ELSEVIER International Journal of Approximate Reasoning 29 (2002) 175–213

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Some relationships between fuzzy and random set-based classifiers and models

Luciano Sánchez [a,*], Jorge Casillas [b], Oscar Cordón [b], María José del Jesus [c]

[a] *Department of Computer Science, Office 1.1.28, Campus de Viesques, University of Oviedo, 33203 Gijón, Spain*

[b] *Department of Computer Science and Artificial Intelligence, E.T.S. de Ingeniería Informática, University of Granada, 18071 Granada, Spain*

[c] *Department of Computer Science, Escuela Politécnica Superior, University of Jaén, Jaén, Spain*

Received 1 December 2000; accepted 1 July 2001

Abstract

When designing rule-based models and classifiers, some precision is sacrificed to obtain linguistic interpretability. Understandable models are not expected to outperform black boxes, but usually fuzzy learning algorithms are statistically validated by contrasting them with black-box models. Unless performance of both approaches is equivalent, it is difficult to judge whether the fuzzy one is doing its best, because the precision gap between the best understandable model and the best black-box model is not known.

In this paper we discuss how to generate probabilistic rule-based models and classifiers with the same structure as fuzzy rule-based ones. Fuzzy models, in which features are partitioned into linguistic terms, will be compared to probabilistic rule-based models with the same number of terms in every linguistic partition. We propose to use these probabilistic models to estimate a lower precision limit which fuzzy rule learning algorithms should surpass. © 2002 Published by Elsevier Science Inc.

Keywords: Fuzzy classifiers; Fuzzy models; Random set-based classifiers; Random set-based models

* Corresponding author. Tel.: +34-85-182130; fax: +34-85-181986.

E-mail addresses: luciano@lsi.uniovi.es (L. Sánchez), casillas@decsai.ugr.es (J. Casillas), ocordon@decsai.ugr.es (O. Cordón), mjj@ujaen.es (M. José del Jesus).

0888-613X/02/$ - see front matter © 2002 Published by Elsevier Science Inc.
PII: S0888-613X(01)00063-9

1. Introduction

Descriptive fuzzy models and classifiers depend on linguistic variables [23,24], which, in turn, are related to fuzzy partitions of input and output variables. Fuzzy rule learning algorithms induce rules and partitions from examples, so both a linguistic description and the meaning of every term in it are obtained.

It is commonly assumed that some precision must be sacrificed to achieve interpretability and thus black-box models and classifiers are more precise than their understandable fuzzy counterparts. But, when designing fuzzy models, one does not always have a reference about the error that the understandable model should have and thus, it is difficult to judge how far the learning algorithm is from doing its best. Many times, fuzzy models are compared to models with a different structure (statistical, neural networks, etc.) but this comparison is relevant only when both methods perform similarly. If not, it cannot be said whether the difference is due to a failure in the learning algorithm or it is inherent to the problem. To solve this problem, we propose to use statistical procedures to obtain models and classifiers very similar to their fuzzy counterparts, so that their performance gives a reference about the minimum quality that should be obtained when designing linguistically understandable fuzzy models or classifiers.

This paper is organized as follows. First, we define statistical classifier systems and regression models. Then we establish the meaning of "linguistically understandable" classifiers or models, and propose simple methods for estimating these classifiers and models' parameters from a sample. Finally, we compare the output of these algorithms with the output of some usual fuzzy rule learning algorithms and discuss the advantages of both approaches.

2. Statistical classifiers and models

2.1. *Definition of a statistical classification problem*

Let us suppose we have a set Ω that contains objects ω, each one of them belonging to a class c_i, $i = 1, \ldots, N_c$, and we perform the set of measurements $X(\omega) = (X_1(\omega), \ldots, X_{N_i}(\omega))$ over every object. Let us also assume that the mapping X fulfills all necessary conditions to be a random variable. We will say that a classification system is a decision rule that maps every element of $X(\Omega)$ to a class c_i, whose main objective is to produce a low number of errors. Alternative objectives could also be used: for instance, the decision rule could be expressed in a natural language (e.g., by using if–then rules) or we could search for decision rules whose algorithmic expression is short enough, to save computation time when making a decision.

We will limit ourselves to the main objective, by now. For example, let Ω be a set of fruits: apples (c_1), pears (c_2) or bananas (c_3). We observe the weight and the color of a randomly selected fruit, for example, $X(\omega) = (\texttt{yellow}, 150)$. Our classification system relates the pair $(\texttt{yellow}, 150)$ to the class c_3, and we wish this relation to be true most of the times (i.e., most of the yellow fruits that weight 150 g are bananas).

Since we did not assume that $\omega_1 \neq \omega_2 \Rightarrow X(\omega_1) \neq X(\omega_2)$ (i.e., we admit that there can exist a yellow pear weighting 150 g) perhaps a decision rule that never fails cannot be defined for this problem. But an optimum classifier can be defined with respect to the average number of errors. Usually we evaluate the expectation of a new random variable that quantifies the cost of assigning the class c_i to an object when it belongs to class c_j, $\mathrm{cost}(i,j)$. If the classifier is a decision rule, $D(X)$, and "class(ω)" is the class of the object ω then the error is

$$\mathrm{err}(D) = \int_\Omega \mathrm{cost}(D(X(\omega)), \mathrm{class}(\omega))\, \mathrm{d}P.$$

If we choose $\mathrm{cost}(i,j) = 1$ when $i \neq j$ and 0 else, the expectation of the cost function is the mean number of errors. This rule is called "minimum error Bayes rule" and the optimum classifier is [9]

$$D(\mathbf{x}) = \arg \max_{i=1,\ldots,N_\mathrm{c}} P(\mathrm{class}(\omega) = c_i \,|\, X = \mathbf{x}).$$

Observe that an algorithm to set up a decision rule from a sample was not given. It was merely stated that if objects are randomly selected and the minimum error Bayes rule is used, then the optimum classifier has this form.

2.2. Definition of a regression problem

Let us suppose again we have a set Ω that contains objects ω, and we perform the set of observations

$$Z(\omega) = (X(\omega), Y(\omega)) = (X_1(\omega), X_2(\omega), \ldots, X_{N_i}(\omega), Y_1(\omega), \ldots, Y_{N_\mathrm{o}}(\omega)),$$

where Z is a random variable. The regression function is a mapping r that approximates the value of "Y" measurements (outputs) with the images of "X" measurements (inputs). We will admit that the best approximation is the one that minimizes the mean difference between Y and $r(X)$,

$$E(r) = \int_\Omega \|Y(\omega) - r(X(\omega))\|\, \mathrm{d}P.$$

If we define $\|Y(\omega) - r(X(\omega))\| = (Y(\omega) - r(X(\omega)))^\mathrm{T}(Y(\omega) - r(X(\omega)))$ it is well known that the solution is the conditional expectation $E(Y\,|\,X)$ and, in this particular case,

$$r(\mathbf{x}) = E(Y\,|\,X = \mathbf{x}).$$

Once the general expression for the regression function or *model* is given, inducting it from a sample of objects can be formulated again as a parametric problem (i.e., linear regression, etc.) or as a non-parametric estimation of the conditional density functions $f(\mathbf{y}\,|\,\mathbf{x})$, making

$$r(\mathbf{x}) = E(Y\,|X = \mathbf{x}) = \int_{Y(\Omega)} \mathbf{y} f(\mathbf{y}\,|\,\mathbf{x})\,\mathrm{d}\mathbf{y}.$$

Alternatively, if Ω contains outliers, it is common to define a robust estimator like

$$r(\mathbf{x}) = \arg\max_{Y(\Omega)} f(\mathbf{y}\,|\,\mathbf{x}).$$

When $Y(\omega)$ is a discrete set, robust regression and classification are the same problem.

3. A linguistically understandable statistical classifier

Sometimes it is needed to obtain a linguistically understandable classifier. This means we need to set up a decision rule that can be codified in a language that allows it to be linguistically communicated. Fuzzy logic techniques allow the obtention of such classifiers [25].

The semantic of a fuzzy classifier depends on the equivalence between linguistic values of attributes and certain fuzzy sets defined over the range of every feature, and on a fuzzy inference procedure. For example, the sentence [6]

$$\texttt{if } \mathbf{x} \texttt{ is } \tilde{A} \texttt{ then class} = (c_1 \texttt{ with conf } p_1, \ldots, c_{N_c} \texttt{ with conf } p_{N_c})$$

means

$$\mathrm{truth}(\tilde{A} \to c_1) = p_1, \ldots, \mathrm{truth}(\tilde{A} \to c_{N_c}) = p_{N_c},$$

where the concept "$\tilde{A}$" is linked to a fuzzy subset of the feature space.

The same sentence can be given a probabilistic meaning [20]: if A is a *crisp* subset of $X(\Omega)$, then the sentence means

$$P(c_1\,|A) = p_1, \ldots, P(c_{N_c}\,|A) = p_{N_c},$$

with $\sum_{i=1}^{N_c} p_i = 1$. The probabilistic logic-based view has some advantages. It can be used to give a statistical meaning to the process of inducting a rule-based classifier from examples, as we will show below. But it is not immediate to compare it to fuzzy logic-based rules, in which "$\tilde{A}$" is a fuzzy set. On the contrary, we will show that fuzzy classifiers can be compared to certain random set-based classifiers, which in turn are defined as the expectation of the probabilistic logic-based ones for a given sample distribution.

3.1. Relationship between fuzzy and probabilistic classifiers

Suppose we have a machine learning procedure to estimate a crisp partition $\{A_j\}_{j=1,\dots,N_r}$ of the feature space, $A_j \cap A_k = \phi$ for $j \neq k$, plus the values $P(c_i \mid \mathbf{x} \in A_j) = p_{ij}$ that define a probabilistic logic-based classifier. The machine learning task takes as input a random sample $\mathbf{X}$ of classified examples and produces a partition $\{A_1^{\mathbf{X}}, A_2^{\mathbf{X}}, \dots\}$ and the values $P(c_i \mid \mathbf{x} \in A_j^{\mathbf{X}}) = p_{ij}^{\mathbf{X}}$. In turn, for an input value $\mathbf{x}$, the classifier that was learned from the sample $\mathbf{X}$ outputs the probabilities of all classes according to the formula

$$P(c_i \mid \mathbf{x}, \mathbf{X}) = \sum_j p_{ij}^{\mathbf{X}} A_j^{\mathbf{X}}(\mathbf{x}), \tag{1}$$

where $A_j^{\mathbf{X}}(\mathbf{x})$ is 1 if $\mathbf{x} \in A_j^{\mathbf{X}}$ and 0 else.

It is well known that the expected error of this classifier (we will use the notation $\langle\,\rangle_{\mathbf{X}}$ to denote expectation with respect to the sample distribution of $\mathbf{X}$) is the sum of two positive terms, *bias* plus *variance*, where the bias is the error of the average classifier

$$P(c_i \mid \mathbf{x}) = \left\langle \sum_j p_{ij}^{\mathbf{X}} A_j^{\mathbf{X}}(\mathbf{x}) \right\rangle_{\mathbf{X}}. \tag{2}$$

Since the variance term is positive, the error of this average classifier is lower than the mean error of the individual classifiers. We can reorder the terms in (2):

$$P(c_i \mid \mathbf{x}) = \sum_j \left\langle p_{ij}^{\mathbf{X}} A_j^{\mathbf{X}}(\mathbf{x}) \right\rangle_{\mathbf{X}}, \tag{3}$$

and if the random variables $p_{ij}^{\mathbf{X}}$ and $A_j^{\mathbf{X}}(\mathbf{x})$ (both defined with respect to $\mathbf{X}$) were independent,

$$P(c_i \mid \mathbf{x}) = \sum_j \left\langle p_{ij}^{\mathbf{X}} \right\rangle_{\mathbf{X}} \left\langle A_j^{\mathbf{X}}(\mathbf{x}) \right\rangle_{\mathbf{X}} = \sum_j \left\langle p_{ij}^{\mathbf{X}} \right\rangle_{\mathbf{X}} \Phi_j(\mathbf{x}), \tag{4}$$

where $\Phi_j(\mathbf{x})$ is the one-point coverage function of the random set $A_j^{\mathbf{X}}$. Observe that $\sum_j A_j^{\mathbf{X}}(\mathbf{x}) = 1$ for all $\mathbf{x}$, so $\langle \sum_j A_j^{\mathbf{X}}(\mathbf{x}) \rangle_{\mathbf{X}} = 1$ and then $\sum_j \Phi_j = 1$ for all values of $\mathbf{x}$ (in words, the sum of the memberships is always equal to 1).

Expression (4) is very similar to the fuzzy inference formula applied to a fuzzy classifier defined by the rules "truth$(\tilde{A}_j \to c_i) = t_{ij}$", $j = 1, \dots, N_r$:

$$\text{truth}(c_i) = \bigvee_j \text{truth}(\tilde{A}_j \to c_i) \wedge \tilde{A}_j(\mathbf{x}) = \bigvee_j t_{ij} \wedge \tilde{A}_j(\mathbf{x}). \tag{5}$$

The truth value t_{ij} is the counterpart of the value $\langle p_{ij}^{\mathbf{X}} \rangle_{\mathbf{X}}$ and the one-point coverage function $\Phi_j(\mathbf{x})$ is related to the membership function of the fuzzy set $\tilde{A}_j(\mathbf{x})$. The *t*-norm $\wedge$ is replaced by the product, and the *t*-conorm $\vee$ by the sum.

The value $\langle p_{ij}^{\mathbf{X}} \rangle_{\mathbf{X}}$ is the expected probability of class i given the set $\hat{A}_j^{\mathbf{X}}$, this is the mean value of $P(c_i | A_j^{\mathbf{X}})$ for all probabilistic logic-based classifiers with respect to the sample distribution. It can be regarded as the degree of truth of the assertion "All elements in A_j belong to class i". The function $\Phi_j(\mathbf{x})$ is the probability of $\mathbf{x}$ being covered by the random set $A_j^{\mathbf{X}}$, and thus can be associated to the truth of the assertion "$\mathbf{x}$ belongs to A_j". It is possible to assign linguistic labels to the random sets $A_i^{\mathbf{X}}$ and draw their coverage functions in a form that closely resemble a Ruspini fuzzy partition (see Fig. 1). In Fig. 2 the three types of classifiers that have been discussed (probabilistic, random sets based and fuzzy sets based) are represented along with their inference procedures.

3.2. Estimating the classifier from a sample

Let us suppose we know the antecedents of the rules (the functions "truth(A_j is $\mathbf{x}$)" or Φ_j, $j = 1, \ldots, N_r$) and we wish to estimate their consequents (the values "truth($A_j \rightarrow c_i$)" or $\langle p_{ij}^{\mathbf{X}} \rangle_{\mathbf{X}}$). For simplicity in the notation, let $\theta_{ij} = \langle p_{ij}^{\mathbf{X}} \rangle_{\mathbf{X}}$ and Θ be the parameter vector of all unknown parameters,

$$\Theta = \left(\langle p_{11}^{\mathbf{X}} \rangle_{\mathbf{X}}, \ldots, \langle p_{1N_r}^{\mathbf{X}} \rangle_{\mathbf{X}}, \ldots, \langle p_{N_c N_r}^{\mathbf{X}} \rangle_{\mathbf{X}}\right) = (\theta_{11}, \ldots, \theta_{1N_r}, \ldots, \theta_{N_c N_r}). \tag{6}$$

The verosimility function V_θ is defined as follows:

$$V_\theta = \prod_{\mathbf{x} \in \mathbf{X}} \sum_{j=1}^{N_r} \theta_{\mathrm{class}(\mathbf{x}),j} \Phi_j(\mathbf{x}) \tag{7}$$

and its maximum, restricted to the conditions $\sum_{i=1}^{N_c} \theta_{ij} = 1$ for $j = 1, \ldots, N_r$ and $\theta_{ij} \geqslant 0$, is a good estimation of the unknowns.

It is easier to work with the logarithm of this function

$$L(\Theta) = \log(V_\theta) = \sum_{\mathbf{x} \in \mathbf{X}} \log \sum_{j=1}^{N_r} \theta_{\mathrm{class}(\mathbf{x}),j} \Phi_j(\mathbf{x}). \tag{8}$$

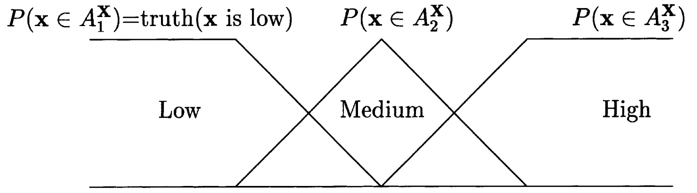
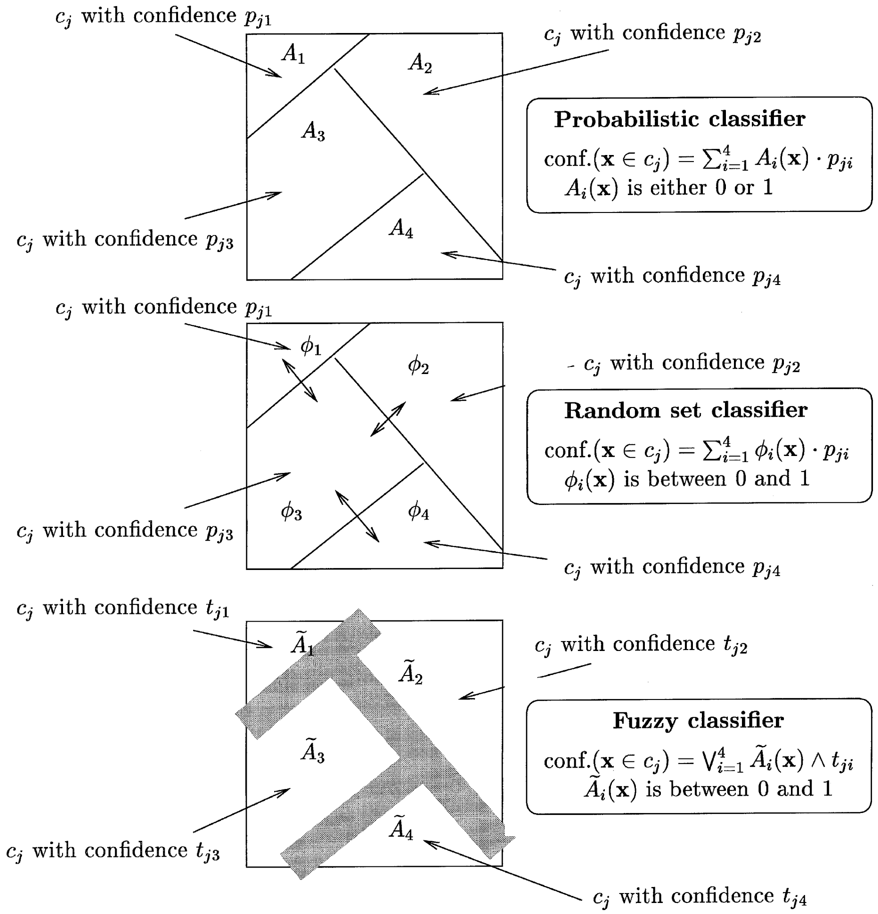

$P(\mathbf{x} \in A_1^{\mathbf{X}})$=truth($\mathbf{x}$ is low)
$P(\mathbf{x} \in A_2^{\mathbf{X}})$
$P(\mathbf{x} \in A_3^{\mathbf{X}})$
Low
Medium
High

Fig. 1. Graph of the degrees of truth of a numeric value being compatible with the properties "low", "medium" and "high". These labels are associated to random sets $A_1^{\mathbf{X}}$, $A_2^{\mathbf{X}}$ and $A_3^{\mathbf{X}}$, respectively, and the degree of truth of the assertion "$\mathbf{x}$ is label" is the probability of $\mathbf{x}$ being covered by the corresponding random set. For example: truth("80 is low") = $P(80 \in A_1^{\mathbf{X}})$.



c_j with confidence p_{j1}
c_j with confidence p_{j2}
c_j with confidence p_{j3}
c_j with confidence p_{j4}
A_1
A_2
A_3
A_4
Probabilistic classifier
conf.$(\mathbf{x} \in c_j) = \sum_{i=1}^4 A_i(\mathbf{x}) \cdot p_{ji}$
$A_i(\mathbf{x})$ is either 0 or 1
ϕ_1
ϕ_2
ϕ_3
ϕ_4
Random set classifier
conf.$(\mathbf{x} \in c_j) = \sum_{i=1}^4 \phi_i(\mathbf{x}) \cdot p_{ji}$
$\phi_i(\mathbf{x})$ is between 0 and 1
c_j with confidence t_{j1}
c_j with confidence t_{j2}
c_j with confidence t_{j3}
c_j with confidence t_{j4}
$\tilde{A}_1$
$\tilde{A}_2$
$\tilde{A}_3$
$\tilde{A}_4$
Fuzzy classifier
conf.$(\mathbf{x} \in c_j) = \bigvee_{i=1}^4 \tilde{A}_i(\mathbf{x}) \wedge t_{ji}$
$\tilde{A}_i(\mathbf{x})$ is between 0 and 1

Fig. 2. Three types of classifiers are used in this paper: probabilistic classifiers output degrees of confidence depending on a crisp partition; the output of random set classifiers is the expectation of probabilistic classifiers' and confidences are added after multiplying them by the coverage functions; fuzzy sets use t-norms and t-conorms instead of products and sums. We will compare random set classifiers with fuzzy classifiers in experiments for which fuzzy memberships and coverage functions are numerically identical.

Using Lagrange multipliers to cope with the restrictions, we have to minimize

$$L_1(\Theta, \lambda_1, \ldots, \lambda_{N_r}) = \sum_{\mathbf{x} \in \mathbf{X}} \log \sum_{j=1}^{N_r} \theta_{\text{class}(\mathbf{x}),j} \Phi_j(\mathbf{x}) + \sum_{j=1}^{N_r} \lambda_j \left(1 - \sum_{i=1}^{N_c} \theta_{ij} \right). \quad (9)$$

Taking derivatives with respect to θ_{ij} and λ_j, we obtain the following conditions that are true in the minimum:

$$\sum_{\substack{\mathbf{x}\in\mathbf{X}\\ \text{class}(\mathbf{x})=i}} \frac{\Phi_j(\mathbf{x})}{\sum_{j=1}^{N_r}\theta_{ij}\Phi_j(\mathbf{x})} = \sum_{\substack{\mathbf{x}\in\mathbf{X}\\ \text{class}(\mathbf{x})=k}} \frac{\Phi_j(\mathbf{x})}{\sum_{j=1}^{N_r}\theta_{kj}\Phi_j(\mathbf{x})}, \tag{10}$$

$$\sum_{i=1}^{N_c}\theta_{ij} = 1 \quad \text{for } j = 1,\dots,N_r;\ \ i,k = 1,\dots,N_c. \tag{11}$$

The first set of equalities (Eq. (10)) produces $N_c - 1$ equations and the second one (Eq. (11)) leads to N_r, thus the search of the parameters consists in numerically solving a system of $N_c \cdot N_r$ nonlinear equations.

The pseudocode of the procedure we used to find the solution is shown in Fig. 3. Observe the similarities and differences between gradient–descent-based rule learning methods: this is a constrained minimization and the objective function *is not* the classification error over a sample, but a function of the classification margins of all examples, which can be related to boosting algorithms [18].

Whilst fuzzy rule banks can be incomplete, probabilistic ones cannot. All consequents are initialized to the value $1/N_c$ to express the initial absence of knowledge. If the learning algorithm finishes and there are still rules like these, they can be skipped when doing an inference (Eq. (4)) without affecting the output of the classifier. We can define an "equivalent number of rules" to

```
minimize($\theta \in \mathbf{R}^{N_c\times N_r}$)
$\mathbf{G},\mathbf{D} \in \mathbf{R}^{N_c\times N_r}$, $\alpha \in \mathbf{R}$, $i \in 1\dots N_c$, $j \in 1\dots N_r$
$\theta_{ij} = 1/N_c$
repeat
    $\mathbf{G}_{ij} = \sum_{\mathbf{x}\in\mathbf{X},\text{class}(\mathbf{x})=i}\Phi_j(\mathbf{x})/\sum_{j=1}^{N_r}\theta_{ij}\Phi_j(\mathbf{x})$
    $\mathbf{D}_{ij} = \mathbf{G}_{ij} - (N_r \cdot N_c)^{-1}\sum_{i=1}^{N_c}\sum_{j=1}^{N_r}\mathbf{G}_{ij}$
    search $\alpha$ that minimizes $L$(normalize($\theta + \alpha\cdot\mathbf{D}$))
    $\theta$ = normalize($\theta + \alpha\cdot\mathbf{D}$)
until $\alpha||D|| < \epsilon$
end of minimize
normalize($\mathbf{X} \in \mathbf{R}^{N_c\times N_r}$)
    if ($\mathbf{X}_{ij} < 0$) $\mathbf{X}_{ij} = 0$
    $\mathbf{X}_{ij} = \mathbf{X}_{ij}/\sum_{i=1}^{N_c}\mathbf{X}_{ij}$
end of normalize
```

Fig. 3. Pseudocode of the numerical algorithm used to solve the set of equations (11). The linear search (determination of the value of α) was implemented with Brent's method. All points examined fulfill (11) because of the function `normalize`, and the algorithm stops when the conditions (10) are approximately true.

compare the complexity of a complete probabilistic rule bank to that of an incomplete fuzzy rule bank as the number of rules for which the degree of confidence in any of their consequent parts is different from $1/N_c$.

Determining which set of $N_r' < N_r$ rules produces the best classifier has practical relevance. When the number of features is high, the number of parameters grows above all practical linguistic interpretability, and then it is useful to decide whether an approximate solution in which most of the parameters are $1/N_c$ (thus they can be ignored) is precise enough. Moreover, many fuzzy rule learning algorithms produce incomplete rule banks and it is reasonable to compare them to random set-based classifiers with the same structure, but also with the same "equivalent number of rules".

As far as we know, finding the best set of N_r' rules in polynomial time is an open problem. A heuristic method to obtain banks with a reduced number of rules is shown in Fig. 4. This algorithm does not guarantee that there is not a different set of rules with higher verosimility, but has good properties in practical problems. It just selects rules containing the parameters for which the partial derivatives of Eq. (9) are higher in the initial point of the minimization ($\theta_{ij} = 1/N_c$).

```
minimize($\theta \in \mathbf{R}^{N_c \times N_r}$)
$\mathbf{G}, \mathbf{D} \in \mathbf{R}^{N_c \times N_r}$, $\alpha \in \mathbf{R}$, $i \in 1 \ldots N_c$, $j \in 1 \ldots N_r$, $\mathbf{V} \in \mathcal{P}(\mathbf{N})$
$\theta_{ij} = 1/N_c$
$\mathbf{G}_{ij} = N_c \sum_{\mathbf{x} \in \mathbf{X}, \text{class}(\mathbf{x})=i} \Phi_j(\mathbf{x}) / \sum_{j=1}^{N_r} \Phi_j(\mathbf{x})$
$\mathbf{D}_{ij} = \mathbf{G}_{ij} - (N_r \cdot N_c)^{-1} \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} \mathbf{G}_{ij}$
importance($j$) $= \sum_{\substack{i=1 \ldots N_c \\ \mathbf{D}_{ij} > 0}} \mathbf{D}_{ij}$
$\mathbf{V} =$ set of indices $j$ of the $N_r'$ rules with higher importance
repeat
    $\mathbf{G}_{ij} = \sum_{\mathbf{x} \in \mathbf{X}, \text{class}(\mathbf{x})=i} \Phi_j(\mathbf{x}) / \sum_{j=1}^{N_r} \theta_{ij} \Phi_j(\mathbf{x})$
    $\mathbf{D}_{ij} = \mathbf{G}_{ij} - (N_r \cdot N_c)^{-1} \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} \mathbf{G}_{ij}$
    if $j \notin \mathbf{V}$ then $\mathbf{D}_{ij} = 0$
    search $\alpha$ that minimizes $L$(normalize($\theta + \alpha \cdot \mathbf{D}$))
    $\theta =$ normalize($\theta + \alpha \cdot \mathbf{D}$)
until $\alpha ||D|| < \epsilon$
end of minimize
normalize($\mathbf{X} \in \mathbf{R}^{N_c \times N_r}$)
    if ($\mathbf{X}_{ij} < 0$) $\mathbf{X}_{ij} = 0$
    $\mathbf{X}_{ij} = \mathbf{X}_{ij} / \sum_{i=1}^{N_c} \mathbf{X}_{ij}$
end of normalize
```

Fig. 4. Pseudocode of the numerical algorithm that uses a heuristic method to learn a bank comprising at most N_r' rules.

3.3. Approximate and descriptive rules

N_c rules of the form $\text{truth}(\tilde{A}_j \to c_k) = p_{kj}$ can be combined into the assert

$$\begin{array}{c}\texttt{if } \mathbf{x} \texttt{ is } \tilde{A}_j \texttt{ then}\\ \texttt{class} = (c_1 \texttt{ with conf } p_{1j}, \ldots, c_{N_c} \texttt{ with conf } p_{N_c j}).\end{array} \tag{12}$$

Unless the concept $\tilde{A}_j$ can be expressed as the conjunction of independent properties defined over every feature, it is difficult to understand the meaning of this last assert; it is easier to use expressions like

$$\begin{array}{c}\texttt{if } x_1 \texttt{ is } \tilde{A}_{j1} \texttt{ and} \cdots \texttt{and } x_{N_i} \texttt{ is } \tilde{A}_{jN_i} \texttt{then}\\ \texttt{class} = (c_1 \texttt{ with conf } p_{1j}, \ldots, c_{N_c} \texttt{ with conf } p_{N_c j}),\end{array} \tag{13}$$

where all fuzzy sets $\tilde{A}_{jk}$, $j = 1, \ldots, N_r$, belong to a given fuzzy partition of the feature k. This is a typical rule structure in the field of fuzzy classifiers. We will call *linguistic* or *descriptive* to classifiers based on expression (13), and *approximate* fuzzy classifiers to those based on expression (12), following the nomenclature in [5].

Not all approximate fuzzy classifiers can be expressed with linguistic classification rules. The conditions they must fulfill are immediate in fuzzy logic

$$\tilde{A}_j(\mathbf{x}) = \tilde{A}_{j1}(x_1) \wedge \cdots \wedge \tilde{A}_{jN_i}(x_{N_i})$$

and exactly the same in probabilistic logic-based rules,

$$\begin{array}{c}\texttt{if } x_1 \texttt{ is } A_{j1} \texttt{ and} \cdots \texttt{and } x_{N_i} \texttt{ is } A_{jN_i} \texttt{then}\\ \texttt{class} = (c_1 \texttt{ with conf } p_{1j}, \ldots, c_{N_c} \texttt{ with conf } p_{N_c j}),\end{array} \tag{14}$$

where the antecedents A_j must be hypercubes in the feature space. If $A_j(\mathbf{x}) = 1$ for all $\mathbf{x} \in A$ and 0 else,

$$A_j(\mathbf{x}) = A_{j1}(x_1) \wedge \cdots \wedge A_{jN_i}(x_{N_i})$$

and all intervals A_{jk} must be elements of the same crisp partition of the feature k.

Recalling Eq. (1), the probabilistic rule-based classifier that was learned from the sample $\mathbf{X}$ outputs the probabilities of all classes according to the formula

$$P(c_i \mid \mathbf{x}, \mathbf{X}) = \sum_j p_{ij}^{\mathbf{X}} \prod_{k=1}^{N_i} A_{jk}^{\mathbf{X}}(x_k), \tag{15}$$

where $\prod_{k=1}^{N_i} A_{jk}^{\mathbf{X}}(x_k)$ is 1 or 0. Operating, we obtain that

$$P(c_i \mid \mathbf{x}) = \sum_j \left\langle p_{ij}^{\mathbf{X}} \right\rangle_{\mathbf{X}} \left\langle \prod_{k=1}^{N_i} A_{jk}^{\mathbf{X}}(x_k) \right\rangle_{\mathbf{X}}. \tag{16}$$

Thus the condition we need to obtain a descriptive random set-based classifier is

$$\Phi_j(\mathbf{x}) = \prod_{k=1}^{N_i} \Phi_{jk}(x_k) = \left\langle \prod_{k=1}^{N_i} A_{jk}^{\mathbf{X}}(x_k) \right\rangle_{\mathbf{X}}, \tag{17}$$

which is fulfilled when random variables $P(x_k \in A_{jk}^{\mathbf{X}})$ are independent. In this last case,

$$\Phi_{jk}(x_k) = \left\langle A_{jk}^{\mathbf{X}}(x_k) \right\rangle_{\mathbf{X}} \tag{18}$$

and the descriptive classifier can then be expressed as a set of rules of the form

$$\texttt{if}\ x_1\ \texttt{is}\ \Phi_{j1}\ \texttt{and}\ \ldots\ \texttt{and}\ x_{N_i}\ \texttt{is}\ \Phi_{jN_i}\ \texttt{then}$$
$$\texttt{class} = (c_1\ \texttt{with conf}\ \langle p_{1j}^{\mathbf{X}} \rangle_{\mathbf{X}}, \ldots, c_{N_c}\ \texttt{with conf}\ \langle p_{N_c j}^{\mathbf{X}} \rangle_{\mathbf{X}}).$$

Selecting a parametric family of coverage functions Φ_{jk} is equivalent to define a family of sample distributions over **X**. It is immediate that all functions Φ_{jk} can be interpreted as elements of a Ruspini's fuzzy partition of feature k. We will use triangular coverage functions in all numerical examples, as shown in Fig. 1.

4. Numerical examples

4.1. Pure linguistic classification problem

For testing the algorithm we first generated a synthetic pure descriptive problem. We defined the following set of rules:

If x_1 is $\tilde{R}_1$ and x_2 is $\tilde{R}_1$ then $\text{class}_1 = 0.90$ and $\text{class}_2 = 0.10$
If x_1 is $\tilde{R}_1$ and x_2 is $\tilde{R}_2$ then $\text{class}_1 = 0.85$ and $\text{class}_2 = 0.15$
If x_1 is $\tilde{R}_1$ and x_2 is $\tilde{R}_3$ then $\text{class}_1 = 0.60$ and $\text{class}_2 = 0.40$
If x_1 is $\tilde{R}_2$ and x_2 is $\tilde{R}_1$ then $\text{class}_1 = 0.40$ and $\text{class}_2 = 0.60$
If x_1 is $\tilde{R}_2$ and x_2 is $\tilde{R}_2$ then $\text{class}_1 = 0.80$ and $\text{class}_2 = 0.20$
If x_1 is $\tilde{R}_2$ and x_2 is $\tilde{R}_3$ then $\text{class}_1 = 0.40$ and $\text{class}_2 = 0.60$
If x_1 is $\tilde{R}_3$ and x_2 is $\tilde{R}_1$ then $\text{class}_1 = 0.20$ and $\text{class}_2 = 0.80$
If x_1 is $\tilde{R}_3$ and x_2 is $\tilde{R}_2$ then $\text{class}_1 = 0.10$ and $\text{class}_2 = 0.90$
If x_1 is $\tilde{R}_3$ and x_2 is $\tilde{R}_3$ then $\text{class}_1 = 0.00$ and $\text{class}_2 = 1.00$

The sets $\tilde{R}_1$, $\tilde{R}_2$ and $\tilde{R}_3$ are shown in Fig. 5. Then we generated two sets of 100 and 1000 examples each, using the algorithm shown in Fig. 6, and applied the algorithm in Fig. 3 to infer the values of the coefficients. After 20 linear searches in both cases (the order of convergence of this algorithm does not depend on the number of examples, but the time needed to calculate the ver-

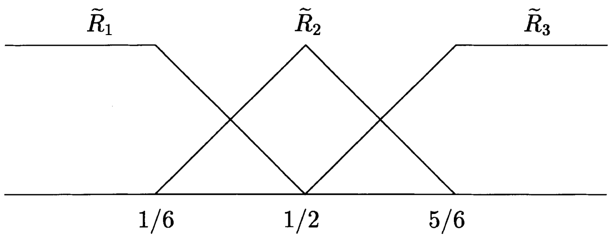


Fig. 5. Membership functions of the example in Section 4.1.

```
$x_1$=random(0,1); $x_2$=random(0,1)
$p_1 = 0.90\tilde{R}_1(x_1)\cdot\tilde{R}_1(x_2) + 0.85\tilde{R}_1(x_1)\cdot\tilde{R}_2(x_2) + 0.60\tilde{R}_1(x_1)\cdot\tilde{R}_3(x_2)+$
      $0.40\tilde{R}_2(x_1)\cdot\tilde{R}_1(x_2) + 0.80\tilde{R}_2(x_1)\cdot\tilde{R}_2(x_2) + 0.40\tilde{R}_2(x_1)\cdot\tilde{R}_3(x_2)+$
      $0.20\tilde{R}_3(x_1)\cdot\tilde{R}_1(x_2) + 0.15\tilde{R}_3(x_1)\cdot\tilde{R}_2(x_2) + 0.00\tilde{R}_3(x_1)\cdot\tilde{R}_3(x_2)$
if (random(0,1)< $p_1$) output $(x_1,x_2,c_1)$ else output $(x_1,x_2,c_2)$
```

Fig. 6. Algorithm used to output a point of the learning sample in the problem discussed in Section 4.1.

0.90	0.40	0.20
0.10	0.60	0.80
0.85	0.80	0.10
0.15	0.20	0.90
0.60	0.40	0
0.40	0.60	1

0.939	0.525	0
0.061	0.574	1
0.945	0.722	0.124
0.054	0.277	0.876
0.606	0.471	0
0.393	0.528	1

0.887	0.345	0.198
0.112	0.654	0.801
0.756	0.744	0.100
0.243	0.255	0.899
0.664	0.354	0
0.335	0.645	1

Fig. 7. True (left) and estimated values for the example explained in Section 4.1. The central table was estimated from 100 samples, the right one from 1000.

osimility function (Eq. (8)) grows linearly with it), the inferred rule banks are summarized in Fig. 7.

4.2. Haykin's two gaussian problems

To make a graphical comparison of this method, we are going to analyze the dataset proposed in [10]: 4000 points taken from two overlapping gaussian distributions with different variances.

5 $\times$ 2cv Dietterich's test [7] was applied to judge the relevance of the differences of the classification methods considered. The sample of 4000 points were randomly permuted first, the first half of samples was used to train the method and the second half to test it and then training and test sets were

swapped and the learning and test phase repeated. This was repeated for five different random permutations. Training errors were discarded so that boxplots show the dispersion of the error in test sets. The mean of test errors are tabulated for the experiment.

The optimal decision surface is a circle, and the bayesian test error is 0.185. The error of the linear classifier is 0.24, which is near enough the optimal solution to confuse many rule learning algorithms. The shape of the decision surface in areas with low density of examples (i.e., the left side of the circle) does not contribute too much to the classification error. For example, the neural network shown in Fig. 8 has an error of 0.20, which is near the bayesian error, but its decision surface is wrong in the left part.

In Figs. 8–11 this problem has been solved with some black-box methods (linear, quadratic, 1-NN, neuronal) and descriptive rule-based methods: Wang and Mendel's (WM) [21], Ishibuchi's (ISH) [11], Pal and Mandal's (PM) [14] and random set-based (RSB). This is not a problem well suited to linguistic classifiers, so it is expected that rule-based methods perform worse than statistical ones. The optimal solution in this case is the quadratic one.

The p-values in the 5×2cv Dietterich's test (Fig. 11) allow us to statistically compare the performance of the different algorithms. Low values (< 0.05) mean that we reject that two algorithms are the same, being better than those with a better mean (a lower value in Fig. 10). Statistical methods like this cannot assess the differences when one of the methods has high variance in its results (most of the fuzzy learning methods do) and should be completed with a boxplot (see Fig. 9).

In Fig. 12, descriptive classifiers are compared when the number of linguistic terms in every partition ranges from 4 to 7. In this figure we observe that the decision surface of the descriptive random set-based classifier tends to the optimum when the number of labels is allowed to increase, as we expected.

Later, we will compare all these classifiers over different datasets following the same comparison methodology.

5. A linguistically understandable statistical model

There are different definitions of fuzzy rule-based models. Consequents can be fuzzy sets, real numbers or hyperplanes [22]. Fuzzy rules in which the consequent is a real number can be converted to linguistic rules in some cases [13] but the latter type [19] cannot. We will adopt the same nomenclature we used in fuzzy classifiers and define an approximate rule-based model as a set of N_r rules of the form

$$\begin{gathered}\texttt{if } \mathbf{x} \texttt{ is } \tilde{A}_j \texttt{ then} \\ \texttt{output} = (\tilde{B}_1 \texttt{ with conf } t_{1j}, \ldots, \tilde{B}_{N_o} \texttt{ with conf } t_{N_o j}).\end{gathered} \tag{19}$$

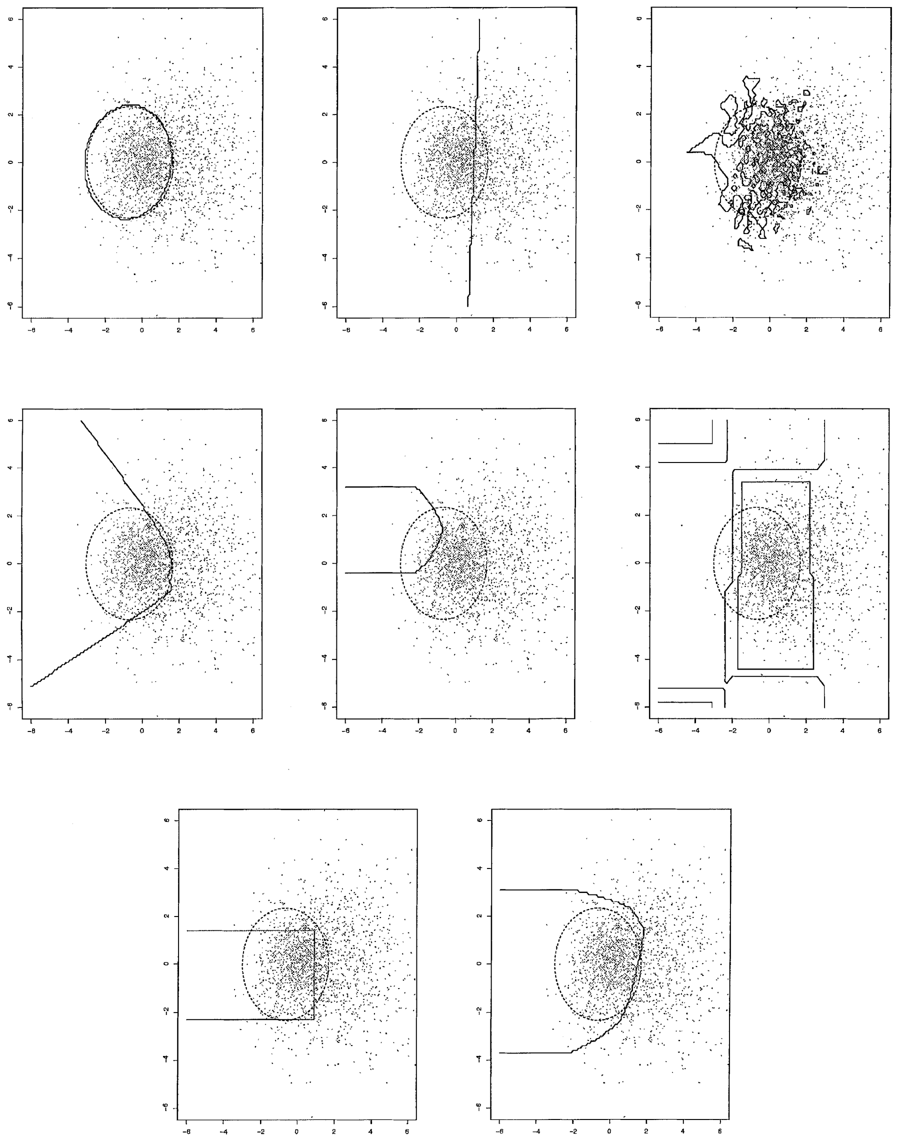


Fig. 8. From left to right, Upper part: decision surfaces of the quadratic (optimal for this problem) and two black-box classifiers (linear and nearest neighbor) inducted for the problem described in the text. Middle: neural network, fuzzy rule-based classifiers inducted by Wang and Mendel [21] and Pal and Mandal's method. Lower part: fuzzy rule-based classifier inducted by Ishibuchi [11], and probabilistic rule-based as proposed in this paper. All rule-based classifiers are of "descriptive" type and four linguistic terms by feature were used. The dashed circle is the optimal decision surface.

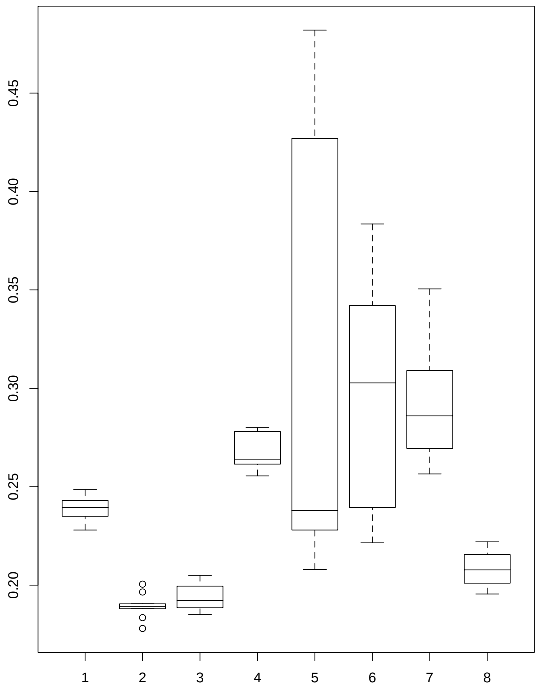


Fig. 9. Comparative performance of classification systems. The boxplots contain the minimum and maximum of test error, the median, 25% and 75% percentiles according to 5×2cv comparison. From left to right: linear, quadratic, neuronal, 1-NN, WM, ISH, PM and RSB classifiers.

Method	Error
Linear	0.2393
Quadratic	0.1894
Neuronal	0.1938
1-NN	0.2667
Fuzzy WM	0.3103
Fuzzy ISH	0.2983
Fuzzy PM	0.2931
RSB	0.2085

Fig. 10. Mean value of classification error rates, estimated from the test sets of 5×2cv method.

Linguistic or descriptive fuzzy models must fulfill additional properties. We will return to this point later.

A probabilistic rule-based model is identical to the fuzzy one except for the antecedents being crisp sets and the interpretation given to them

$$\begin{gathered}\texttt{if } \mathbf{x} \texttt{ is } A_j \texttt{ then} \\ \texttt{output} = (B_1 \texttt{ with conf } p_{1j}, \ldots, B_{N_o} \texttt{ with conf } p_{N_o j}).\end{gathered} \tag{20}$$

	Quad	Neu	1-NN	WM	ISH	PM	RSB
Lin	0.0029	0.00038	0.014	0.28	0.96	0.059	0.17
Quad		0.81	0.0013	0.18	0.54	0.021	0.025
Neu			0.00020	0.18	0.55	0.021	0.069
1-NN				0.38	0.67	0.14	0.038
WM					0.42	0.62	0.22
ISH						0.41	0.86
PM							0.029

Fig. 11. p-Values in 5×2cv test for the Haykin problem.

In addition, $\sum_{i=1}^{N_o} p_{ij} = 1$ for $j = 1, \ldots, N_r$. In our framework, the only difference between a linguistic model and a linguistic classifier is in their consequents: they are integer numbers in a classifier, but elements of a fuzzy or crisp partition of the output space in fuzzy or probabilistic models, respectively.

We will interpret the output of the probabilistic model with the help of a density function with constant value over each set B_k like the one shown in Fig. 13. The values p_{ij} are the probability masses of every element of the output partition

$$P(\mathbf{y} \in B_i \mid \mathbf{x}) = \sum_j p_{ij} A_j(\mathbf{x}), \tag{21}$$

and the density function is

$$f(\mathbf{y} \mid \mathbf{x}) = \sum_i \frac{B_i(\mathbf{y})}{\sigma(B_i)} \sum_j p_{ij} A_j(\mathbf{x}), \tag{22}$$

where $\sigma(B_i)$ is the measure of the set B_i and $B_i(\mathbf{y})$ is 1 if $\mathbf{y} \in B_i$ and 0 otherwise. Let us extend now the relationship between probabilistic and random set-based classifiers, that was introduced in Section 3.1, to the modeling problem. Let us suppose that a machine learning procedure produces $A_j^{\mathbf{X}}$, $p_{ij}^{\mathbf{X}}$ and $B_i^{\mathbf{X}}$. The density function that the model learned from the sample $\mathbf{X}$ outputs is

$$f^{\mathbf{X}}(\mathbf{y} \mid \mathbf{x}) = \sum_i \left(\frac{B_i(\mathbf{y})}{\sigma(B_i)} \right)^{\mathbf{X}} \sum_j p_{ij}^{\mathbf{X}} A_j^{\mathbf{X}}(\mathbf{x}). \tag{23}$$

The average value of $f^{\mathbf{X}}(\mathbf{y} \mid \mathbf{x})$ is

$$\langle f^{\mathbf{X}}(\mathbf{y} \mid \mathbf{x}) \rangle_{\mathbf{X}} = \sum_i \left\langle \left(\frac{B_i(\mathbf{y})}{\sigma(B_i)} \right)^{\mathbf{X}} \sum_j p_{ij}^{\mathbf{X}} A_j^{\mathbf{X}}(\mathbf{x}) \right\rangle_{\mathbf{X}} \tag{24}$$

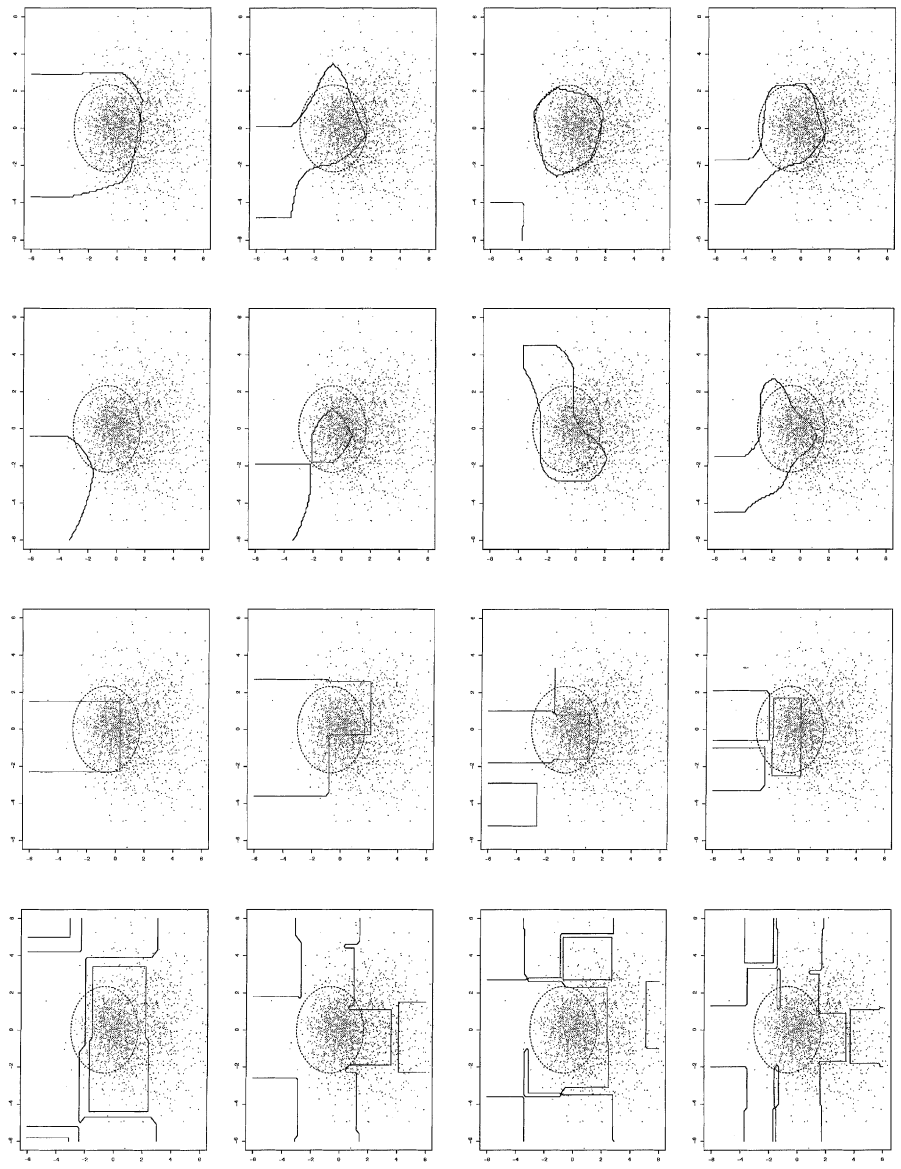


Fig. 12. Effect of the number of elements in the partition: from upper to lower. All parts, from left to right: decision surfaces induced by RSB, fuzzy WM, fuzzy ISH and fuzzy PM rule-based classifiers with 4, 5, 6 and 7 terms/feature. The dashed line is the optimal decision surface.

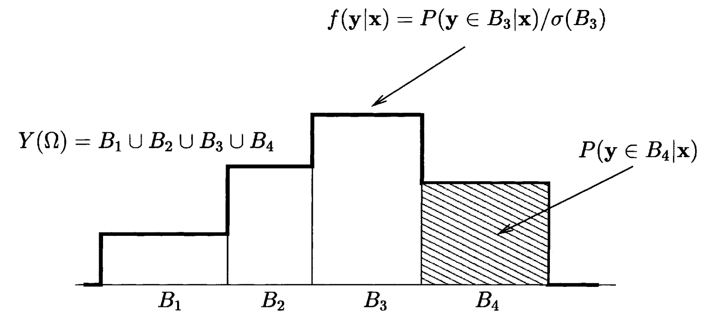


Fig. 13. The output of a probabilistic rule-based model can be regarded as a simple density function. $\sigma(B)$ is the measure of the set B.

assuming that the random variables $(B_i(\mathbf{y})/\sigma(B_i))^{\mathbf{X}}$, $p_{ij}^{\mathbf{X}}$ and $A_j^{\mathbf{X}}(\mathbf{x})$ are independent,

$$\begin{aligned}\langle f^{\mathbf{X}}(\mathbf{y}\,|\,\mathbf{x})\rangle_{\mathbf{X}} &= \sum_i \left\langle \left(\frac{B_i(\mathbf{y})}{\sigma(B_i)}\right)^{\mathbf{X}} \right\rangle_{\mathbf{X}} \sum_j \left\langle p_{ij}^{\mathbf{X}} \right\rangle_{\mathbf{X}} \left\langle A_j^{\mathbf{X}}(\mathbf{x}) \right\rangle_{\mathbf{X}} \\ &= \sum_i \Gamma_i(\mathbf{y}) \sum_j \left\langle p_{ij}^{\mathbf{X}} \right\rangle_{\mathbf{X}} \Phi_j(\mathbf{x}),\end{aligned} \tag{25}$$

where $\sum_j \Phi_j(\mathbf{x}) = 1$, $\sum_i \langle p_{ij}^{\mathbf{X}} \rangle_{\mathbf{X}} \Phi_j(\mathbf{x}) = 1$ and $\int_{Y(\Omega)} \Gamma_i(\mathbf{y})\,\mathrm{d}\mathbf{y} = 1$. The expected value of this density function is also a density function,

$$\int_{Y(\Omega)} \langle f^{\mathbf{X}}(\mathbf{y}\,|\,\mathbf{x})\rangle_{\mathbf{X}}\,\mathrm{d}\mathbf{y} = \sum_i \sum_j \left\langle p_{ij}^{\mathbf{X}} \right\rangle_{\mathbf{X}} \Phi_j(\mathbf{x}) \int_{Y(\Omega)} \Gamma_i(\mathbf{y})\,\mathrm{d}\mathbf{y} = 1, \tag{26}$$

so it makes sense to define

$$r(\mathbf{x}) = \int_{\mathbf{Y}(\Omega)} \mathbf{y} \sum_i \Gamma_i(\mathbf{y}) \sum_j \langle p_{ij}^{\mathbf{X}} \rangle_{\mathbf{X}} \Phi_j(\mathbf{x})\,\mathrm{d}\mathbf{y} = \sum_j \Phi_j(\mathbf{x}) \sum_i \left\langle p_{ij}^{\mathbf{X}} \right\rangle_{\mathbf{X}} M_i, \tag{27}$$

where M_i is the center of gravity of the area under $\Gamma_i(\mathbf{y})$:

$$M_i = \int_{\mathbf{Y}(\Omega)} \mathbf{y}\Gamma_i(\mathbf{y})\,\mathrm{d}\mathbf{y}. \tag{28}$$

Observe the similarities between Eq. (27) and the output resulting from fuzzy inference in the fuzzy model (19) followed by a center of gravity defuzzification. $\Phi_j(\mathbf{x})$ corresponds to the membership function $\tilde{A}_j(\mathbf{x})$, and $\Gamma_i(\mathbf{y})$ corresponds to $\tilde{B}_i(\mathbf{y})$ divided by its area.

5.1. Estimating the model from a sample

Let us suppose we know the functions Φ_j and Γ_i, and we wish to infer the values of the parameters $\langle p_{ij}^{\mathbf{X}} \rangle_{\mathbf{X}}$ from a sample $\mathbf{X} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots\}$. Let Θ be the vector of unknown parameters

$$\Theta = \left(\langle p_{11}^{\mathbf{X}} \rangle_{\mathbf{X}}, \ldots, \langle p_{1N_r}^{\mathbf{X}} \rangle_{\mathbf{X}}, \ldots, \langle p_{N_\mathrm{o}N_r}^{\mathbf{X}} \rangle_{\mathbf{X}}\right) = (\theta_{11}, \ldots, \theta_{1N_r}, \ldots, \theta_{N_\mathrm{o}N_r}). \tag{29}$$

We want to maximize the following verosimility function:

$$L(\Theta) = \sum_{(\mathbf{x}_k,\mathbf{y}_k)\in\mathbf{X}} \log \sum_j \Phi_j(\mathbf{x}_k) \sum_i \theta_{ij}\Gamma_i(\mathbf{y}_k) \tag{30}$$

restricted to the conditions $\sum_i \theta_{ij} = 1$, $\theta_{ij} \geqslant 0$. Introducing the adequate Lagrange multipliers

$$L_1(\Theta, \lambda_1, \ldots, \lambda_{N_r}) = \sum_{(\mathbf{x}_k,\mathbf{y}_k)\in\mathbf{X}} \log \sum_j \Phi_j(\mathbf{x}_k) \sum_i \theta_{ij}\Gamma_i(\mathbf{y}_k) + \sum_j \lambda_j \left(1 - \sum_i \theta_{ij}\right) \tag{31}$$

and taking the partial derivatives with respect to θ_{ij} and λ_j, we obtain the following conditions that fulfill in the maximum:

$$\sum_{(\mathbf{x}_k,\mathbf{y}_k)\in\mathbf{X}} \frac{\Phi_r(\mathbf{x}_k)\Gamma_i(\mathbf{y}_k)}{\sum_i \sum_j \Phi_j(\mathbf{x}_k)\Gamma_i(\mathbf{y}_k)\theta_{ij}} = \sum_{(\mathbf{x}_k,\mathbf{y}_k)\in\mathbf{X}} \frac{\Phi_q(\mathbf{x}_k)\Gamma_i(\mathbf{y}_k)}{\sum_i \sum_j \Phi_j(\mathbf{x}_k)\Gamma_i(\mathbf{y}_k)\theta_{ij}}, \tag{32}$$

$$\sum_{i=1}^{N_\mathrm{o}} \theta_{ij} = 1 \quad \text{for } j = 1, \ldots, N_r; \;\; i, q, r = 1, \ldots, N_\mathrm{o}. \tag{33}$$

The pseudocode of the procedure we used to find the solution is shown in Fig. 14.

5.2. Numerical example

In Fig. 15, we plot the solutions of a modeling problem that is comparable in difficulty to the gaussian classification problem that was proposed in Section 4.2. 2000 points of the curve $y = x^2 + \epsilon$, where $\epsilon \to N(0, 1)$, were used to induce the different models. The 5×2cv test is again considered to perform the comparison.

In Figs. 16 and 17, the boxplot of the dispersion of the results and the numerical values of the mean test error are given. Optimal mean square error is 1 (the variance of the noise) and the linear solution (the model with constant output) has an error near 1.2, which is not far from the optimal solution.

```
minimize($\theta \in \mathbf{R}^{N_o \times N_r}$)
G,D $\in \mathbf{R}^{N_o \times N_r}$, $\alpha \in \mathbf{R}$, $i \in 1 \ldots N_o$, $j \in 1 \ldots N_r$
$\theta_{ij} = 1/N_o$
repeat
    $\mathbf{G}_{ij} = \sum_{(\mathbf{x}_k,\mathbf{y}_k)\in\mathbf{X}} \left(\sum_{i=1}^{N_o} \Phi_j(\mathbf{x}_k)\Gamma_i(\mathbf{y}_k)\right) / \left(\sum_{i=1}^{N_o}\sum_{j=1}^{N_r} \Phi_j(\mathbf{x}_k)\Gamma_i(\mathbf{y}_k)\theta_{ij}\right)$
    $\mathbf{D}_{ij} = \mathbf{G}_{ij} - (N_r \cdot N_o)^{-1} \sum_{i=1}^{N_o}\sum_{j=1}^{N_r} \mathbf{G}_{ij}$
    search $\alpha$ that minimizes $L$(normalize($\theta + \alpha \cdot \mathbf{D}$))
    $\theta =$ normalize($\theta + \alpha \cdot \mathbf{D}$)
until $\alpha * ||D|| < \epsilon$
end of minimize
normalize($\mathbf{X} \in \mathbf{R}^{N_c \times N_r}$)
    if ($\mathbf{X}_{ij} < 0$) $\mathbf{X}_{ij} = 0$
    $\mathbf{X}_{ij} = \mathbf{X}_{ij} / \sum_{i=1}^{N_c} \mathbf{X}_{ij}$
end of normalize
```

Fig. 14. Pseudocode of the numerical algorithm used to solve the set of equations (32) and (33). The linear search (determination of the value of α) was implemented with Brent's method. All points examined fulfill Eq. (33) because of the function `normalize`, and the algorithm stops when the conditions (32) are approximately true.

We applied linear, quadratic (which is the optimal model, in this case), neuronal, weighted least squares (WLS) and WM model, as well as our RSB model (Fig. 18).

6. Numerical results, classification

6.1. Experimental framework

We have selected five real-world problems, widely used in the machine learning literature: Iris (multiclass, low noise), Pima (two classes, moderate noise), Cancer (two classes, low noise), Glass (multiclass, high noise), Skulls (multiclass, very high noise). We applied an extension of the WM method modeling method to classification [2,5], ISH [11] and PM methods [14]. Statistical classification methods were linear and quadratical discriminant analysis, neural networks and nearest neighbor. When the number of rules was higher than 100, restricted probabilistic rule banks were generated with the method proposed here, to judge the influence of the number of rules in some problems.

5×2cv Dietterich's test [7] is to be applied again to judge the performance of the different classifiers. Datasets were randomly permuted first, the first half of samples was used to train the method and the second half to test it and then training and test sets were swapped and the learning and test phase repeated. This was repeated for five different random permutations. Training errors were

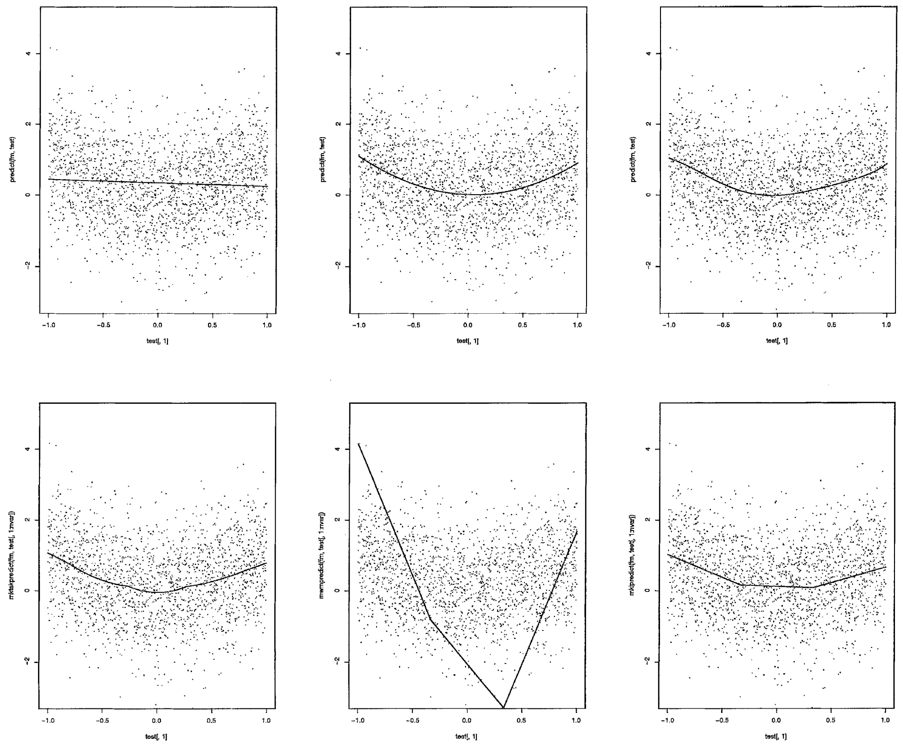


Fig. 15. From left to right, Upper part: linear, quadratical (optimal) and neuronal models. Middle: WLS, WM and RSB model. All rule-based models are built from size-4 partitions.

discarded so that boxplots show the dispersion of the error in test sets. Mean and median of test errors are tabulated for every experiment. We choose the median to estimate the mean test error because some of the learning methods produce somewhat disperse results.

6.2. *Iris*

This is a multiclass linear with low noise problem, and all methods perform correctly in it (Figs. 19–21). Probabilistic method is superior to ISH and PM, but similar to modified WM method. The differences between black-box methods and linguistic ones are negligible.

6.3. *Pima*

Pima's Indians diabetes is a two-class almost linear problem, with a moderate amount of noise. Contrary to Iris, WM method does not perform cor-

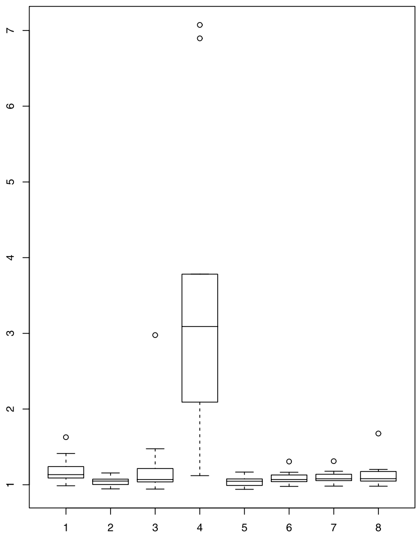


Fig. 16. Dispersion of test results after applying different methods to the problem discussed in the text. From left to right: linear, quadratical, neuronal, WM, RSB with 4, 3 and 2 rules.

Method	Error
Linear	1.199
Quadratical	1.044
Neuronal	1.292
WM	3.387
WLS	1.044
RSB (4 rules)	1.092
RSB (3 rules)	1.106
RSB (2 rules)	1.148

Fig. 17. Mean of test errors of linear, quadratical, neuronal, WM, WLS and RSB methods.

rectly here and it is worse than ISH and PM methods (Figs. 22–24). RSB method is superior to all fuzzy methods even if restricted to 65 rules. There is a significant loss of power in linguistic models with respect to black-box ones, thus direct comparison between them would not be correct in this case.

6.4. Cancer

Breast Cancer dataset has a low amount of noise and thus can be solved with low error with black-box methods. RSB method scores very well with 512 rules, but poorly with 51. RSB is better than WM, PM and ISH fuzzy methods in this case (Figs. 25–27).

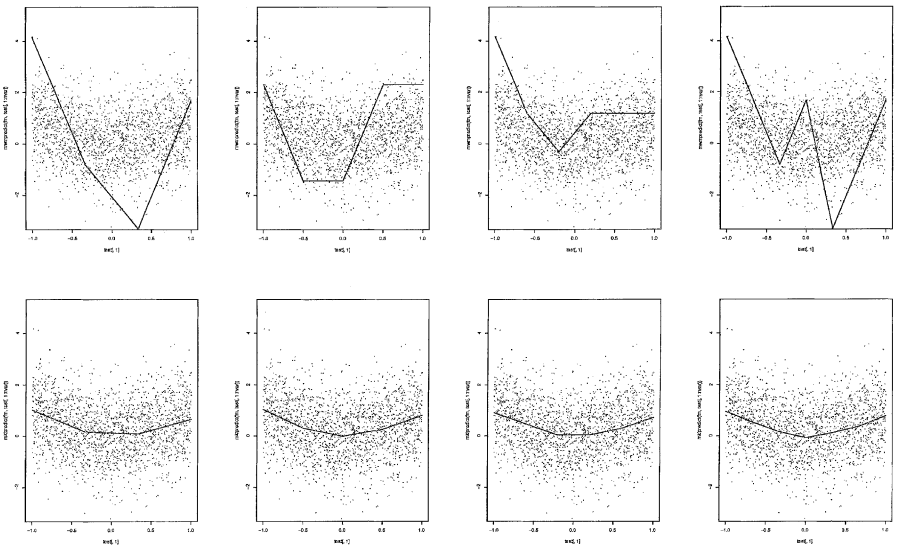


Fig. 18. Effect of the number of linguistic terms in RSB and WM methods. Upper part: WM method. Lower part: RSB. Both parts, from left to right: 4, 5, 6 and 7 terms/partition.

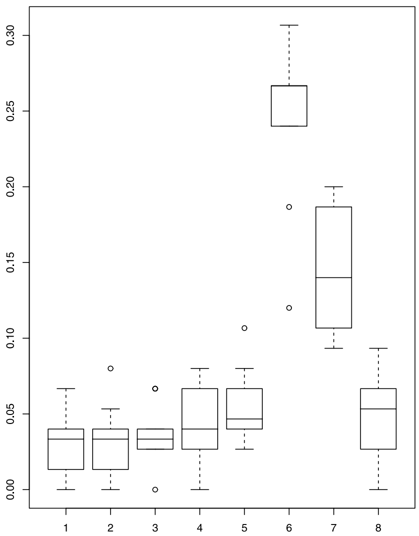


Fig. 19. Test results of Iris database with linear, quadratic, neuronal, 1-NN, fuzzy WM, ISH, PM and RSB classifiers with 60 rules.

	QUA	NEU	1NN	WM	ISH	PM	RSB
LIN	1	1	0.22	0.45	0.022	0.064	0.29
QUA		1	0.61	0.64	0.023	0.11	0.61
NEU			0.36	0.58	0.024	0.041	0.44
1NN				1	0.038	0.057	1
WM					0.058	0.10	1
ISH						0.29	0.025
PM							0.062

Fig. 20. p-Values of Iris classification problem.

	Mean	Median
LIN	0.0293	0.033
QUA	0.0306	0.033
NEU	0.036	0.033
1NN	0.041	0.040
WM	0.054	0.046
ISH	0.246	0.266
PM	0.144	0.140
RSB	0.046	0.053

Fig. 21. Mean and median of test errors in Iris problem.

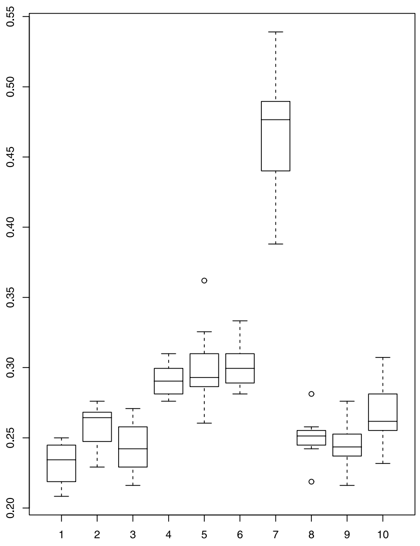


Fig. 22. Test results of Pima database with linear, quadratic, neuronal, 1-NN, fuzzy WM, ISH, PM and RSB classifiers with 6500 rules (column number 8), 650 rules (number 9) and 65 rules (number 10).

	QUA	NEU	1NN	WM	ISH	PM	RSB-1	RSB-2	RSB-3
LIN	0.0018	0.35	0.03	0.12	0.0039	0.0035	0.033	1	0.19
QUA		0.062	0.12	0.43	0.044	0.0051	0.068	0.018	0.8
NEU			0.027	0.17	0.0042	0.0048	0.2	0.6	0.31
1NN				0.58	0.34	0.045	0.038	0.022	0.066
WM					0.94	0.025	0.25	0.14	0.33
ISH						0.010	0.0011	0.00087	0.18
PM							0.0047	0.0031	0.020
RSB-1								0.05	0.62
RSB-2									0.28

Fig. 23. p-Values of Pima classification problem.

	Mean	Median
LIN	0.235	0.234
QUA	0.259	0.264
NEU	0.243	0.242
1NN	0.291	0.290
WM	0.300	0.292
ISH	0.301	0.299
PM	0.469	0.476
RSB-1	0.250	0.251
RSB-2	0.244	0.243
RSB-3	0.265	0.261

Fig. 24. Mean and median of test errors in Pima problem.

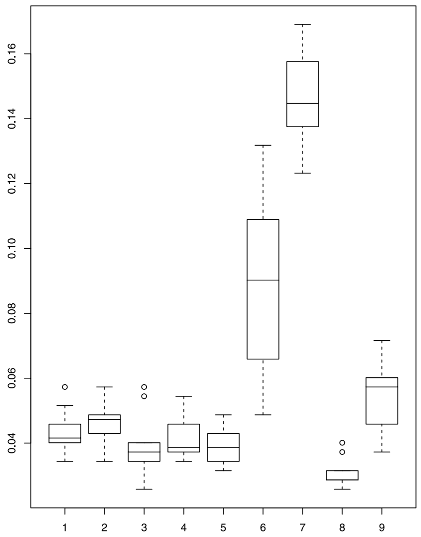


Fig. 25. Test results of Cancer database with linear, quadratic, neuronal, 1-NN, fuzzy WM, ISH, PM and RSB classifiers with 512 rules (column number 8), and 51 rules (number 9).

	QUA	NEU	1NN	WM	ISH	PM	RSB-1	RSB-2
LIN	0.74	0.72	1	0.25	0.24	0.00095	0.088	0.098
QUA		0.17	0.66	0.3	0.17	0.0096	0.12	0.47
NEU			0.68	0.83	0.14	0.0041	0.44	0.14
1NN				0.23	0.20	0.0018	0.078	0.14
WM					0.14	0.0011	0.10	0.088
ISH						0.34	0.11	0.47
PM							0.0012	0.0037
RSB-1								0.043

Fig. 26. p-Values of Cancer classification problem.

	Mean	Median
LIN	0.043	0.041
QUA	0.046	0.047
NEU	0.039	0.037
1NN	0.041	0.038
WM	0.039	0.038
ISH	0.089	0.090
PM	0.146	0.144
RSB-1	0.031	0.028
RSB-2	0.055	0.057

Fig. 27. Mean and median of test errors in Cancer problem.

6.5. *Glass*

This problem has a low number of samples for some classes, and this prevented some statistical methods from being directly applicable. Quadratic analysis is not directly applicable because the size of some classes is too small. Despite the apparently high number of rules, Glass dataset can be conveniently solved with less than 200 rules and there should not be significant differences between linguistic and black-box methods (Figs. 28–30). RSB is superior to WM, ISH and PM, and their models are roughly the same with 19 683 and 1968 rules.

6.6. *Skulls*

The Egyptian Skulls problem has a very high degree of noise, and it is normally used to contrast hypotheses about the influence of some factors in the class. The error of the best classifier is not much lower than the error of a random classifier (Figs. 31–33). There is no statistical evidence to support that one method is superior to all of them. Linguistic classifiers tend to perform worse than linear discriminant analysis.

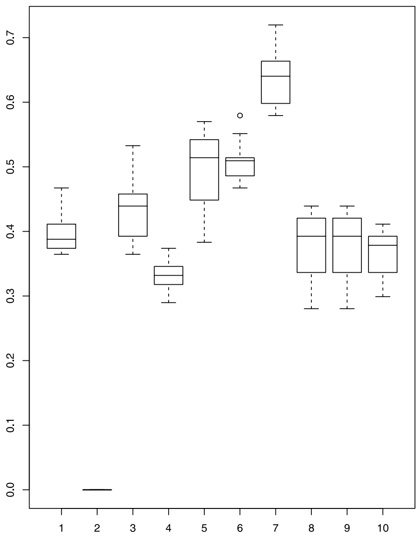


Fig. 28. Test results of Glass database with linear, quadratic (not applicable), neuronal, 1-NN, fuzzy WM, ISH, PM and RSB classifiers with 19 683 rules (column number 8), 1968 rules (number 9) and 196 rules (number 10).

	NEU	1NN	WM	ISH	PM	RSB-1	RSB-2	RSB-3
LIN	0.073	0.064	0.053	0.13	0.0025	0.45	0.45	0.14
NEU		0.052	0.3	0.78	0.012	0.24	0.24	0.083
1NN			0.0052	0.0042	0.0043	0.85	0.85	0.69
WM				0.42	0.36	0.063	0.063	0.010
ISH					0.069	0.020	0.020	0.0025
PM						0.019	0.019	0.0042
RSB-1							1	0.56
RSB-2								0.56

Fig. 29. p-Values of Glass classification problem.

7. Numerical results, modeling

7.1. Experimental framework

We have selected four problems. Two of them are synthetic, and the other two are real-world problems. We added different amounts of gaussian noise to synthetic data, in order to study the behavior of all methods with corrupted data.

	Mean	Median
LIN	0.397	0.387
NEU	0.440	0.439
1NN	0.334	0.331
WM	0.499	0.514
ISH	0.509	0.509
PM	0.635	0.640
RSB-1	0.375	0.392
RSB-2	0.375	0.392
RSB-3	0.367	0.378

Fig. 30. Mean and median of test errors in Glass problem.

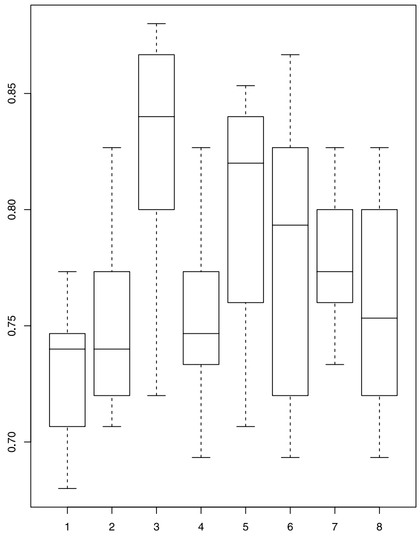


Fig. 31. Test results of Egyptian Skulls database with linear, quadratic, neuronal, 1-NN, fuzzy WM, ISH, PM and RSB classifiers.

Seven linguistic fuzzy models were evaluated over each dataset, plus classical regression (linear and quadratical), neural networks, WLS and RSB. 5×2cv Dietterich's test [7] was applied to judge the relevance of the differences as before. The boxplots show the dispersion of square error in test sets. Mean and median of test errors are tabulated for every experiment.

WM and CH fuzzy models are described in [3]. Both WM and CH select rules with the highest importance degree in groups defined by the antecedents. WM learning is guided from examples, CH learning is guided by a fuzzy grid

	QUA	NEU	1NN	WM	ISH	PM	RSB-1
LIN	0.4	0.061	0.22	0.85	0.61	1	0.020
QUA		0.35	0.34	0.82	0.79	0.37	0.16
NEU			0.75	0.55	0.9	0.23	0.47
1NN				0.25	0.59	0.093	1
WM					0.45	0.74	0.35
ISH						0.48	0.71
PM							0.08

Fig. 32. p-Values of Skulls classification problem.

	Mean	Median
LIN	0.729	0.74
QUA	0.749	0.74
NEU	0.829	0.84
1NN	0.754	0.75
WM	0.800	0.82
ISH	0.778	0.79
PM	0.778	0.77
RSB-1	0.757	0.75

Fig. 33. Mean and median of test errors in Skulls problem.

[1] and the importance degree is the maximum in type '1' methods, the mean in type '2' and the product between maximum and mean in type '3' methods. NIT models are taken from [13].

Weighted least squares (WLS) is a similar method to TSK fuzzy models. We used the same coverage functions for the random sets that defined the antecedents in descriptive models. "Consequents" are local linear models, fitted by least squares to a resampling of the training set in which every example appears a number of times proportional to the probability of being covered by the random set defined in its corresponding antecedent. This procedure does not yield linguistic rules, and it is expected to obtain results comparable to a neural network except when the points are not evenly distributed. In this case, this model is prone to overfit. In neural networks, this is solved by adding a regularity constraint, i.e., limiting the norm of the gradient of the estimated function. In WLS the most conservative assumption (norm of the derivative equal to 0 in all linear models) produces the NIT model, thus it is only expected to outperform WLS when data are unevenly distributed.

All methods but RSB learn by minimizing the square error over the training set, thus they rely on the residual being normally distributed. RSB should be superior when data are not evenly distributed and noise is not symmetrical.

7.2. Three-dimensional function F1

The first synthetic dataset comprises 676 points of the function $z = x^2 + y^2$ in $[0, 1] \times [0, 1]$. Linear regression is not included because its error is greater than the minimum by more than one order of magnitude in all cases. Gaussian noise with zero mean and standard deviation of 10%, 20%, 30% and 50% of the standard deviation of noiseless data was added. The boxplots of the residuals are shown in Fig. 34, whilst the mean and median of test errors in Fig. 35. The p-values of the 5×2cv test are displayed in Figs. 36 and 37. Observe in Fig. 37 that RSB method tends to be better when noise is high, while WM and CH fuzzy models achieve better results when noise is low. The quadratical model is the optimal one, in this case, as expected due to the original function shape.

7.3. Three-dimensional function F2

This dataset comprises 674 points of the function $z = 10 \cdot \frac{x - x \cdot y}{x - 2x \cdot y + y}$ in $[0, 1] \times [0, 1]$. The boxplots with the test errors are given in Fig. 38, the mean and median of them in Fig. 39 and the p-values in Figs. 40 and 41. The same conclusions drawn from the previous dataset can be applied here (see Fig. 41): probabilistic methods are better when the noise is high.

7.4. Function building-1

This dataset is taken from [15]. It is not a synthetic problem, but the amount of noise is very small. Since there are 14 inputs, the number of elements in each partition was reduced to 2 in order to keep the problem small enough to be linguistically understandable. The boxplot of the test error is shown in Fig. 42, the mean and median of errors are shown in Fig. 43 and the p-values of the comparison in Fig. 44. The best linguistic model is NIT in this case.

7.5. Electrical line length

This dataset was taken from [4,16]. It is a real-world problem with a moderate amount of noise. Data are sparse, thus WLS method performs poorly. There are significant differences between the mean and the median of test error, as can be seen in Fig. 46. The boxplot of the test error is shown in Fig. 45 and the p-values of the comparisons are shown in Fig. 47. The best linguistic model is the probabilistic one.

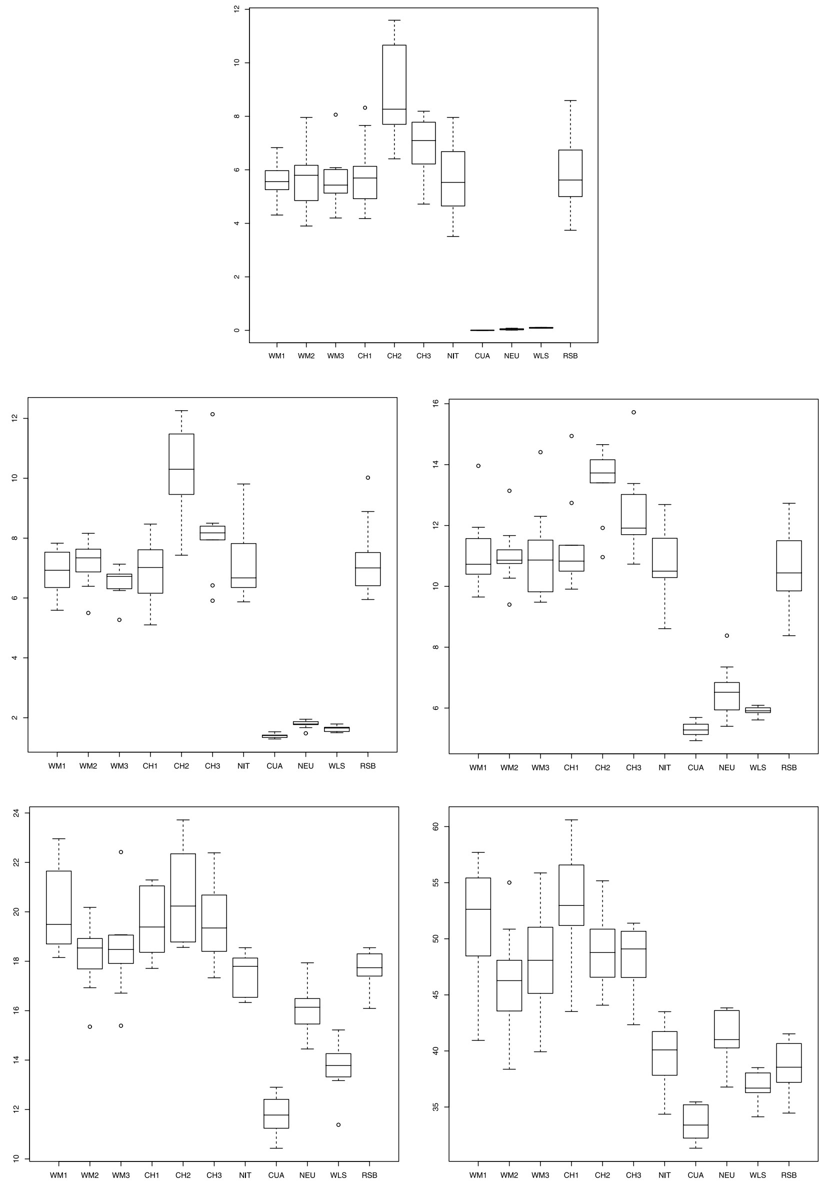


Fig. 34. Upper part: test error of WM-1, WM-2, WM-3, CH-1, CH-2, CH-3 and NIT fuzzy models, a quadratical model, a neural network, WLS and RSB models over the function $z = x^2 + y^2$. Middle and lower parts: the same experiments over the function plus 10%, 20%, 30% and 50% additive gaussian noise. Observe that the probabilistic model tends to be better when the noise is high.

	WM-1	WM-2	WM-3	CH-1	CH-2	CH-3	NIT	QUA	NEU	WLS	RSB
0%	5.65	5.73	5.57	5.82	8.90	6.93	5.63	0.00	0.04	0.09	5.78
10%	6.89	7.19	6.54	6.84	10.15	8.20	7.16	1.40	1.78	1.62	7.28
20%	11.07	10.99	11.06	11.33	13.45	12.42	10.63	5.29	6.57	5.90	10.51
30%	19.99	18.26	18.47	19.64	20.59	19.41	17.57	11.80	15.97	13.69	17.62
50%	51.78	46.40	47.80	53.48	48.94	48.16	39.65	33.53	41.40	36.76	38.59

	WM-1	WM-2	WM-3	CH-1	CH-2	CH-3	NIT	QUA	NEU	WLS	RSB
0%	5.56	5.79	5.43	5.69	8.26	7.09	5.53	0	0.05	0.1	5.62
10%	6.925	7.34	6.72	7.02	10.3	8.175	6.67	1.405	1.79	1.655	7.005
20%	10.725	10.86	10.865	10.83	13.725	11.915	10.5	5.28	6.52	5.91	10.44
30%	19.49	18.535	18.475	19.385	20.235	19.345	17.795	11.775	16.14	13.78	17.74
50%	52.63	46.27	48.085	52.965	48.77	49.095	40.095	33.395	41.01	36.69	38.55

Fig. 35. Upper part: mean of test errors over $F1$ dataset. Lower part: median of test errors.

	WM2	WM3	CH1	CH2	CH3	NIT	QUA	NEU	WLS	RSB
WM1	0.82	0.25	0.47	0.056	0.18	0.31	0.00093	0.00088	0.00093	0.15
WM2		0.044	0.66	0.058	0.29	0.42	0.0024	0.0022	0.0024	0.23
WM3			0.31	0.015	0.083	0.15	0.0043	0.0041	0.0045	0.10
CH1				0.13	0.64	0.76	0.0032	0.0033	0.0034	0.35
CH2					0.039	0.058	0.0022	0.0022	0.0023	0.18
CH3						0.76	0.0017	0.0016	0.0017	0.67
NIT							0.0057	0.0057	0.0059	0.053
QUA								0.15	0.0011	0.0052
NEU									0.051	0.0052
WLS										0.0053

	WM2	WM3	CH1	CH2	CH3	NIT	QUA	NEU	WLS	RSB
WM1	0.45	0.62	0.87	0.017	0.066	0.65	0.00054	0.0012	0.00068	0.85
WM2		0.16	0.42	0.0082	0.026	0.42	0.0011	0.0024	0.0014	0.56
WM3			0.64	0.0035	0.025	0.76	8e-05	0.00039	0.00013	1
CH1				0.0076	0.032	0.58	0.0055	0.0096	0.0068	0.76
CH2					0.93	0.011	0.00058	0.00096	7e-04	0.016
CH3						0.062	0.0025	0.0037	0.0028	0.067
NIT							0.016	0.017	0.019	0.29
QUA								0.22	0.013	0.011
NEU									0.53	0.012
WLS										0.013

	WM2	WM3	CH1	CH2	CH3	NIT	QUA	NEU	WLS	RSB
WM1	0.58	0.40	0.85	0.55	0.34	0.6	0.0096	0.11	0.019	0.53
WM2		0.4	0.38	0.37	0.15	0.65	0.011	0.13	0.018	0.57
WM3			0.26	0.17	0.032	0.98	0.069	0.28	0.086	0.88
CH1				0.62	0.47	0.49	0.027	0.14	0.036	0.43
CH2					0.9	0.24	0.0053	0.025	0.0032	0.20
CH3						0.36	0.012	0.082	0.015	0.31
NIT							0.063	0.11	0.072	0.31
QUA								0.37	0.39	0.087
NEU									0.46	0.15
WLS										0.10

Fig. 36. p-Values of 5×2cv test in $F1$ dataset for 0%, 10% and 20% of gaussian noise added.

	WM2	WM3	CH1	CH2	CH3	NIT	QUA	NEU	WLS	RSB
WM1	0.32	0.099	0.42	0.31	0.31	0.21	0.021	0.044	0.035	0.23
WM2		0.99	0.68	0.011	0.0064	0.66	0.0013	0.21	0.005	0.72
WM3			0.77	0.051	0.017	0.65	0.016	0.082	0.033	0.72
CH1				0.051	0.073	0.41	0.0025	0.12	0.0089	0.39
CH2					0.38	0.015	8.6e-05	0.018	0.00011	0.025
CH3						0.028	8e-04	0.020	0.0014	0.055
NIT							0.0018	0.043	0.0037	0.96
QUA								0.088	0.0026	0.0054
NEU									0.22	0.069
WLS										0.012

	WM2	WM3	CH1	CH2	CH3	NIT	QUA	NEU	WLS	RSB
WM1	0.46	0.35	0.32	0.23	0.56	0.025	0.011	0.19	0.042	0.024
WM2		0.93	0.19	0.28	0.99	0.10	0.017	0.31	0.068	0.052
WM3			0.17	0.39	0.96	0.074	0.013	0.31	0.06	0.044
CH1				0.057	0.17	0.0054	0.0010	0.032	0.0057	0.0028
CH2					0.072	0.19	0.017	0.43	0.077	0.092
CH3						0.036	0.003	0.13	0.016	0.0086
NIT							0.066	0.46	0.46	0.64
QUA								0.024	0.021	0.016
NEU									0.055	0.3
WLS										0.47

Fig. 37. p-Values of 5×2cv test in $F1$ dataset for 30% and 50% of gaussian noise added.

8. Concluding remarks and future work

A new family of linguistically understandable, probabilistic classifiers and models has been introduced. Our initial aim was not to improve the properties of fuzzy rule bases, but to study the advantages of fuzzy rule bases over classical techniques. Therefore, it was necessary to design models to which these techniques could be applied, while sharing a common structure with fuzzy rule bases. We finished up with a random set-based rule base, that is numerically identical to a Mamdani-type fuzzy rule base, except for the use of the t-norm product and the sum operation instead of the t-conorm. We expected the quality of both, fuzzy and probabilistic rule approaches, to be roughly equivalent; however, probabilistic models were not worse than the fuzzy methods studied, and often they were significantly better. Also, some inconsistencies were shown. There was no improvement in certain fuzzy learning algorithms when new examples were added to the datasets.

Our experimental results show that there exist very little differences between black boxes and probabilistic rules. These differences decrease with the number of rules and are statistically significant only when the rule base is rather small. Taking into account that our algorithm does not modify the fuzzy memberships in the antecedents, we can conclude that the effect of tuning the antecedents should be balanced against the right selection of rule importances. It

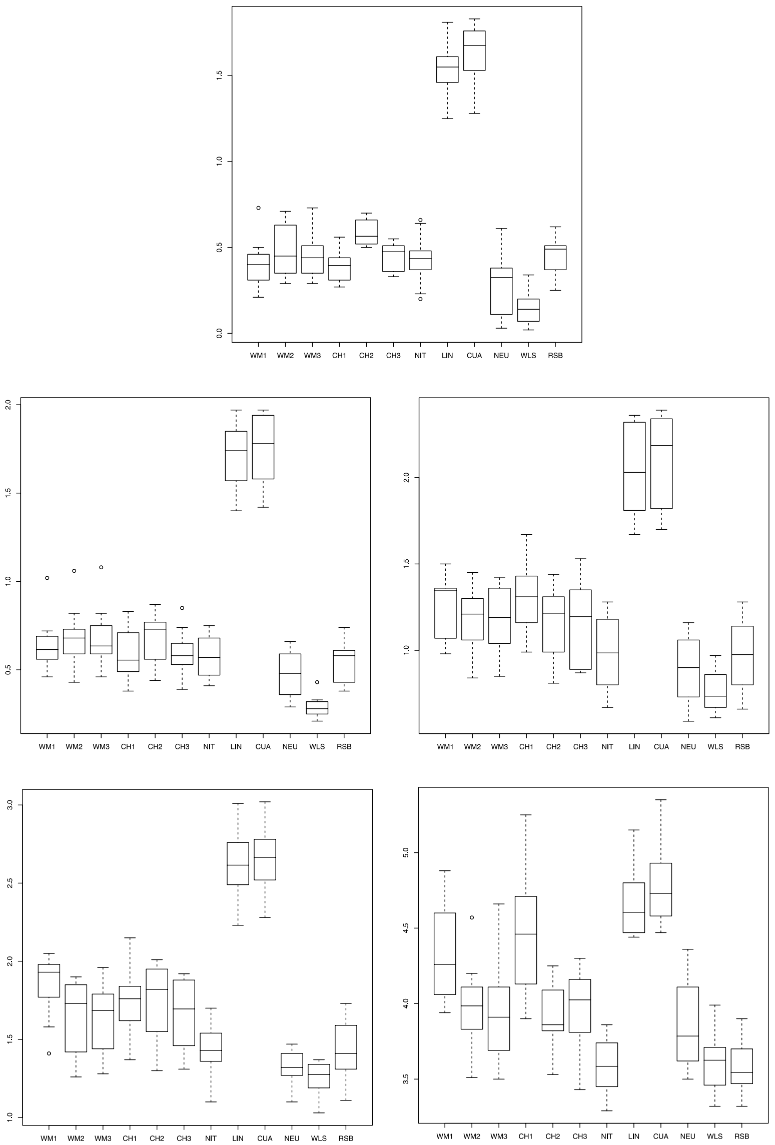


Fig. 38. Upper part: test error of WM-1, WM-2, WM-3, CH-1, CH-2, CH-3 and NIT fuzzy models, a linear model, a quadratical model, a neural network, WLS and RSB models over the function $z = 10 \cdot \frac{x - x \cdot y}{x - 2x \cdot y + y}$. Middle and lower parts: the same experiments over the function plus 10%, 20%, 30% and 50% additive gaussian noise. Probabilistic model tends to be better when the noise is high also in this case.

	WM-1	WM-2	WM-3	CH-1	CH-2	CH-3	NIT	LIN	QUA	NEU	WLS	RSB
0%	0.41	0.48	0.45	0.40	0.59	0.45	0.43	1.54	1.61	0.29	0.15	0.46
10%	0.64	0.68	0.68	0.59	0.68	0.60	0.58	1.71	1.75	0.48	0.29	0.55
20%	1.27	1.16	1.17	1.29	1.15	1.17	0.97	2.04	2.09	0.88	0.76	0.96
30%	1.84	1.65	1.63	1.76	1.75	1.67	1.45	2.63	2.66	1.31	1.25	1.43
50%	4.34	3.98	3.94	4.47	3.90	3.97	3.59	4.67	4.78	3.83	3.62	3.58

Fig. 39. Mean and median of test values in dataset $F2$.

	WM2	WM3	CH1	CH2	CH3	LIN	NIT	QUA	NEU	WLS	RSB
WM1	0.87	1	0.098	0.52	0.083	0.46	0.00025	0.00018	0.48	0.00069	0.11
WM2		0.73	0.30	0.74	0.11	0.65	0.00012	0.00011	0.66	0.035	0.44
WM3			0.14	0.56	0.072	0.54	0.00036	0.00029	0.56	0.009	0.31
CH1				0.18	0.16	0.4	0.00058	0.00059	0.78	0.065	0.44
CH2					0.011	1	0.0018	0.0016	0.73	0.011	0.68
CH3						0.18	0.0010	0.0011	0.36	0.52	0.15
LIN							2.1e-05	4.4e-05	0.77	0.054	0.47
NIT								0.42	0.0032	0.00023	9e-05
QUA									0.0031	0.00016	0.00010
NEU										0.14	0.95
WLS											0.027

	WM2	WM3	CH1	CH2	CH3	LIN	NIT	QUA	NEU	WLS	RSB
WM1	0.48	0.33	0.12	0.15	0.12	0.017	0.0034	0.0036	0.087	0.0029	0.03
WM2		0.81	0.049	0.067	0.037	0.0088	0.0024	0.0029	0.11	0.0062	0.012
WM3			0.084	0.12	0.16	0.022	0.0049	0.0054	0.088	0.0063	0.043
CH1				0.79	0.67	0.68	0.0013	0.0019	0.47	0.088	0.61
CH2					0.56	0.84	0.00011	0.00012	0.64	0.089	0.68
CH3						0.39	0.0017	0.0020	0.33	0.026	0.26
LIN							9.6e-05	8.8e-05	0.64	0.043	0.84
NIT								0.17	0.004	0.001	0.00018
QUA									0.0035	0.00087	0.00017
NEU										0.16	0.67
WLS											0.032

	WM2	WM3	CH1	CH2	CH3	LIN	NIT	QUA	NEU	WLS	RSB
WM1	0.01	0.11	0.41	0.11	0.046	0.012	0.037	0.036	0.0020	0.0027	0.014
WM2		0.76	0.046	0.61	0.26	0.029	0.0053	0.005	0.011	0.014	0.030
WM3			0.18	0.41	0.076	0.12	0.0061	0.0069	0.037	0.078	0.12
CH1				0.14	0.053	0.0094	0.17	0.16	0.012	0.0035	0.0056
CH2					0.32	0.27	0.0017	0.0015	0.044	0.18	0.28
CH3						0.58	9.2e-05	0.00012	0.26	0.43	0.57
LIN							0.00049	0.00037	0.30	0.52	1
NIT								0.38	0.0011	0.0048	0.00053
QUA									0.00093	0.0044	0.00041
NEU										0.86	0.36
WLS											0.5

Fig. 40. p-Values of test between all methods in $F2$ dataset, 0%, 10% and 20% of gaussian noise added.

has been argued that altering the semantic value of the linguistic terms produces rule bases whose meaning can be better understood, thus this method should be preferred, but a deeper study is needed. In the near future, we plan to

	WM2	WM3	CH1	CH2	CH3	LIN	NIT	QUA	NEU	WLS	RSB
WM1	0.57	0.86	0.36	0.9	0.71	0.05	0.044	0.039	0.036	0.0074	0.019
WM2		0.30	0.82	0.67	0.84	0.11	0.038	0.034	0.063	0.022	0.063
WM3			0.6	0.84	0.73	0.077	0.045	0.041	0.045	0.013	0.031
CH1				0.61	0.78	0.18	0.049	0.043	0.064	0.026	0.071
CH2					0.6	0.013	0.0013	0.0011	0.051	0.021	0.018
CH3						0.041	0.013	0.012	0.051	0.015	0.029
LIN							0.00015	0.00011	0.24	0.15	0.72
NIT								0.24	0.0047	0.0016	0.00054
QUA									0.0039	0.0013	0.00040
NEU										0.66	0.28
WLS											0.14

	WM2	WM3	CH1	CH2	CH3	LIN	NIT	QUA	NEU	WLS	RSB
WM1	0.79	0.51	0.46	0.65	0.82	0.16	0.41	0.25	0.36	0.3	0.12
WM2		0.35	1	0.75	0.78	0.083	0.13	0.059	0.12	0.27	0.053
WM3			0.64	0.73	0.41	0.33	0.015	0.0027	0.79	0.79	0.21
CH1				0.88	0.93	0.29	0.27	0.15	0.53	0.48	0.21
CH2					0.44	0.051	0.11	0.072	0.19	0.38	0.048
CH3						0.033	0.17	0.095	0.017	0.15	0.021
LIN							0.0021	0.0026	0.097	0.13	0.14
NIT								0.035	0.022	0.009	0.0015
QUA									0.014	0.0063	0.0018
NEU										1	0.041
WLS											0.06

Fig. 41. p-Values of $F2$, 30% and 50% of gaussian noise added.

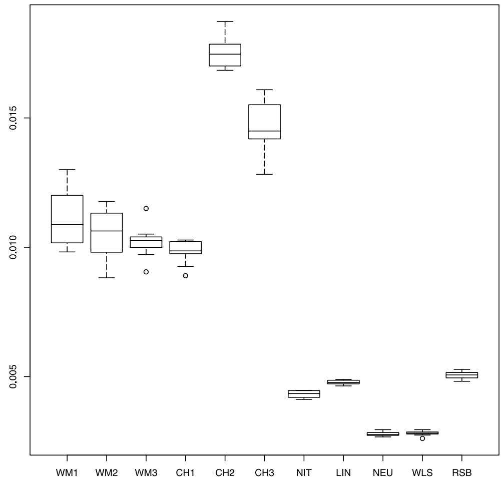


Fig. 42. WM-1, WM-2, WM-3, CH-1, CH-2, CH-3 and NIT fuzzy models, linear model, neural network, WLS and RSB models over the first output in the modeling problem "building" [15]. Only two linguistic elements partition each variable, in order to keep the size of the model small enough to be linguistically understandable.

	WM-1	WM-2	WM-3	CH-1	CH-2	CH-3	NIT	LIN	NEU	WLS	RSB
Mean	0.011	0.011	0.010	0.010	0.018	0.015	0.004	0.005	0.003	0.003	0.005
Median	0.011	0.011	0.010	0.010	0.017	0.014	0.004	0.005	0.003	0.003	0.005

Fig. 43. Mean and median of test errors in the "building" database.

	WM2	WM3	CH1	CH2	CH3	NIT	LIN	NEU	WLS	RSB
WM1	0.64	0.97	0.97	0.00016	0.014	0.003	0.0053	0.0013	0.0013	0.0061
WM2		0.72	0.77	0.00026	0.005	0.0020	0.0035	0.00097	1e-03	0.0038
WM3			0.98	1.9e-05	0.008	0.00013	0.00028	7.6e-05	6.9e-05	0.00025
CH1				0.00014	0.00039	9e-05	9.8e-05	2.6e-05	3.1e-05	0.00015
CH2					0.17	2.7e-06	5.9e-06	3.1e-06	2.7e-06	4.1e-06
CH3						0.00012	0.00015	7.5e-05	7.6e-05	0.00016
NIT							0.0042	9.2e-05	2e-05	8.5e-05
LIN								4e-07	4e-07	0.16
NEU									0.35	2.3e-05
WLS										4.7e-06

Fig. 44. p-Values of the comparison between all methods in the "building" problem.

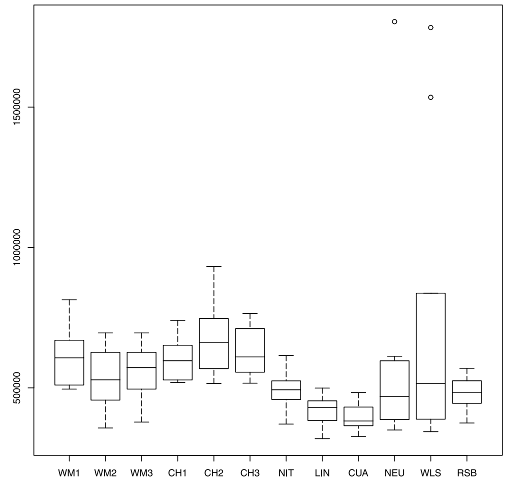


Fig. 45. WM-1, WM-2, WM-3, CH-1, CH-2, CH-3 and NIT fuzzy models, linear model, neural network, WLS and RSB models over the first output in the modeling problem "electrical line length".

	WM-1	WM-2	WM-3	CH-1	CH-2	CH-3	NIT	LIN	QUA	NEU	WLS	RSB
Mean	608	538	561	601	670	624	495	418	393	603	741	485
Median	607	528	572	596	662	610	493	430	382	469	516	484

Fig. 46. Mean and median of errors in the "electrical line length" problem.

	WM2	WM3	CH1	CH2	CH3	NIT	LIN	QUA	NEU	WLS	RSB
WM1	0.81	0.81	0.65	0.51	0.38	0.35	0.038	0.039	0.9	0.75	0.17
WM2		1	0.72	0.78	0.75	0.46	0.051	0.08	0.85	0.78	0.28
WM3			0.65	0.76	0.7	0.39	0.028	0.067	0.87	0.8	0.21
CH1				0.33	0.1	0.64	0.098	0.062	0.95	0.74	0.21
CH2					1	0.21	0.12	0.08	0.8	0.88	0.086
CH3						0.054	0.051	0.032	0.79	0.88	0.027
NIT							0.065	0.041	1.0	0.7	0.15
LIN								0.63	0.81	0.55	0.26
QUA									0.78	0.55	0.088
NEU										0.75	0.92
WLS											0.64

Fig. 47. *p*-Values of comparison between all methods for "electrical line length" problem.

compare both approaches and decide whether it is better to adjust the membership functions, and use models without rule importances, or to adjust the rule importances, and use models with fixed memberships.

The efficiency of the learning algorithm proposed in this work decreases when the number of variables is high, because it is necessary to store a vector containing all parameters in computer memory. The number of parameters grows with the product of the number of linguistic terms in all variables. This problem could be solved with an incremental algorithm, able to obtain rules one by one, that replaces the global minimization proposed in this work. Some recent results [8] suggest us that boosting classifiers [12] and backfitting models [17] are related to such an incremental version of the algorithm.

References

[1] R. Alcala, J. Casillas, J.L. Castro, A. Gonzales, F. Herrera, A multicriteria genetic tuning for fuzzy logic controllers, Mathware Soft Comput. 8 (2) (2001).

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

[3] O. Cordón, F. Herrera, A proposal for improving the accuracy of linguistic modeling, IEEE Transactions on Fuzzy Systems 8 (3) (2000) 335–344.

[4] O. Cordón, F. Herrera, L. Sánchez, Solving electrical distribution problems using hybrid evolutionary data analysis techniques, Applied Intelligence 10 (1) (1999) 5–24.

[5] O. Cordón, M.J. Del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 20 (1) (1999) 21–45.

[6] J.M. Fernandez-Garrido, I. Requena, A methodology for constructing fuzzy rule based classification systems, Mathware Soft Comput. 7 (2) (2000).

[7] G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Computation 10 (7) (1998) 1895–1924.

[8] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Annals of Statistics 28 (2) (2000) 337–374.

[9] D.J. Hand, Discrimination and Classification, Wiley, New York, 1981.

[10] S. Haykin, Neural Networks, Prentice-Hall, Englewood Cliffs, NJ, 1999.

[11] H. Ishibuchi, Distributed representation of fuzzy rules and its application to pattern classification, Fuzzy Sets and Systems 52 (1992) 21–32.
[12] L. Junco, L. Sanchez, Using the Adaboost algorithm to induce fuzzy rules in classification problems, in: Proc. ESTYLF, Sevilla, 2000, pp. 297–301.
[13] K. Nozaki, H. Ishibuchi, H. Tanaka, A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets and Systems, vol. 86, pp. 251–270.
[14] S.K. Pal, D.P. Mandal, Linguistic recognition system based in approximate reasoning, Information Sciences 61 (1992) 135–161.
[15] L. Prechelt, PROBEN1 – A set of benchmarks and benchmarking rules for neural network training algorithms, Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, 1994.
[16] L. Sánchez, Interval-valued GA-P algorithms, IEEE Transactions on Evolutionary Computation 4 (1) (2000) 64–72.
[17] L. Sánchez, A fast genetic method for inducting linguistically understandable fuzzy models, in: Proc. IFSA NAFIPS, 2001.
[18] R.E. Schapire, Theoretical views of boosting and applications, Lecture Notes in Artificial Intelligence 1720 (1999) 13–25.
[19] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Transactions on System, Man and Cybernetics 15 (1) 116–132.
[20] E. Trillas, C. Alsina, J. Terricabras, Introducción a la lógica borrosa, Ariel Matemática (1995).
[21] L.X. Wang, J. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man and Cybernetics 25 (2) (1992) 353–361.
[22] L.X. Wang, Adaptive Fuzzy Systems and Control, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[23] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man and Cybernetics SMC-3 (1973) 28–44.
[24] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Part I, Information Science 8 (1975) 199–249; Part II, Information Science 8 (1975) 301–357; Part III, Information Science 9 (1975) 43–80.
[25] L.A. Zadeh, Fuzzy languages and their relation to human and machine intelligence, in: Klir, Yuan (Eds.), Fuzzy Sets, Fuzzy Logic and Fuzzy Systems, World Scientific, Singapore, 1996, pp. 148–179.