_c}$  with  $w_{kN_c}$ ”.

### 3 Estimation of a RSRBS from data: optimality conditions

Assuming the linguistic partitions are predetermined, learning a RSRBS from data consists in inferring the matrix  $\mathbf{M}$  from a sample. In this section we will define a set of necessary conditions, that the weights of the rules in RSRBS must fulfill after the training process.

**Lemma 3.1.** *Let a RSRBS comprising  $N$  linguistic rules*

“if  $\tilde{A}_k$  then  $C_1$  with  $w_{k1}$  and  $\dots$  and  $C_{N_c}$  with  $w_{kN_c}$ ”.

*Given a sample of data  $\{(\mathbf{x}_s, \mathbf{c}_s)\}_{s=1, \dots, N_s}$ , the best assignment of weights fulfills that*

$$\sum_{s:\mathbf{c}_s=a} \frac{\tilde{A}_i(\mathbf{x}_s)}{\sum_{k=1}^N \tilde{A}_k(\mathbf{x}_s)w_{kc_s}} = \sum_{s:\mathbf{c}_s=b} \frac{\tilde{A}_i(\mathbf{x}_s)}{\sum_{k=1}^N \tilde{A}_k(\mathbf{x}_s)w_{kc_s}} \quad \text{for all } a, b \in \{1, \dots, N_c\}, i = 1, \dots, N \quad (19)$$

*Proof.* The likelihood of the RSRBS is

$$L(\mathbf{M}) = \sum_{s=1}^{N_s} \log \sum_{i=1}^N \Phi_i(\mathbf{x}_s) p_{i\mathbf{c}_s} \quad (20)$$

and there are  $N$  constraints

$$\sum_{c=1}^{N_c} p_{ic} = 1. \quad (21)$$

We convert the constrained problem into an unconstrained one with the help of  $N$  Lagrange multipliers,

$$L'(\mathbf{M}) = \sum_{s=1}^{N_s} \log \sum_{i=1}^N \Phi_i(\mathbf{x}_s) p_{i\mathbf{c}_s} + \sum_{i=1}^N \lambda_i \left(1 - \sum_{c=1}^{N_c} p_{ic}\right). \quad (22)$$

Taking derivatives with respect to  $p_{ic}$  and  $\lambda_i$ , we obtain the following conditions

$$\sum_{s:\mathbf{c}_s=c} \frac{\Phi_i(\mathbf{x}_s)}{\sum_{k=1}^N \Phi_k(\mathbf{x}_s) p_{kc_s}} = \lambda_i \quad (23)$$

for  $i = 1, \dots, N; c = 1, \dots, N_c$

$$\sum_{c=1}^{N_c} p_{ic} = 1, \quad \text{for } i = 1, \dots, N \quad (24)$$

thus eq. 19 fulfills.  $\square$

```

normalize ( $\mathbf{X} \in \mathbf{R}^{N \times N_c}$ )
    if ( $\mathbf{X}_{ic} < 0$ )  $\mathbf{X}_{ic} = 0$ 
     $\mathbf{X}_{ic} = \mathbf{X}_{ic} / \sum_{d=1}^{N_c} \mathbf{X}_{id}$ 
end of normalize

minimize ( $\mathbf{M} \in \mathbf{R}^{N \times N_c}$ , selected  $\in \{0, 1\}^N$ )
 $\lambda, \mathbf{D} \in \mathbf{R}^{N \times N_c}$ ,  $\alpha \in \mathbf{R}$ ,  $c \in 1 \dots N_c$ ,  $i \in 1 \dots N$ 
 $\mathbf{M}_{ic} = 1/N_c$ 
repeat
     $\lambda_{ic} = \sum_{s:\mathbf{c}_s=c} \Phi_i(\mathbf{x}_s) / \sum_{k=1}^N \Phi_k(\mathbf{x}_s) \mathbf{M}_{kc_s}$ 
    if selected[i] then  $\mathbf{D}_{ic} = \lambda_{ic} - N_c^{-1} \sum_{c=1}^{N_c} \lambda_{ic}$ 
    else  $\mathbf{D}_{ic} = 0$ 
    Brent search of  $\alpha$  that minimizes
         $L(\text{normalize}(\mathbf{M} + \alpha \cdot \mathbf{D}))$ 
     $\mathbf{M} = \text{normalize}(\mathbf{M} + \alpha \cdot \mathbf{D})$ 
until  $\alpha \|D\| < \epsilon$ 
end of minimize

```

**Figure 1. Pseudocode of the numerical algorithm used to solve the set of equations (24).**

In Figure 1 we propose to use a deterministic algorithm for obtaining an approximated solution, that combines solving these  $N(N_c + 1)$  nonlinear equations with a descent step based on a Brent linear search [12], combined with a projection of the search direction in the feasible space. The parameter called “selected” in this function allows us to select which rows of  $\mathbf{M}$  intervene in the optimization problem. The unselected rows will end up with weights equal to  $1/N_c$  for all classes, thus the corresponding rules vote the same for all classes and can be removed. This parameter allows us to guide the search of a compact rulebase with a genetic algorithm, as we will show later.

### 3.1 Interval data

Let us assume that the input data cannot be precisely observed, but we perceive intervals that contain them. This include, for instance, inexact measurements, censored data and missing values. In the most general case, we have a sample  $\{(\Gamma_s, \mathbf{c}_s)\}_{s=1, \dots, N_s}$  where  $\Gamma_s = [x_{1s}^-, x_{1s}^+] \times \dots \times [x_{ns}^-, x_{ns}^+]$  is an interval of  $\mathbf{R}^n$ .

The likelihood of the RSRBS is, in this case, the set

$$[L^-(\mathbf{M}), L^+(\mathbf{M})] = \left\{ \sum_{s=1}^{N_s} \log \sum_{i=1}^N \Phi_i(\mathbf{g}_s) p_{i\mathbf{c}_s} \mid \mathbf{g}_s \in \Gamma_s \right\} \quad (25)$$

with the same  $N$  constraints as before. However, the optimization problem is considerably more complex, as we can no longer determine an unique set of weights which is the best for the whole range of the inputs. We can, at the most,

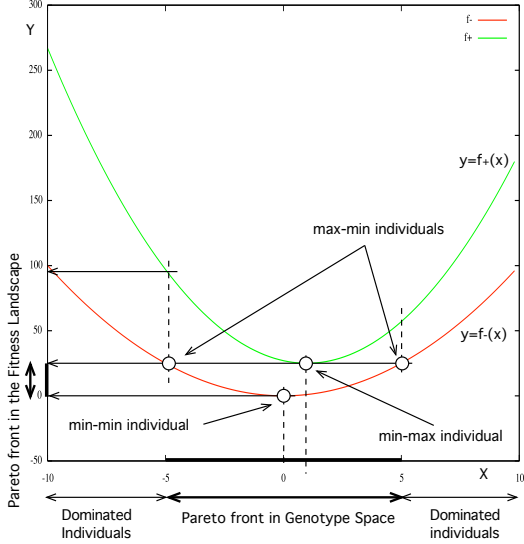


Figure 2. Interval optimization

find the smallest set of nondominated matrices that contains the solution:

$$\{M \mid \nexists M' \text{ s.t. } L^+(M') \leq L^-(M)\}. \quad (26)$$

### 3.1.1 Genetic algorithms and interval-valued optimization

Genetic algorithms are one of the best suited numerical methods for this kind of search, that is closely related to multicriteria optimization [10]. Let us clarify, with the help of an example, the genetic problem we need to solve: in Figure 2 we depict a case where we want to find the minimum  $x_0$  of a partially known function  $f$ , that lies between  $f^-$  and  $f^+$ . We know that the value of the objective function in the minimum,  $f(x_0)$ , is in the segment we have labelled “Pareto front in the fitness landscape”. In turn,  $x_0$  in the area marked “Pareto front in Genotype Space”.

In previous works, we have designed multicriteria genetic algorithms whose fitness function is an interval or fuzzy. Those algorithms produce sets of individuals contained in the Pareto front in the genotype space [20]. However, in this paper we will follow a new, different approach. We want to identify those “max-min” and “min-min” individuals in Figure 2. In particular, we want to identify two crisp samples  $\{x_s^{\text{low}}, c_s\}$  and  $\{x_s^{\text{high}}, c_s\}$ , with  $x_s^{\text{low}}, x_s^{\text{high}} \in \Gamma_s$ , such that the models obtained from these two samples (by applying the algorithm introduced in the preceding section) score both extrema  $L^-$  and  $L^+$  of the Pareto front in the Fitness Landscape. In the next section we propose a co-evolutionary Genetic Algorithm that provides a solution to this problem.

## 4 A proposal of coevolutionary learning of RSRBS from vague data

The algorithm that we propose depends on three populations. The first one contains different model candidates  $M$  (each individual represents a model, Pitts style [9]), and the other two codify the crisp samples  $\{x_s^{\text{low}}\}$  and  $\{x_s^{\text{high}}\}$  mentioned in the preceding section. In these two last populations, each individual represents one point in the sample, and the whole population is the solution (cooperative approach [7]). These three populations coevolve to find the best RSRBS and the extrema  $\{x_s^{\text{low}}\}$ ,  $\{x_s^{\text{high}}\}$ , as described in this section.

### 4.1 Representation of an individual

Each model in the first population can be univocally represented with a binary vector, that was called “selected” in Figure 1. This vector stores the set of rows of the matrix  $M$  whose terms are different from  $1/N_c$ ; in other words, if a bit is set to 1 then we emit the rule whose antecedent is associated to the position of the bit. Observe that this vector can have a significant size, therefore we codify it as a sparse vector, an ordered list of the indices of the non zeros.

The elements of the second and third populations are points of the intervals  $\Gamma_s$  that form the input part of the training set. Each element  $x_s$  will be codified as a pair  $(s, \delta)$ , where  $\delta(x_s) = (x_s - x_s^-)/(x_s^+ - x_s^-)$ . For instance, if we are given a sample of two imprecise values  $\{(\mathbf{x}_1 = [0, 3] \times [1, 2] \times [3, 4], \text{class} = 1), (\mathbf{x}_2 = [3, 4] \times [1, 1] \times [3, 3], \text{class} = 2)\}$ , the list  $\{1, (0.5, 1, 0.25)\}$  is a valid individual, and it represents a point  $(1.5, 2, 3.25) \in [0, 3] \times [1, 2] \times [3, 4]$ .

### 4.2 Fitness value

The fitness value of a model is an interval of values of likelihood (see Eq. 20). Hence, for computing the fitness of a model we need two crisp training sets. In this algorithm, these sets are obtained from the second and third populations.

The fitness value of an individual in these last two populations is, respectively, the gain or loss in the lower and upper bounds of the likelihood of the model, when the point contained in the individual is replaced by the midpoint of  $\Gamma_s$  (where  $s$  is the index codified in the individual, as mentioned). This way, the sum of the fitness values of all the individuals in the population equals the difference between the likelihood of the sample comprising the midpoints of the interval-valued training set and the likelihood of the sample codified by the whole population. The genetic evolution tends, therefore, to produce sets of values with respectively lower and higher likelihoods.

It is remarked that, in case that an index appears more than once, the fitness values of all the individuals but the best one must be set to zero, or else the sum of the fitness values is no longer the mentioned difference.

### 4.3 Coevolutionary scheme

The coevolutionary scheme is as follows:

1. All populations (models,  $x^{\text{low}}$  and  $x^{\text{high}}$ ) are initialized with random values.
2. Repeat steps 3 to 9,  $G_1$  times:
  3. Each model in the first population is optimized (see Figure 1) for a sample comprising the semi-sum of the values encoded in the populations  $x^{\text{low}}$  and  $x^{\text{high}}$ .
  4. This first population is ranked by means of a precedence operator between intervals [11]. The elite is copied apart. Tournament selection, crossover and mutation are performed in this population, and the offspring is inserted in place of the worst individuals in the tournament.
  5. Repeat steps 6 to 9,  $G_2$  times:
    6. The first element of the second population ( $x^{\text{low}}$ ) is temporarily replaced by the midpoint of its corresponding interval  $\Gamma_s$  in the training set. The likelihood of the elite model is reevaluated. The gain with respect to the lower bound of the likelihood of the elite, is the fitness of this first element. Changes are reverted, and this procedure is repeated for all the elements in this population.
    7. The first element of the third population is replaced by the midpoint of its corresponding interval  $\Gamma_s$  in the training set, and the process described in the preceding step is repeated, now for the higher bound of the likelihood.
    8. For the two last populations, if an element of the sample appears more than once, the fitness of all the instances of the element but the best one are assigned a value 0.
    9. Crossover and mutation are performed in these last two populations, and the offspring is inserted back in place (steady state).

### 4.4 Genetic operators

All algorithms are steady state and based in a tournament selection. The offspring of the winners of the tournament replace the last two elements of the tournament, whose length is used to control the selective pressure.

Standard two-point crossover and mutation are used in the first population, which is binary encoded. The other populations need custom operators. Two individuals  $(s_1, \delta_1)$  and  $(s_2, \delta_2)$  are crossed as follows:

- If  $s_1 = s_2$ , we do an arithmetic crossover between  $\delta_1$  and  $\delta_2$  [13].

- If  $s_1 \neq s_2$ , we insert a copy of the best individual and randomly generate the other.

## 5 Numerical results

This section contains the initial results of the algorithm, when applied to synthetic problems. There are two categories of datasets that should be suitable for a method like this:

1. Data for which the classification rules can be expressed with a compact rulebase: low to moderate number of features, not too complex decision surface.
2. Low quality data: censoring, interval valued and missing features in the data.

To comply with our first requirement, we have built a RSRBS comprising 9 rules in a problem with two inputs between 0 and 1, and two classes. This RSRBS is a model of a joint probability of the input features and the class (Section 2). Since we know the distribution of the population, we have generated datasets whose Bayes error is also known, and for which there exists a RSRBS which is the optimal solution. Two datasets of sizes 100 and 1000 were generated.

The second requirement has been fulfilled by adding imprecision to these datasets. We have considered three different categories of imprecision:

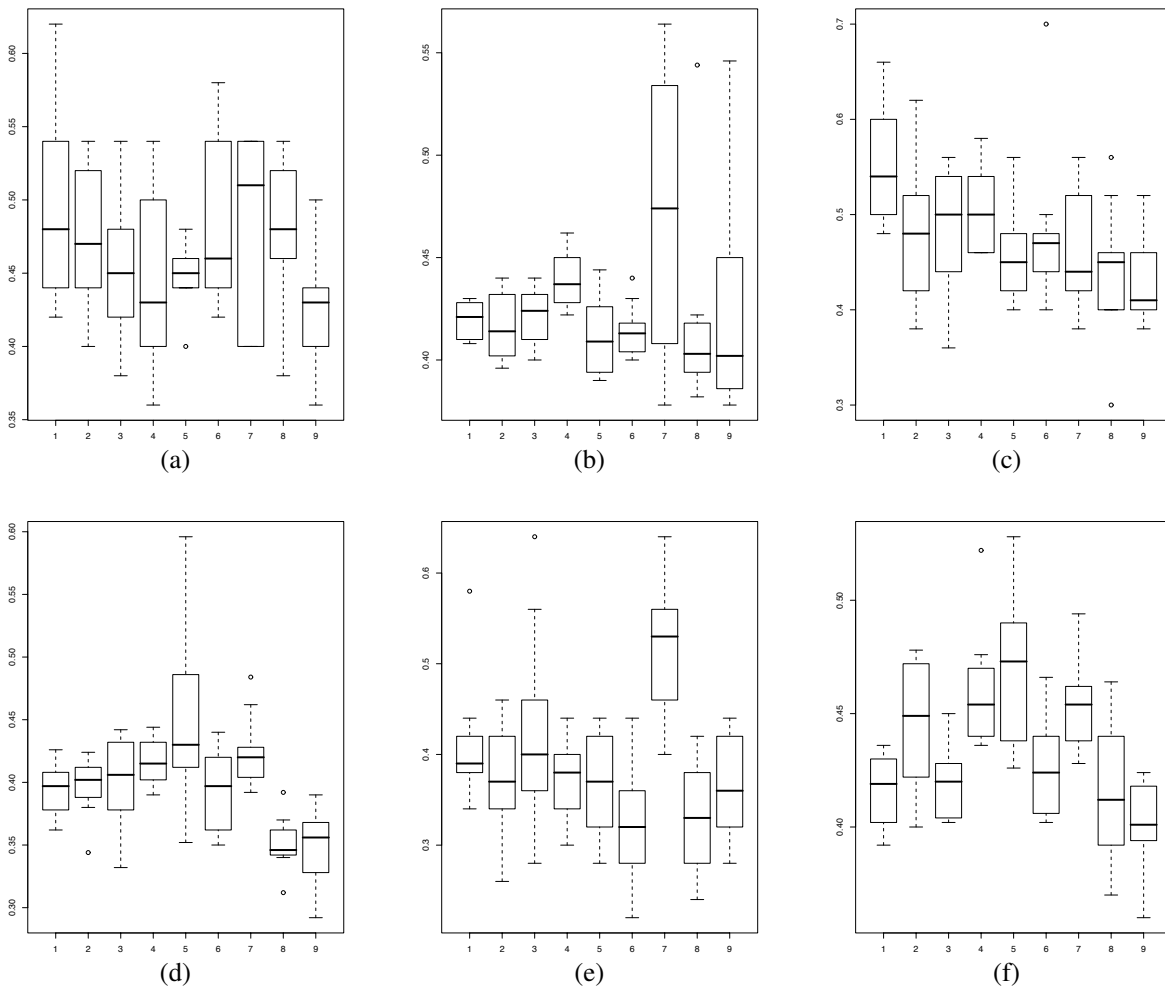
1. Censoring: in the 50% of cases, the training data  $x_s$  is replaced by the interval  $[0, x_s]$ . The other cases were replaced by the interval  $[x_s, 1]$ .
2. Interval valued data: each training data is replaced by the interval  $[x_s, x_s + 0.4]$ . or  $[x_s, 1]$  if  $x_s + 0.4 > 1$
3. Missing values: 40% of the points in the training set had one of their features replaced by the interval  $[0, 1]$ .

These three additions were performed for both datasets, giving the six problems we will use in this section. Other details of the experimental setup are: each experiment has been repeated 10 times, with a 5x2cv experimental design. The size of the first genetic population is 25. Second and third populations are of sizes 100 or 1000, depending on the dataset. The number of generations  $G_1$  is 50 and  $G_2$  is 5 (see Section 4.3). The probabilities of crossover and mutation in the first population are 0.7 and 0.1, and the probability of crossover in the second and third populations are equal to 0.9. The tournament size is 5.

For crisp algorithms (LDA and QDA discriminant analysis [3], multilayer perceptron [4], KNN classifier, Chi [1], Ishibuchi [6], Pal-Mandal [14] and RSRBS [19]), we replaced each interval by its midpoint. We expect that our approach performs the best in all the cases we selected, and

	Linear	Quadratic	Neural	KNN	WM	ISH	PM	Crisp RSRBS	Interval RSRBS
censored - 100	0.492	0.478	0.460	0.448	0.448	0.488	0.478	0.478	<b>0.424</b>
censored - 1000	0.421	0.414	0.424	0.437	0.409	0.413	0.474	0.403	<b>0.402</b>
interval - 100	0.554	0.478	0.490	0.506	0.460	0.478	0.458	0.442	<b>0.432</b>
interval - 1000	0.394	0.397	0.402	0.416	0.450	0.393	0.424	0.351	<b>0.346</b>
missing - 100	0.408	0.372	0.426	0.376	0.364	<b>0.328</b>	0.518	0.330	0.372
missing - 1000	0.416	0.445	0.412	0.461	0.470	0.426	0.456	0.415	<b>0.401</b>

**Table 1. Numerical results: Crisp algorithms (LDA and QDA discriminant analysis [3], multilayer perceptron [4], KNN classifier, Chi [1], Ishibuchi [6], Pal-Mandal [14] and RSRBS [19] were compared to Interval-RSRBS. The best test results are boldfaced.**



**Figure 3. Boxplots showing the dispersion of the results in Figure 1. Censored data, sizes 100 (a) and 1000 (b). Interval valued data, sizes 100 (c) and 1000 (d). Missing data, sizes (e) and 1000 (f). The algorithms being compared are in the same order as they appear in Figure 1.**

also that the final populations  $x^{\text{low}}$  and  $x^{\text{high}}$  contain the most conflictive points for the classifier (i.e., those points that, if removed, reduce the most the width of the interval of likelihoods of the model).

The mean value of the test results are shown in Table 1, and the boxplots depicting the relevance of the differences are displayed in Figure 3. We have obtained the expected results in all cases but one (40% of missing data, datasets of size 100), where the crisp version of the same algorithm improved the results. At the sight of these preliminary results, we think that this algorithm is a promising new technique for exploiting interval data in rule-based classification problems.

## 6 Concluding remarks

In this paper we have tried a different approach for obtaining linguistically understandable classifiers from interval-valued data. We have defined a particular case of FRBS and its optimal assignment of weights. Then we have combined a descent algorithm with a coevolutionary scheme and searched in parallel for the best set of rules, and for the two selections of the training set where the lowest and highest likelihood are reached. These two bounds are used to find a model which is not dominated by other models, and that results in a robust estimation under vague input data. We have checked, with an ad-hoc dataset, that this approach is able to obtain better models than some statistical and fuzzy classifiers, if the conditions are appropriate.

## Acknowledgements

This work was supported by the Spanish Ministry of Science and Innovation, under grants TIN2008-06681-C06-04 and MTM2007-61193.

## References

[1] Chi, Z., Yan, H., Pham, T. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. World Scientific. 1996.

[2] Goodman, Nguyen, *Uncertainty Models for Knowledge-based Systems*, North-Holland. 1985

[3] Hand, D. J. *Discrimination and Classification*. Wiley. 1981

[4] Haykin, S. *Neural Networks*. Prentice Hall, 1999.

[5] Ho, T. Data Complexity Analysis: Linkage between Context and Solution in Classification. SSPR/SPR 2008, 986-995. 2008

[6] Ishibuchi, H., Nakashima, T., Murata, T., A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems. In Proc. of 2nd IEEE CEC, 759–764. 1995

[7] Ishibuchi H, Nakashima T, Murata T, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man and Cybernetics. Part B-Cybernetics* 29(5):601-618. 1999.

[8] Ishibuchi, H., Nakashima, T. and Morisawa, T., Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems*, vol 103, no. 2, pp 223-239, 1999.

[9] De Jong KA, Spears WM, Gordon D.F. Using genetic algorithms for concept learning. *Machine Learning* 13:161-188. 1993.

[10] Koeppen, M., Franke, K., and Nickolay, B., Fuzzy-Pareto-Dominance driven multi-objective genetic algorithm. in Proc. 10th International Fuzzy Systems Association World Congress (IFSA), Istanbul, Turkey, 2003: 450–453. 2003.

[11] Limbourg, P., Multi-objective optimization of problems with epistemic uncertainty. in EMO 2005: 413–427. 2005.

[12] Luenberger, D. G. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[13] Michalewicz, Z. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992

[14] Pal, S. K., Mandal, D. P. “Linguistic recognition system based in approximate reasoning”. *Information Sciences* **61**, pp. 135-161. 1992.

[15] Potter, M., De Jong, K. A cooperative coevolutionary approach to function optimization. *Lecture Notes in Computer Science* 866. 249-257. 2006.

[16] Ruspini, E.H. A new approach to clustering. *Inf. Control* **15** pp. 22-32. 1969.

[17] Ratschek, H. Some recent aspects of interval algorithms for global optimization. In: Moore RE (ed) *Reliability in Computing: The Role of Interval Methods in Scientific Computing*. Acad. Press, New York, pp 325? 339. 1988.

[18] Sánchez, L., A random sets-based method for identifying fuzzy models. *Fuzzy Sets and Systems* 98 (3) 343-354. 1998.

[19] Sánchez, L., Casillas, J. Cordón, O., del Jesus, M. J. Some relationships between fuzzy and random classifiers and models. *International Journal of Approximate Reasoning* 29, pp. 175-213. 2002.

[20] Sánchez, L., Couso, I., Casillas, J. Modeling vague data with genetic fuzzy systems under a combination of crisp and imprecise criteria. First IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007). Honolulu, Hawaii, USA, 2007.

[21] Sánchez, L., Couso, I., Casillas, J. Genetic learning of fuzzy rules based on low quality data. *Fuzzy Sets and Systems*. 160 (17) 2524-2552. 2009