**Noname manuscript No.**
(will be inserted by the editor)

# Obtaining fuzzy rules from interval censored data with genetic algorithms and a random sets-based semantic of the linguistic labels

**Luciano Sánchez, and Inés Couso**

the date of receipt and acceptance should be inserted later

**Abstract** Fuzzy memberships can be understood as coverage functions of random sets. This interpretation makes sense in the context of fuzzy rule learning: a random sets-based semantic of the linguistic labels is compatible with the use of fuzzy statistics for obtaining knowledge bases from data. In particular, in this paper we formulate the learning of a fuzzy rule based classifier as a problem of statistical inference. We propose to learn rules by maximizing the likelihood of the classifier.

Furthermore, we have extended this methodology to interval censored data, and propose to use upper and lower bounds of the likelihood to evolve rule bases. Combining descent algorithms and a co-evolutionary scheme, we are able to obtain rule-based classifiers from imprecise datasets, and can also identify the conflictive instances in the training set: those that contribute the most to the indetermination of the likelihood of the model.

## 1 Introduction

It is well known that fuzzy memberships can be interpreted as coverage functions of random sets [5,7]. This interpretation makes sense from a possibilistic point of view and is also related to the likelihood-based vision of a fuzzy set [6], therefore it can be used for assessing the semantics of linguistic labels that have been numerically obtained from data.

Computer Science Department, University of Oviedo, Campus de Viesques, 33071 Gijón, Asturias, Spain E-mail: luciano@uniovi.es · Statistics Department, University of Oviedo, Facultad de Ciencias, 33071 Oviedo, Asturias, Spain E-mail: couso@uniovi.es

In the context of machine learning, many properties of this interpretation can be exploited for learning Knowledge Bases (KBs) from data. This has been done in combination with clustering algorithms and also Genetic Fuzzy Systems (GFSs): in [27] it was derived a procedure for fitting a one point coverage function to a cloud of data and this procedure was embedded in a single linkage hierarchical clustering for obtaining scatter fuzzy rules from data. More recently, in the GFS field, it has been introduced the concept of random sets-rule based system (from now on, RSRBS) and it was shown that, under certain conditions, RSRBSs are numerically equivalent to fuzzy rule based systems (FRBSs) and thus they can be regarded as such [28].

The initial purpose of RSRBSs was to serve as a threshold when determining the quality of other GFSs. 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 [12]) or the learning algorithm is accountable because it has not found the best knowledge base (KB). As we will show later in this paper, there are not analytical results about the optimal weight assignment for FRBSs in the general case, but this optimal assignment can be found for RSRBSs. As a consequence of this result, RSRBSs can be estimated from data with deterministic descent algorithms. Comparing the quality of a suitable RSRBS to that of the GFS under study, we can find those cases where a GFS has not properly converged to an appropriate KB (but the information in its corresponding dataset can be represented with an FRBS) or, by the contrary, its decision surface is too complex for a linguistic classifier and we cannot expect that a GFS scores well for the problem. In the same paper [28]

it has been shown that a Genetic Algorithm (GA) can perform a rule selection in an RSRBS and the resulting KB is still competitive with state-of-the-art GFSs in both accuracy and interpretability.

Furthermore, in this paper we have generalized these RSRBSs to interval censored data. That is to say, data that is either known through a pair of bounds, or an upper or lower bound of the true value. We will extend the results of [28] to this kind of data, which is seldom considered in the GFS field [10], and use RSRBSs for detecting those datasets where the inaccuracy of the data prevents us from finding a useful decision surface. We also want to combine the method with a genetic rule selection for deriving a linguistically understandable fuzzy rule based classifier that can take advantage of this particular case of imprecise data.

There are, however, many numerical difficulties when obtaining an RSRBS from interval data, because the precise minimum of the objective function cannot be produced. At the most, we will obtain a set of feasible solutions constrained by the bounds of the values of each training data [26]. In previous works on this subject it has been proposed to use evolutionary schemes guided to obtain nondominated sets of bounds of the objective function by means of multicriteria GAs [29, 30]. This could be used to solve this problem, however the computational cost is high. In this work, we propose a more efficient coevolutionary scheme [24] 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 uncertainty about the fitness of the classifier. This list of instances is crucial for improving the computer efficiency of our approach. We will show later that all the elements in the training set can be approximated by crisp data except the elements in that list. Reducing the number of imprecise elements in the training set is the crux for being competitive in cost with crisp GFSs.

This paper is structured as follows: in Section 2 we recall and update the definition of RSRBS introduced in [28], and its similarities with an FRBS. In Section 3 we state the maximum likelihood estimate of an 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 $C \in \{1, \ldots, l\}$ be the class labels, $x = (x^1, \ldots, x^n)$, the features with which we perceive an object, and let $X$ be the input space, $x \in X = X^1 \times \ldots \times X^n$. Lastly, let the Bayes minimum error classifier be

$$\text{class}(x_0) = \arg\max_c P(C = c \mid X = x_0). \tag{1}$$

We will consider that a rule-based classifier is a parametric model of $P(c \mid x)$ 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 an instrumental model that will be used later in the definition of an RSRBS. We will call "crisp parametric model" to a pair comprising a partition $\{A_1, \ldots, A_m\}$ of the input space $X$ and a matrix

$$\mathbf{M} = \begin{pmatrix} p_{11} & \cdots & p_{1l} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{ml} \end{pmatrix} \tag{2}$$

where

$$p_{ic} = P(c \mid A_i). \tag{3}$$

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

$$\begin{aligned} P(c_0 \mid x) = & \\ &= \sum_{i=1}^{m} P(c_0 \mid A_i) P(A_i \mid x) \\ &= \sum_{i=1}^{m} p_{ic_0} I_{A_i}(x) \end{aligned} \tag{4}$$

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 (see Figure 1),

$$A_i = A_i^1 \times \ldots \times A_i^n, \quad A_i^j \subset X^j \tag{5}$$

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

**if** $x_1$ **is** $A_i^1$ **and** $\ldots$ **and** $x_n$ **is** $A_i^n$ **then** class is $c_i$ **with** $p_i$.

In case there exists a partition $\{L_k^1, \ldots, L_k^{m_j}\}$ on each variable $j$,

$$X^j = \bigcup_{k=1}^{m_j} L_k^j, \quad L_k^j \cap L_m^j = \emptyset \text{ for } m \neq j \tag{6}$$

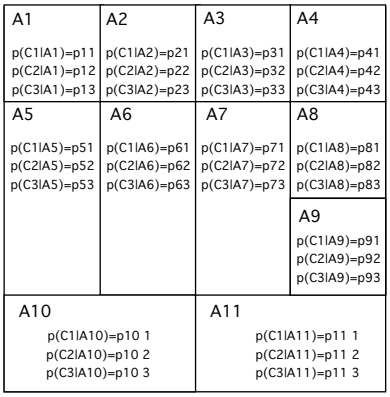


**Fig. 1** A crisp parametric model is an instrumental model that comprises a crisp partition of the input space and a matrix of probabilities. A crisp model for a problem with two input variables and three classes is shown. The input partition is decomposable and has 11 elements. Each cell can be linguistically expressed as three interval rules "If $(x_1, x_2) \in A_i$ the class is $c_1$ with $p_{i1}$", "If $(x_1, x_2) \in A_i$ the class is $c_2$ with $p_{i2}$" and 'If $(x_1, x_2) \in A_i$ the class is $c_3$ with $p_{i3}$."

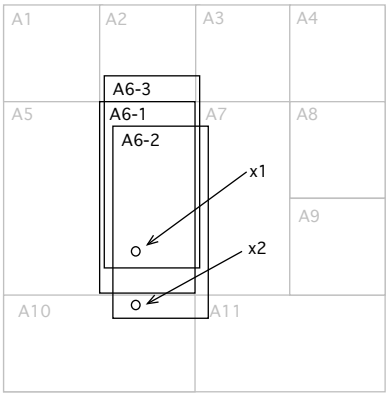


**Fig. 2** A random sets-based model comprises a family of crisp models and a probability defined in this family. In the figure we depict the case where the family comprises three elements. The 6-th cell of the corresponding partitions is shown in detail. Observe that the point $x_1$ always belongs to $A_6$, but $x_2$ belongs to $A_6$ with certain probability. This probability (that is, the one-point coverage function of the random set with images $\{A_6^1, A_6^2, A_6^3\}$) will be understood as a fuzzy membership function to a fuzzy cell $\widetilde{A}_6$.

such that all terms in the antecedent of the rules fullfill that $A_i^j = L_k^j$ for some k, then the linguistic rule is *descriptive*. Otherwise, it is a *scatter* rule [4].

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 {SMALL, LARGE} with SMALL= $[0, 0.5]$, LARGE= $[0.5, 1]$ a linguistic partition of $X^1$, and let {SHORT, LONG} with SHORT= $[1, 1.5]$, LONG= $[1.5, 2]$ a linguistic partition of $X^2$. Lastly, consider the rule that follows:

**if** $x_1$ **is** SMALL **and** $x_2$ **is** SHORT **then** class 2 **with** 0.8.

The information provided by this rule is

$$P(C = 2 \mid x \in [0, 0.5] \times [1, 1.5]) = 0.8.$$

In the next section we will define a random-sets based model by means of a family of crisp models and a probability distribution defined over this family.

## 2.2 Random sets-based model

As we have just mentioned, 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 $\{A_1^\theta, \ldots, A_m^\theta\}$ of the input space $X$ (see Figure 2).

To classify an input value $x$, we average the outputs of all the crisp models in the family:

$$\begin{aligned} P(c_0 \mid x) &= \int_\Theta \left( \sum_{i=1}^m p_{ic_0} I_{A_i^\theta}(x) \right) dP_\theta \\ &= \sum_{i=1}^m p_{ic_0} \int_\Theta I_{A_i^\theta}(x) dP_\theta \\ &= \sum_{i=1}^m p_{ic_0} \Phi_i(x) \end{aligned} \tag{7}$$

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

$$\Phi_i(x) = \prod_{j=1}^n \Phi_i^j(x_j) \tag{8}$$

where

$$\Phi_i^j(x_j) = \int_\Theta I_{A_i^\theta}^j(x) dP_\theta \tag{9}$$

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

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

According to [5, 7], the one point coverage function of a random set can be understood as a fuzzy membership

funcion. If the functions $\Phi_i$ are regarded as membership functions, then

$$\begin{aligned}\sum_{i=1}^{m}\Phi_i(x) &= \sum_{i=1}^{m}\int_{\Theta} I_{A_i^\theta}(x)dP_\theta \\ &= \int_{\Theta}\left(\sum_{i=1}^{m} I_{A_i^\theta}(x)\right)dP_\theta \\ &= \int_{\Theta} dP_\theta \\ &= 1\end{aligned} \tag{10}$$

thus they form a Ruspini's fuzzy partition of $X$ [25].

This means that the linguistic information of a random sets-based model is compatible with that of a fuzzy model, at least for certain t-norms, t-conorms and inference procedures. Observe that the inference in an FRBS comprising rules "if $\widetilde{A}_k$ then $c_k$ with $w_k$" is:

$$\text{class}(x) = \arg\max_c \left\{ \bigvee_{i:c_i=c} \left( \bigwedge_j \widetilde{A}_i^j(x) \wedge w_i \right) \right\} \tag{11}$$

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

$$\text{class}(x) = \arg\max_c \left\{ \sum_{i:c_i=c} \left( \prod_j \Phi_i^j(x) \cdot w_i \right) \right\} \tag{12}$$

that is to say, the RSRBS is a particular case of fuzzy classifier where $\sum \widetilde{A}(x) = 1$ for any $x$, and voting-based inference [17] and product t-norm are used.

It is remarked that these fuzzy classifiers may be expressed with type-III fuzzy rules [3]; each group of $l$ random set-based rules like

$$\begin{gathered}\text{if } \Phi_k \text{ then } c_1 \text{ with } w_{k1} \\ \vdots \\ \text{if } \Phi_k \text{ then } c_l \text{ with } w_{kl}\end{gathered}$$

carries the same meaning that

"if $\widetilde{A}_k$ then $c_1$ with $w_{k1}$ and ... and $c_l$ with $w_{kl}$".

## 3 Estimation of an RSRBS from data

If the linguistic partitions are not modified during the learning, then obtaining an RSRBS from data consists of infering the weights of the rules, i.e. the matrix $\mathbf{M}$. This is similar to many of the algorithms used for obtaining weighed fuzzy classification rules from data [13]: we want to determine a sparse set of weights for the elements of a large set, comprising all the candidate rules. This large set can contain either an exhaustive enumeration of all the valid antecedents or comprise the rules produced by another learning algorithm that was applied to the training data [15][16].

It is remarked that many authors prefer using rules with binary weights and also optimize the parameters defining the membership functions [4]. While there is nothing in our previous explanation that prevents either way, we have decided to explore the case where the membership functions are not learnt neither tuned, but the rule weighs are. Our initial pool of rules comprises the set of all possible antecedents. Observe also that we are not assuming that the consequents of these rules are the alternatives with highest confidence [15]: cases can be found where this assignment is not optimal [28].

The most relevant consequence of our decision is the set of necessary conditions that follows. The weights of the rules in an RSRBS must fulfill these conditions after the training process, and the same conditions will be the base of a numerical algorithm that we will propose later.

**Lemma 1** *Let an RSRBS comprise $m$ linguistic rules*

*"if $\widetilde{A}_k$ then $c_1$ with $w_{k1}$ and ... and $c_l$ with $w_{kl}$"*

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

$$\sum_{s:\mathbf{c}_s=a} \frac{\widetilde{A}_i(\mathbf{x}_s)}{\sum_{k=1}^{m}\widetilde{A}_k(\mathbf{x}_s)w_{k\mathbf{c}_s}} = \sum_{s:\mathbf{c}_s=b} \frac{\widetilde{A}_i(\mathbf{x}_s)}{\sum_{k=1}^{m}\widetilde{A}_k(\mathbf{x}_s)w_{k\mathbf{c}_s}} \tag{13}$$

*for all $a, b \in \{1, \ldots, l\}$, and $i = 1, \ldots, m$*

*Proof* *Let us consider that an RSRBS produces the probabilities $p(c \mid x)$ of each class, conditioned to the input, as we have stated in eq. 7. This way, the the log-likelihood of the RSRBS is*

$$L(\mathbf{M}) = \sum_{s=1}^{q} \log \sum_{i=1}^{m} \Phi_i(\mathbf{x}_s) p_{i\mathbf{c}_s} \tag{14}$$

*and there are $m$ constraints*

$$\sum_{c=1}^{l} p_{ic} = 1. \tag{15}$$

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

$$\begin{aligned}L'(\mathbf{M}) = \\ \sum_{s=1}^{q} \log \sum_{i=1}^{m} \Phi_i(\mathbf{x}_s) p_{i\mathbf{c}_s} + \sum_{i=1}^{m} \lambda_i (1 - \sum_{c=1}^{l} p_{ic})\end{aligned} \tag{16}$$

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

$$\sum_{s:\mathbf{c}_s=c} \frac{\Phi_i(\mathbf{x}_s)}{\sum_{k=1}^{m}\Phi_k(\mathbf{x}_s)p_{kc_s}} = \lambda_i \tag{17}$$

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

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

**Fig. 3** Pseudocode of the numerical algorithm used to solve the set of equations (13).

*for* $i = 1, \ldots, m$, $c = 1, \ldots, l$ *and*

$$\sum_{c=1}^{l} p_{ic} = 1 \tag{18}$$

*thus eq. 13 fulfills.*

Observe that these conditions are necessary but not sufficient, because the likelihood function is not always unimodal. However, in practice good solutions are found starting from an uniform assignment of weights (all weights equal to $1/l$) and using the deterministic algorithm in Figure 3. This algorithm combines solving these $m(l+1)$ nonlinear equations with a descent step based on a Brent linear search [21] and 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 $1/l$ 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.

The linear search (determination of the value of $\alpha$) was implemented with Brent's method. All points examined fulfill eq. (18) because of the function `normalize`, and the algorithm stops when the conditions (17) are approximately true.

### 3.1 Generalization to interval-valued data

Let us study the case where the input data cannot be precisely observed, but we perceive intervals that contain them. This includes, for instance, inexact measurements, censored data and missing values (represented by an interval that spans the range of the unknown variable). In particular, consider that we have a sample

$$\{(\Gamma_s, \mathbf{c}_s)\}_{s=1,\ldots,q}$$

where

$$\Gamma_s = [x_{1s}^-, x_{1s}^+] \times \ldots \times [x_{ns}^-, x_{ns}^+]$$

is an interval of $\mathbf{R}^n$. Let $\mathbf{g}_s = (x_{1s}, x_{2s}, \ldots, x_{ns})$ be a vector or $\mathbf{R}^n$, such that $x_{is} \in [x_{is}^-, x_{is}^+]$, thus the sequence $(\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_q)$ is a selection of the sample, $\mathbf{g}_s \in \Gamma_s$.

If the true training set was this selection, its likelihood would be

$$L(\mathbf{M}) = \sum_{s=1}^{q} \log \sum_{i=1}^{m} \Phi_i(x_i) p_{i\mathbf{c}_s}. \tag{19}$$

It is clear that the likelihood of the RSRBS, given the information provided by the interval sample, is an unknown value in the set

$$[L^-(\mathbf{M}), L^+(\mathbf{M})] = \left\{ \sum_{s=1}^{q} \log \sum_{i=1}^{m} \Phi_i(\mathbf{g}_s) p_{i\mathbf{c}_s} \mid \mathbf{g}_s \in \Gamma_s \right\}, \tag{20}$$

with the same $m$ constraints as before.

It is also clear that, generally speaking, we can no longer determine an unique set of weights $\mathbf{M}$ but we want to find the largest set of nondominated matrices

$$\{\mathbf{M} \mid L^+(\mathbf{M}') > L^-(\mathbf{M}) \text{ for all } \mathbf{M}'\}. \tag{21}$$

#### *3.1.1 GAs and interval-valued optimization*

Genetic algorithms are well suited for this kind of search, that is closely related to multicriteria optimization [19].

Let us clarify the task at hand with the help of an example: in Figure 4 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, $x_0$ is in the area marked "Pareto front in Genotype Space".

Former genetic solutions to this problem [29] defined the fitness of each individual $x$ as the set $[f^-(x), f^+(x)]$ and then introduced a precedence between fitness values, for instance

$$[a,b] \prec [c,d] \iff b < c. \tag{22}$$

Many different precedence operators can be used [11]. Once the operator is selected, a GA depending on it can be easily defined. If the precedence operator induces a total order in the set of fitness values, a scalar algorithm is suitable, or else a multicriteria GA is needed. These extended multicriteria GA produce sets of individuals contained in the Pareto front in the genotype space.

In this paper we will improve this schema, as we will detail in the next section. We want to identify those "max-min" and "min-min" individuals in Figure 4. 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.

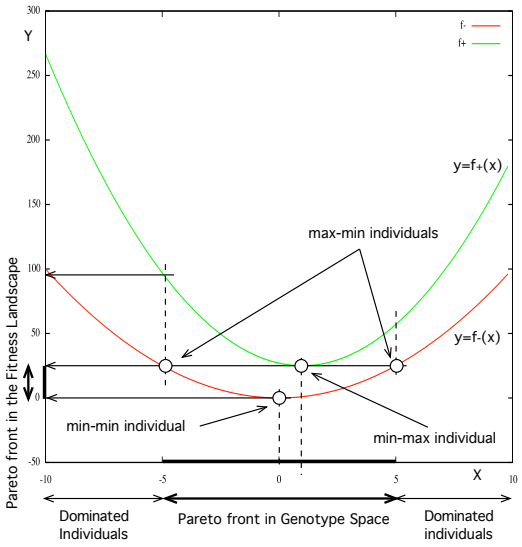


**Fig. 4** Interval optimization: $f^-(x) \leq f(x) \leq f^+(x)$. $f^-$ and $f^+$ are known, but $f$ is not. Hence, the minimum of $f$ cannot be known, but we can bound the values of $x$ and $f(x)$ at the minimum.

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

In this section we propose a novel coevolutionary algorithm that solves the optimization problem mentioned in the last section. We want to obtain the bounds of the likelihood for an interval censored dataset, and at the same time perform a rule selection that produces a compact rule base.

In addition, we want to save as much computer time as possible, so this method is comparable to crisp GFSs in execution time. It is remarked that for the most part, the extra overhead of an interval-data based GFS is consumed evaluating the set of classes to which an imprecise input belongs, i.e. determining the set

$$\text{class}(\Gamma) = \left\{ \arg\max_c \left\{ \sum_{i:c_i=c} \left( \prod_j \Phi_i^j(x) \cdot w_i \right) \right\} \mid x \in \Gamma \right\} \tag{23}$$

Solving eq. 23 requires, in turn, of a new optimization algorithm, that can be rather costly.

However, there is no need for computing eq. 23 for all the elements of the training set. Observe, for instance, the situation depicted in Figure 5. If we wanted to obtain the misclassification rate of the classifier given by the decision surface in the figure, it is clear that we can replace all those instances that do not intersect the decision surface with points. Instead, if we want to obtain the likelihood this is not exact, nevertheless it is still true that we can replace most of the points in the dataset by crisp instances without committing large errors in the approximation.

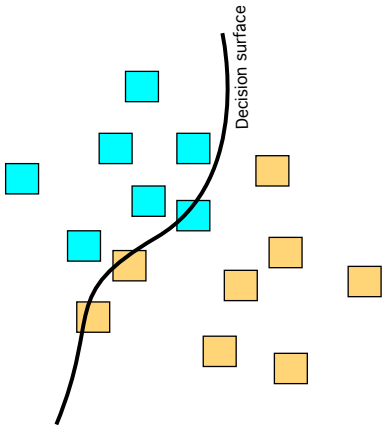


**Fig. 5** Classification with interval data: all the elements that are not crossed by the decision surface can be replaced by any point in their interior without altering the error rate, and with little influence in the specificity of the likelihood of the classifier.

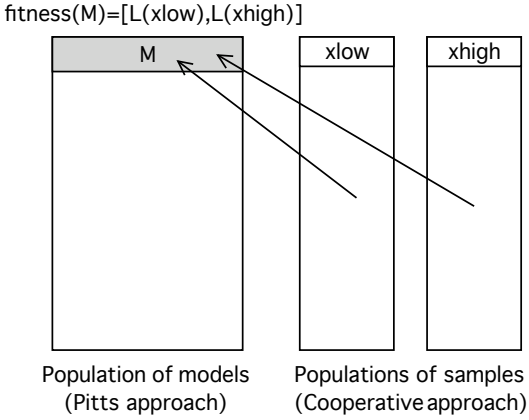


**Fig. 6** Populations of the coevolutionary genetic algorithm. The first one contains different model candidates, the second and the third contain selections of the sample where the upper and lower bounds of the likelihood are reached.

Likewise, we propose an algorithm that searches the best selection of rules, and at the same time finds those elements of the sample that can be replaced by crisp selections with the lowest approximation error. The algorithm depends on three populations (see Figure 6). The first one contains different model candidates, represented by their matrices $\mathbf{M}$ (thus each individual represents a model, Pitts style [18]), and the other two contain 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 [14]). These three populations coevolve to find the best RSRBS and the extrema $\{x_s^{\text{low}}\}$, $\{x_s^{\text{high}}\}$. The parts and operators defining this GA are described in detail in the remaining of the 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 3. This vector stores the set of rows of the matrix $\mathbf{M}$ whose terms are different from $1/l$; 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 encode 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 contained in the intervals $\Gamma_s$ that form the input part of the training set. Recall that we eventually want to find the upper and lower extrema of the likelihood of the classifier. The actual values of the elements of the sample are not known, but we know bounds for them: the unknown value of the $s$-th input $x_s$ is in the interval $\Gamma_s$. In addition, we assume that, for most of the elements in the sample, making a wrong guess about $x_s$ does not influence too much the likelihood of the classifier: those elements can be replaced with the midpoints of $\Gamma_s$.

Only those values of $x_s$ that make a difference in the likelihood will be stored. It is needed to determine which elements are those, and for each one of them we need to know the point in $\Gamma_s$ where the extremum is reached, therefore each element of the population will be a pair (index,value). The value of $x_s$ will be normalized so that the lower and upper bounds of its coordinates $x_{js}^-$ and $x_{js}^+$ are mapped to the values zero and one, respectively, at the corresponding alleles. That is to say, each element $x_s$ will be codified as a pair $[s, (\delta_{1s}, \ldots, \delta_{ns})]$, where $\delta_{js} = (x_{js} - x_{js}^-)/(x_{js}^+ - x_{js}^-)$. 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 $(1.5, 2, 3.25) \in [0,3] \times [1,2] \times [3,4]$.

It is remarked that the index $s$ is included in the representation because we will manage population sizes lower than the number of instances in the training set, and also because the same index can appear more than once in the same population, associated with different candidates for these selections which maximize and minimize the likelihood for the s-th element of the sample.

### 4.2 Fitness value

The fitness value of a model is an interval of values of likelihood (see Eq. 14). The extrema of this interval are reached for certain selections of the interval-valued sample. These selections are stored in the second and third populations.

Due to this, 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 (see Figure 7 for an actual plot of the bounds of the likelihood of the best

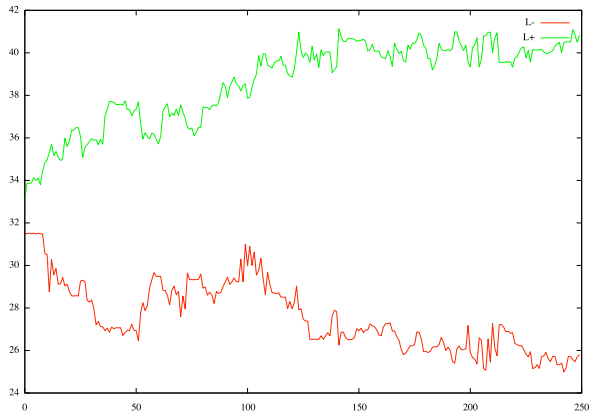


**Fig. 7** Example run of the GA: Bounds of the likelihood of the best model in the first population when the second and third populations evolve.

model in the first population when the second and third populations evolve).

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 and the algorithm would not converge to the best solution, but to populations containing many copies of the element that makes the fork of values of the likelihood to grow the most. It may be argued that these duplicate elements need not to be stored, thus making room for new individuals and achieving a higher diversity. Nevertheless, if this decision was made, on the one side, we would also need to evolve an additional mechanism for deciding how many conflictive elements will be considered; this last mechanism is implicit in the codification we are proposing in this section. On the other side, the removal of duplicates would prevent the evolution of the value of $x_s$, as we will discuss in Section 4.5.

### 4.3 Coevolutionary scheme

The coevolutionary scheme is described with the pseudocode that 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 3) 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 [20]; this definition is reproduced in Section 4.4 for making this paper more self-contained. 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 Uniform dominance

The precedence operator between interval-valued fitness values has been adapted, as mentioned, from [20]. In short, for deciding whether an interval $[a_1, b_1]$ is preferred to another interval $[a_2, b_2]$ we define two uniform probability distributions in both intervals and assume that the two unknown fitness values $f_1$ and $f_2$ fulfill

$$f_1 \to \mathcal{U}[a_1, b_1], \quad f_2 \to \mathcal{U}[a_2, b_2]. \tag{24}$$

so that

$$[a_1, b_1] \preceq [a_2, b_2] \iff p(f_1 \geq f_2) \geq p(f_2 > f_1). \tag{25}$$

### 4.5 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 $\delta_2$ [22].
– If $s_1 \neq s_2$, we insert a copy of the best individual and randomly generate the other.

This last operator might seem too disruptive, however consider that individuals with the same index "$s$" are actually being part of a subpopulation, since their "delta" parts can be regarded as approximations to the best value of $x_s$. Individuals from different subpopulations have completely unrelated delta parts, thus we have decided to regard the crossover of individuals of different subpopulations as a decimation operator and promote the introduction of new genetic material.

## 5 Numerical results

In this section we have performed three different analysis of the algorithm:

1. Study of the robustness of the algorithm for increasing vagueness of the input data.
2. Exploitation of the information in linguistic datasets with censoring, interval valued data and missing values.
3. Study of crisp classification problems with missing values, for gaining insight into the advantages or disadvantages of an interval-based representation.

### 5.1 Robustness of the algorithm

The first set of tests is intended to show that this algorithm is consistent and the quality of the rules obtained with it degrades less with highly imprecise datasets than crisp classifiers. We have used a subset of size 100 of the Haykin's two gaussians problem [9], and have added interval-valued imprecision to the data in two different manners:

1. Each sample has been enclosed by a square of random size, not centered in the point (one of the vertices of the square is the actual value of the instance). The training sets comprise sets of squares whose sides are of varying lengths, with uniform distributions between 0 and 0.8, 1.0 and 1.2, respectively.
2. Some of the samples were enclosed in intervals spanning the values between the lowest point in the scale and the actual point, other samples were enclosed in intervals that origin in the actual point and reach the highest value in the scale (censored data).

In both cases, we have begun with a small amount of imprecision and we have gradually increased it, plotting the corresponding decision surfaces of the RSRBSs and 1NN (nearest neighbor) classifiers in Figure 8. In the upper part of the figure we show how the decision surface of RSRBS is approximately the same for different uncertainties of the sample, while the 1NN surface changes because of the changes in the centers of the squares. However, the most important differences are shown in the lower part of the figure, where a different fraction of the inputs instances are censored. Observe that the decision surface of the RSRBS is almost immune to the presence of censored data, while the 1NN is largely affected.

### 5.2 Exploitation of the information in synthetic problems

We expect that the algorithm described in this paper is able to efficiently exploit the available information in imprecisely perceived datasets. It is well known that when there are missing, censored or interval data, standard GFSs must preprocess the information and make up suitable replacements for the incomplete instances. This generated information might or might not match the actual, unknown instances. In this last case, we want to check that the degradation of the quality of the new algorithm is lower than that of the combination of a crisp algorithm and a suboptimal preprocessing of the data.

Given that the GFS proposed in this paper is not optimized for large datasets, because the representation of an individual in the first population is potentially very consuming in space, we have designed a benchmark for which

1. We know that the classification rules can be expressed with a compact rulebase: low to moderate number of features, not too complex decision surface.
2. The data has low quality, including censoring, interval valued and missing features.

To comply with our first requirement, we have built an 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 an 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:

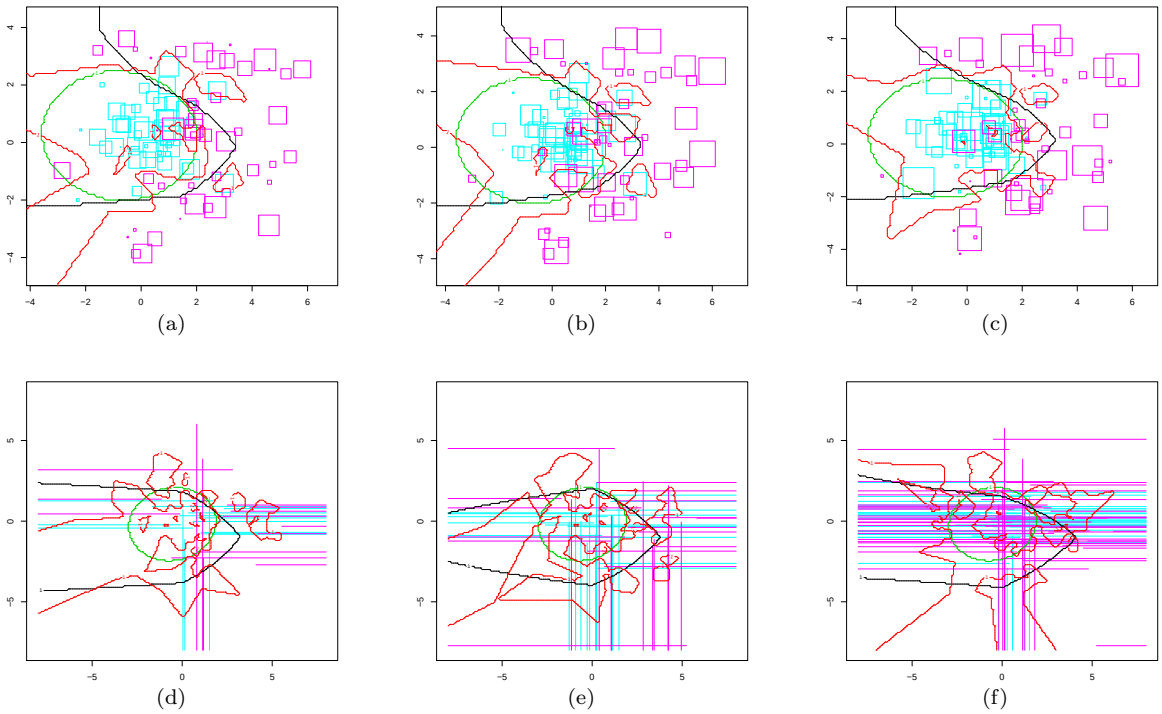


**Fig. 8** Upper part: (a), (b) and (c): The addition of interval-valued imprecision to the data alters the decision surface of classical classifiers, but the RSRBS is less affected. The average length of the side of the squares is lowest in (a) and highest in (c). Lower part: (d), (e) and (f): Different percentages of upper or lower censored data have a large influence in the decision surface of crisp classifiers, not so for RSRBSs. In all the figures, the green ellipse is the theoretical decision surface for the crisp, precise data. The classical classifier (red line) is the 1NN. The black line is the decision surface of the RSRBS, with three labels for each variable, and trained with a Genetic Algorithm as explained in the text.

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 [8], multilayer perceptron [9], KNN classifier, Chi [2], Ishibuchi [13], Pal-Mandal [23] and RSRBS [28]) we replaced each interval by its midpoint. We expect that our approach performs the best in all the cases we selected, and 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 9. 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.

### 5.3 Crisp benchmarks with missing data

While this method is not expected to improve previous algorithms for crisp data and, in particular, will produce the same results as in [28] for crisp datasets, we have studied the three datasets in the KEEL Dataset

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

**Table 1** Numerical results: Crisp algorithms (LDA and QDA discriminant analysis [8], multilayer perceptron [9], KNN classifier, Chi [2], Ishibuchi [13], Pal-Mandal [23] and RSRBS [28] were compared to Interval-RSRBS. The best test results are boldfaced.

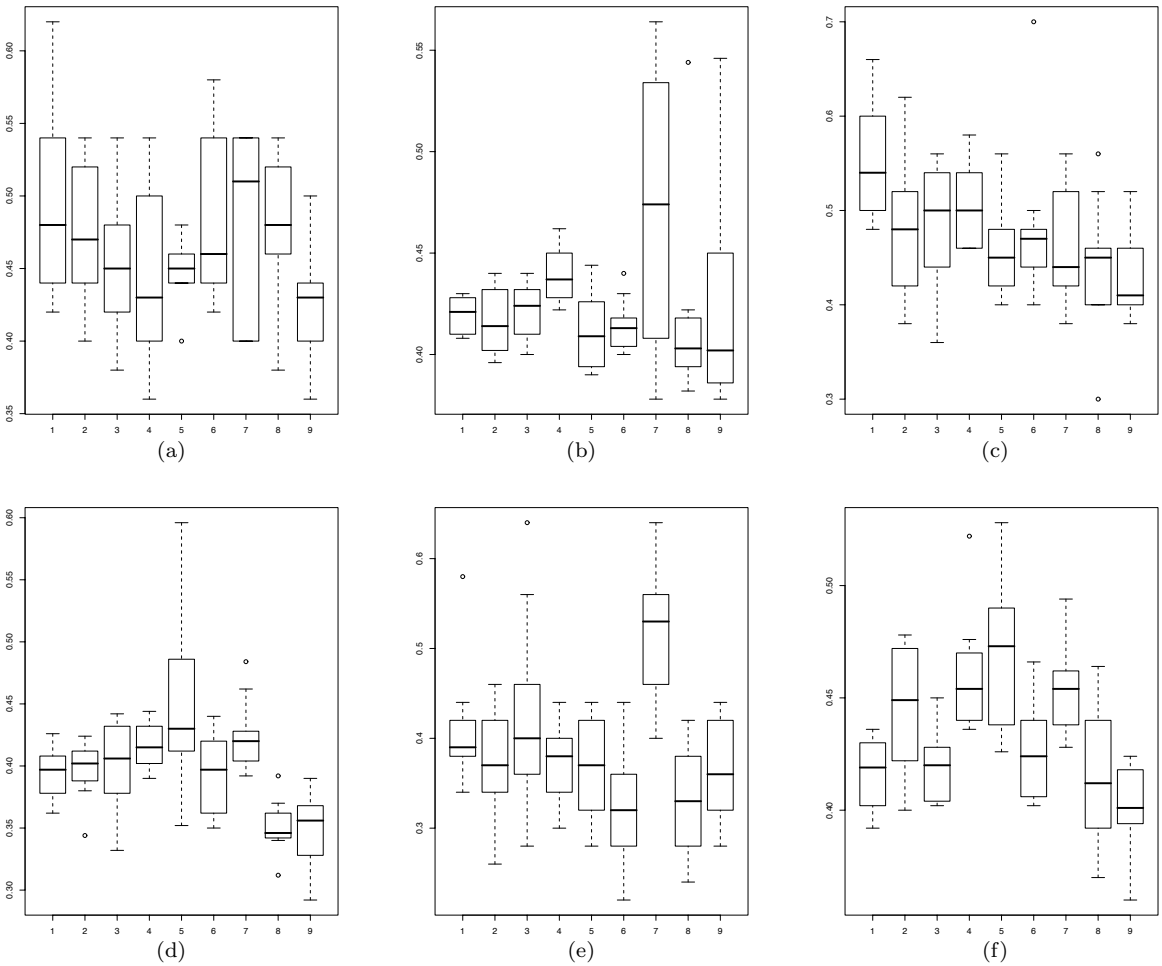


**Fig. 9** Boxplots showing the dispersion of the results in Table 1. Censored data, sizes 100 (a) and 1000 (b). Interval valued data, sizes 100 (c) and 1000 (d). Missing data, sizes (e) and 1000 (f). The algorithms being compared are in the same order as they appear in Table 1.

webpage [1] that have missing values (see the mentioned reference for a description of these datasets). We have carried a compared study of these, using the same battery of algorithms in the preceding section, for determining whether the use of the new coevolutionary genetic algorithm, combined with an interval-valued representation of the missing data, is competitive with a crisp algorithm where the missing value is replaced by the mean of the remaining elements of the variable. We expect that the improvements are minimal, if any, but also that the new algorithm is not worse than its crisp version.

The results of the experimentation are shown in Table 2 and Figure 10. From the mean values in Table 2 we can conclude that there is a small, not statistically significant advantage to this method in all the cases (see

| | Linear | Quadratic | Neural | KNN | CHI | ISH | PM | Crisp RSRBS | Interval RSRBS |
|---|---|---|---|---|---|---|---|---|---|
| Cleveland | **0.410** | - | 0.437 | 0.433 | 0.466 | **0.410** | 0.427 | 0.413 | **0.410** |
| Credit | 0.145 | 0.143 | 0.151 | 0.239 | 0.275 | 0.134 | 0.147 | 0.135 | **0.133** |
| Dermatology | 0.158 | - | 0.103 | 0.106 | 0.191 | 0.098 | 0.336 | **0.095** | **0.095** |

**Table 2** Numerical results: Crisp algorithms (LDA and QDA discriminant analysis [8], multilayer perceptron [9], KNN classifier, Chi [2], Ishibuchi [13], Pal-Mandal [23] and RSRBS [28] were compared to Interval-RSRBS. The best test results are boldfaced.

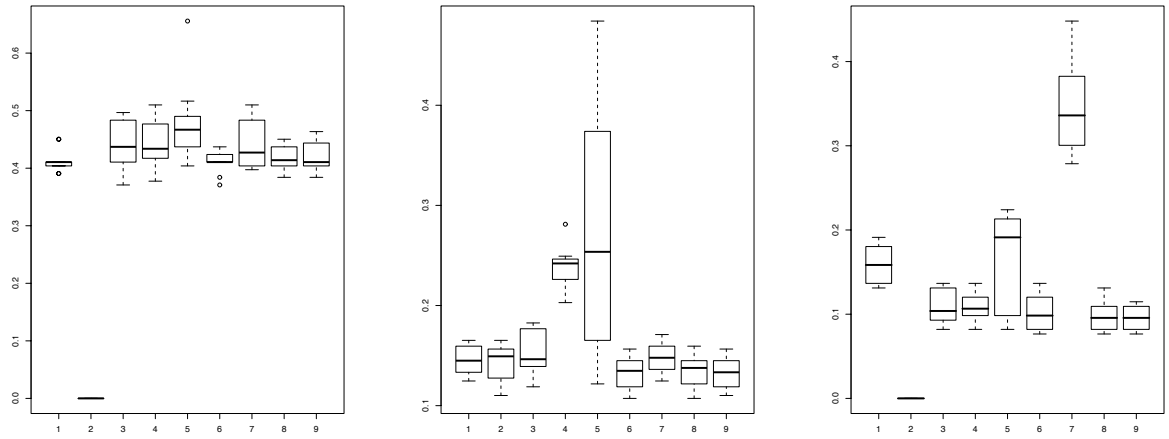


**Fig. 10** Boxplots showing the dispersion of the results in Table 2. Censored data, sizes 100 (a) and 1000 (b). Interval valued data, sizes 100 (c) and 1000 (d). Missing data, sizes (e) and 1000 (f). The algorithms being compared are in the same order as they appear in Table 2.

the boxplots in Figure 10 for information about the dispersion of the test results). This is an expected result, because the number of missing values is small and their influence in the fitness value is not very noticeable.

## 6 Concluding remarks

In this paper we have proposed a new 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. Lastly, we have checked that this approach is able to obtain better models than many statistical and fuzzy classifiers.

This is, however, a seminal work that might be benefited in the future from some changes in the representation of the rule base. In this sense, we plan to include "don't care" terms among other, more flexible descriptions of linguistic rulebases that allow reducing the number of degrees of freedom of this model in problems with a large number of input features.

## Acknowledgements

This work was supported by the Spanish Ministry of Science and Innovation, under grants TIN2008-06681-C06-04 and TIN2007-67418-C03-03.

## References

1. Alcala J. et. al. KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. Soft Computing 13:3 307-318. 2009
2. Chi, Z., Yan, H., Pham, T. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition.* World Scientific. 1996.
3. Cordón O, Jesus M.J, Herrera F., A proposal on reasoning methods in fuzzy rule-based classification systems. International Journal of Approximate Reasoning, vol. 20, no. 1, pp. 21-45, January 1999.
4. Cordón, O. Herrera, F., Hoffmann, F., Magdalena, L. *Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases.* World Scientific Publishing Company, Singapore. 2001
5. Dubois, D., Prade, H., The three semantics of fuzzy sets. Fuzzy Sets and Systems 90, pp 141-150. 1997

6. Dubois, D., Moral, S., Prade, H. A semantics for possibility theory based on likelihoods. Journal of Mathematical Analysis and Applications 205, 359-380. 1997
7. Goodman, I. R. Nguyen, N. T. *Uncertainty Models for Knowledge-based Systems*, North-Holland. 1985
8. Hand, D. J. *Discrimination and Classification.* Wiley. 1981
9. Haykin, S. Neural Networks, A Comprehensive Foundation. Prentice Hall, 1999
10. Herrera, F. Genetic Fuzzy Systems: Taxonomy, Current Research Trends and Prospects. Evolutionary Intelligence 1: 27–46. 2008.
11. Huynh, V. N., Nakamori, Y., Lawry, J. Ranking fuzzy numbers using targets. Proc. IPMU 2006, 140-149. 2006.
12. Ho, T. Data Complexity Analysis: Linkage between Context and Solution in Classification. SSPR/SPR 2008, 986-995. 2008
13. 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
14. Ishibuchi H, Nakashima T, Murata T, Performance evaluation of fuzy classifier systems for multidimensional pattern classification problems. IEEE Transactions on Systems, Man and Cybernetics. Part B-Cybernetics 29(5):601-618. 1999.
15. Ishibuchi, H., Takashima, T., Effect of rule weights in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems, 3(3),260-270, 2001.
16. Ishibuchi, H., Yamamoto, T., Rule weight specification in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems, 13(4),260-270, 2005.
17. 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.
18. De Jong K. A., Spears W. M., Gordon D. F. Using genetic algorithms for concept learning. Machine Learning 13:161-188. 1993.
19. Koeppen, M., Franke, K., and Nickolay, B., Fuzzy-Pareto-Dominance driven multi-objective genetic algorithm. in Proc. 10th International Fuzzy Systems Assotiation World Congress (IFSA), Istanbul, Turkey, 2003: 450–453. 2003.
20. Limbourg, P., Multi-objective optimization of problems with epistemic uncertainty. in EMO 2005: 413–427. 2005.
21. Luenberger, D. G. *Linear and Nonlinear Programming.* Addison-Wesley, 1984.
22. Michalewicz, Z. *Genetic algorithms + Data Structures = Evolution Programs.* Springer-Verlag, 1992
23. Pal, S. K., Mandal, D. P. Linguistic recognition system based in approximate reasoning. *Information Sciences* **61**, pp. 135-161. 1992.
24. Potter, M., De Jong, K. A cooperative coevolutionary approach to function optimization. Lecture Notes in Computer Science 866. 249-257. 2006.
25. Ruspini, E.H. A new approach to clustering. *Inf. Control* **15** pp. 22-32. 1969.
26. 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.
27. Sánchez, L., A random sets-based method for identifying fuzzy models. Fuzzy Sets and Systems 98 (3) 343-354. 1998.
28. 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.
29. 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.
30. 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