A fuzzy definition of Mutual Information with application to the design of Genetic Fuzzy Classifiers

Luciano Sánchez
Dept. of Computer Science
University of Oviedo
33271 - Gijón, Spain
Email: luciano@uniovi.es

M. Rosario Suárez
Dept. of Computer Science
University of Oviedo
33271 - Gijón, Spain
Email: mrsuarez@uniovi.es

Inés Couso
Dept. of Statistics
University of Oviedo
33071 - Oviedo, Spain
Email: couso@pinon.ccu.uniovi.es

Abstract— **The equation used by fuzzy boosting algorithms to assign weights to rules precludes either the tuning of memberships while the rule base is been generated, or a final tuning stage, like that used in genetic iterative learning. In both cases, the tuning would need to alter the weights of all rules in the base, thus destroying the incremental nature of the learning. Therefore, if we are not given a semantic for the linguistic variables in the classifier, but want to tune the memberships of the variables, the calculus must be done in advance, via a discretization algorithm. Since the strength of boosting algorithms is in their ability to produce small rule bases, this discretization should be directed to preserve as much dependence as possible between the input and the output variables, with special interest in situations with a low number of labels. The statistical measure of this dependence is the mutual information.**

Contrary to its customary use, in this paper we suggest that the natural definition of the mutual information between a fuzzy variable and a crisp variable is a fuzzy number, and not a numerical value. Therefore, it is proposed a new definition of the statistical dependence between a fuzzified continuous variable and a crisp variable, that can be used in combination with boosting-related fuzzy rule learning algorithms in classification problems.

I. Introduction

Fuzzy boosting techniques are iterative rule learning methods that incrementally obtain fuzzy knowledge bases from data. The first fuzzy boosting algorithm was derived from AdaBoost [5][9], but there also exist a backfitting-based approach [12], and a version able to use standard max-min inference [13].

There are many similarities between fuzzy boosting and Iterative Rule Learning [3], but also some fundamental differences. In particular, IRL performs a tuning of the membership functions, but fuzzy boosting does not alter the shape or position of these functions. Conversely, knowledge bases learned by boosting contain weighted fuzzy rules. There are some works discussing the relative benefits of weighting rules vs. tuning memberships, and besides it can be argued that the semantic of a linguistic variable should not be altered in linguistically understandable classifiers [14], it is also clear that there exist problems where such an alteration is admissible. In this last case, fuzzy boosting could also benefit from an adequate selection of the fuzzy partitions of the input variables.

Contrary to IRL, in fuzzy boosting it is not easy to integrate the tuning of the membership functions with the learning algorithm. With this last algorithm, each time a rule is added to the base, a weight is calculated for it, as a function of the misclassified examples of the data base. This weight depends on the membership functions associated to the linguistic variables, thus any change on these functions would imply to recalculate the weights of the already emitted rules. This is obviously impractical, and we propose that the selection of the membership functions is done in advance, before any rules are emitted.

There exist many techniques to design fuzzy memberships (some of them will be reviewed in section III,) that could be used to solve this problem [3]. In this paper, we suggest that this decision can be guided by a measure of the loss of information that happens between the output and the input variables when the input data is discretized. This magnitude is the *Mutual Information* between the output variable and the input variables. The Mutual Information (M.I.) is a widely used measure of the statistical dependence between these magnitudes. Apart from its main use, in feature selection algorithms, the M.I. is useful to assess the quality of a discretization [2]. In addition, if combined with a search algorithm, this statistic serves to select the partition that loses the less information about the output variable, that will ultimately lead to the best classifier system. It will also be shown that, contrary to its customary use, the natural definition of the mutual information between a fuzzy variable and a crisp variable is a fuzzy number (and not a numerical value.)

The main objective of this work is to propose a new definition of the statistical dependence between a fuzzified continuous variable and a crisp variable, that can be used in combination with boosting-related fuzzy rule learning algorithms in classification problems. The structure of this paper is as follows: in the next section, fuzzy classifiers are introduced and it is explained how Adaboost can be applied to induce them from data. Then (section III) it is explained why a tuning

stage is not appropriate in combination with boosting, thus other techniques to obtain fuzzy memberships are reviewed and a design based on mutual information is proposed. The paper finishes with a graphical analysis of the properties of the boosting algorithm under fuzzy partitions of different quality.

II. Boosting Fuzzy Classifiers

A. Notation

At this point we introduce the basic notation employed throughout the paper. Let $\mathbf{X}$ be the feature space, and let $\mathbf{x}$ be a feature vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{X}$. Let p be the number of classes. The training set is a sample of m classified examples $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbf{X}$, $1 \leq y_i \leq p$, $1 \leq i \leq m$.

The antecedents of all fuzzy rules in the classifier form a fuzzy partition $\mathcal{A}$ of the feature space $\mathcal{A} = \{A^j\}_{j=1\ldots N}$, with $A^j \subset \widetilde{\mathcal{P}}(\mathbf{X})$, where $\widetilde{\mathcal{P}}(\mathbf{X})$ stands for "fuzzy parts of $\mathbf{X}$". In the remaining part of this paper, we will assume that the training examples will be indexed by the letter i, the rules by j, the features by f and the classes by k; the ranges of these variables are $1 \leq i \leq m$, $1 \leq j \leq N$, $1 \leq f \leq n$ and $1 \leq k \leq p$. For example, if we write "for all $\mathbf{x}_i$" we mean $\mathbf{x}_i$, $1 \leq i \leq m$; from now on, this range will not be explicitly stated unless its absence leads to confusion.

We will define a fuzzy rule based classifier by means of a fuzzy relationship defined on $\mathcal{A} \times \{1, \ldots, p\}$. Values of this relationship describe the degrees of compatibility between the fuzzy subsets of the feature space collected in $\mathcal{A}$, and each one of the classes. In other words, for every antecedent A^j we have p numbers between 0 and 1 that represent our degree of knowledge about the assert "All elements in the fuzzy set A^j belong to class number k". Values near to 1 mean "high confidence," and values near 0 mean "absence of knowledge about the assertion."

B. Linguistic interpretation of fuzzy classifiers

Fuzzy rule based classifiers are understandable to humans as they can be expressed as linguistic sentences. There are different standards when translating the former fuzzy relationship into linguistic statements. In this paper, we combine p instances of the fuzzy relationship,

$$\text{compatibility}(A^j, c_k) = s_k \quad k = 1, \ldots p,$$

into a single sentence, as follows:

$$\text{if } \mathbf{x} \text{ is } A^j \text{ then truth}(c_1) = s_1^j \text{ and } \cdots \text{ and truth}(c_p) = s_p^j$$

Furthermore, the antecedents of various rules with the same consequent

$$\begin{array}{c}\text{if } \mathbf{x} \text{ is } A \text{ then truth}(c_1) = s_1 \text{ and } \cdots \text{ and truth}(c_p) = s_p \\ \text{if } \mathbf{x} \text{ is } A' \text{ then truth}(c_1) = s_1 \text{ and } \cdots \text{ and truth}(c_p) = s_p\end{array}$$

can be combined with the help of the "or" connective, given a compound rule:

$$\begin{array}{c}\text{if } (\mathbf{x} \text{ is } A) \text{ or } (\mathbf{x} \text{ is } A') \text{ then} \\ \text{truth}(c_1) = s_1 \text{ and } \cdots \text{ and truth}(c_p) = s_p.\end{array}$$

In practical cases, we work with asserts A^j that can be decomposed in a Cartesian product of fuzzy sets defined over each feature, $A^j = A_1^j \times A_2^j \times \ldots \times A_n^j$, thus the rules are

$$\begin{array}{c}\text{if } (x_1 \text{ is } A_1^j \text{ and } \ldots \text{ and } x_n \text{ is } A_n^j) \text{ or } (x_1 \text{ is etc. }) \\ \text{then truth}(c_1) = s_1^j \text{ and } \cdots \text{ and truth}(c_p) = s_p^j.\end{array}$$

The linguistic expression of the fuzzy classifier does not include the terms for which confidence values are null. In case there exist fuzzy subsets for which all confidence values are null, the rule base will comprise less sentences (fuzzy rules,) than elements exist in the fuzzy partition $\mathcal{A}$.

We can restrict the definition further by defining n linguistic variables (one linguistic variable for every feature) and requiring that all terms sets A_f^j in the antecedents are associated with one linguistic term in its corresponding linguistic variable. In this case, we obtain a fuzzy rule based *descriptive* classifier. If we do not apply the latter restriction, we obtain an *approximate* classifier.

Observe that in a descriptive fuzzy classifier the set of possible rules is finite due to the discrete number of possible linguistic labels associated to each rule. Conversely, there is an infinite number of possible approximate classifiers as fuzzy rules use continuous parameters to define the characteristic points of their underlying fuzzy sets.

C. Fuzzy inference

Fuzzy reasoning methods define how rules are combined and how to infer from a given input to the corresponding output. The actual inference method is solely defined in terms of the fuzzy relationship, and is therefore independent of the classifier being approximate or descriptive. An instance $\mathbf{x}$ is assigned to the class

$$\arg\max_{k=1,\ldots,p} \bigvee_{j=1}^{N} A^j(\mathbf{x}) \wedge s_k^j \tag{1}$$

where "$\wedge$" and "$\vee$" can be implemented by different operators; for example, "$\vee$" can be the maximum operator [10] or the arithmetic sum, so called "maximum voting scheme" [8]. "$\wedge$" is always a t-norm, usually the minimum or the product. In this paper, we will combine the product with the maximum vote scheme to do the fuzzy inference.

D. The Adaboost algorithm

Let us define a set $\{g^1, g^2, \ldots, g^N\}$ of simple, but possibly unreliable binary classifiers. Boosting consists in combining these low quality classifiers (so called "weak hypotheses" in boosting literature) with a voting scheme to produce an overall classifier that performs better than any of its individual constituents alone. We will show later that a fuzzy rule can be regarded as a particular case of weak hypothesis, and a fuzzy rule base can be compared to a weighted combination of weak hypotheses.

Weak hypotheses take feature values as input and produce both a class number as well as a degree of confidence in the given classification. In two-class problems, these two outputs

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m), \quad \mathbf{x}_i \in \mathbf{R}^n,\ y_i \in \{-1, +1\}$
Initialize $D_1(i) = 1/m$
Select the number of weak hypotheses N
For $j = 1, \ldots, N$:

1) Get weak hypothesis $g^j : \mathbf{X} \to \mathbf{R}$
2) Find numerically the value α_j that minimizes $Z_j(\alpha) = \sum_{i=1}^{m} D_j(i) \exp(-\alpha y_i g^j(\mathbf{x}_i))$
3) Update the weights:

$$D_{j+1}(i) = \frac{D_j(i) \exp(-\alpha_j y_i g^j(\mathbf{x}_i))}{Z_j}$$

where Z_j is a normalization factor, so that D_{j+1} is a distribution.

Output the final hypothesis

$$H(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^{N} \alpha_j g^j(\mathbf{x})\right)$$

Fig. 1. Generalized Adaboost algorithm. Two classes version.

can be encoded with a single real number, $g^j(\mathbf{x}) \in \mathbf{R}$, whose sign is interpreted as the label of $\mathbf{x}$ and whose absolute value is interpreted as the confidence in the classification, the higher the better. AdaBoost is intended to produce a linear threshold of all hypotheses:

$$\text{sign}\left(\sum_{j=1}^{N} \alpha^j g^j(\mathbf{x})\right). \tag{2}$$

An outline of the Adaboost algorithm is shown in Figure 1. Observe that Adaboost can operate with any learning algorithm that generates a confidence rated classifier, given a weighted data set. There are different algorithms for assigning a number of votes to a weak hypothesis, and for adjusting the weights of the examples. For example, in confidence-rated Adaboost [15] the number of votes of the weak hypothesis g^j is given by the value α^j that minimizes the following function:

$$Z(\alpha) = \sum_{i=1}^{m} w_i \exp(-\alpha y_i g^j(\mathbf{x}_i)) \tag{3}$$

and the weights of the examples are updated according to the formula

$$w_i \leftarrow w_i \exp(-\alpha^j y_i g^j(\mathbf{x}_i))/v \tag{4}$$

where v is the value that makes $\sum w_i = 1$. There are analytical approximations and even heuristics that may replace this formula in specific problems.

E. Boosting fuzzy rules

Fuzzy rules are weak learners in fuzzy boosting. Each fuzzy rule is a confidence rated classifier that can produce the output '0' if the pattern is not covered by its antecedent, or both a class number and a confidence value between 0 and 1 else [5]. Therefore, boosting fuzzy rules can be based on an algorithm able to fit one single fuzzy rule to a set of weighted examples. This algorithm will be repeated so many times as rules in the base, and the Adaboost algorithm produces the number of votes each rule is assigned and recalculates the weight of every example when the rule is added to the base.

For the sake of simplicity, we restrict the discussion for the time being to two-class problems. A function $R_j(\cdot)$ can be assigned to the rule

$$\begin{array}{c}\text{if } x_1 \text{ is } A_1^j \text{ and } \ldots \text{ and } x_n \text{ is } A_n^j \\ \text{then t}(c_1) = s_1 \text{ and t}(c_2) = s_2\end{array}$$

$R_j(\mathbf{x})$ is defined as the product of the membership degree of instance $\mathbf{x}$ with the rule antecedent and the difference between the degrees of truth of the two classes in its consequent: $R_j(\mathbf{x}) = A^j(\mathbf{x})(s_1 - s_2)$. Assuming the product as the conjunction operator $\wedge$, the output of the fuzzy classifier given in eq. 1 can be written as

$$\text{sign}\left(\sum_{j=1}^{N} R_j(\mathbf{x})\right).$$

Noticing, the similarity between the above expression and eq. 2. it allows us to apply the boosting mechanism to descriptive fuzzy rules. The space of weak hypotheses becomes identified with the fuzzy partition $\mathcal{A}$. A linear threshold of elements of $\mathcal{A}$ is

$$\text{sign}\left(\sum_{j=1}^{N} \alpha_j A_j(\mathbf{x})\right)$$

and the values of α_j, along with the N elements A_j selected from $\mathcal{A}$ are obtained by the usual Adaboost algorithm. Positive values of α correspond to rules for which $s_1 > s_2$ and negative ones to rules with $s_2 > s_1$. Since the values of α_j that Adaboost produces are not constrained to the interval $[0, 1]$, it may happen that they no longer constitute valid confidence rates. Therefore, the degrees α_j in the consequents are normalized to a range $[-1, 1]$ once the entire rule base has been generated.

Figure 2 shows the outline of the final algorithm, as proposed in [9][5].

III. Membership tuning in fuzzy boosting

It is clear from Figure 2 that there is a strong dependence between the membership functions A^j and the weights α_j. A small change in any of the memberships implies a change in the degrees in the confidence of rule number 1, which in turn affects the weights of the examples and it might occur that not only the weight of the remaining rules, but also their linguistic expression, must be changed to keep a coherent set of weights. Obviously, a membership tuning, as stated in [3], can not be applied here.

Instead, we must resort to other techniques. The design of fuzzy membership functions was originally solved with statistical techniques based on frequencies, that were obtained by measuring the percentage of experts who answer "yes" to a question about whether an object belongs to a particular set, or who were asked directly to grade the object on an

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m), \quad \mathbf{x_i} \in \mathbf{R}^n,\ y_i \in \{-1, +1\}$
Initialize $D_1(i) = 1/m$, $s_1^j = 0$, $s_2^j = 0$
Select the number of rules N
For $j = 1, \ldots, N$:

- Find the fuzzy membership $A^j \in \mathcal{A}$ that minimizes $Z = \min_{A \in \mathcal{A}} (\sum_{i=1}^m D_j(i) \exp(-y_i A(\mathbf{x}_i)), \sum_{i=1}^m D_j(i) \exp(y_i A(\mathbf{x}_i)))$
- Find numerically the value α_j that minimizes $Z_j(\alpha) = \sum_{i=1}^m D_j(i) \exp(-\alpha y_i A^j(\mathbf{x}_i))$
- If $\alpha_j > 0$ then $s_1^j = \alpha_j$ else $s_2^j = -\alpha_j$.
- Update the weights:

$$D_{j+1}(i) = \frac{D_j(i) \exp(-\alpha_j y_i A^j(\mathbf{x}_i))}{K}$$

  where K is another normalization factor, so that D_{j+1} is a distribution.

End For
$s_1^j = s_1^j / \max_{k,j}(s_k^j), s_2^j = s_2^j / \max_{k,j}(s_k^j); k = 1, 2; j = 1, \ldots, N$
Generate the rules

if x_1 is A_1^j and $\ldots x_n$ is A_n^j then $\text{tr}(c_1) = s_1^j$ and $\text{tr}(c_2) = s_2^j$

Fig. 2. Adaboost algorithm applied to the induction of a descriptive, fuzzy rule based classification system. Two classes version.

scale [17]. According to [16], these *manual* techniques to estimate membership functions are four: direct rating, polling, set valued statistics and reverse rating.

Soft computing approaches to design or tune membership functions tend to leave the expert out of consideration; at most, the expert guesses an initial solution. The generation of the membership functions is guided to obtain the best numerical performance and linguistic concerns are often secondary. We could state that the primary objective of the *automatic* estimation of the membership functions is to reduce the classification error, thus they can be seen as *supervised* techniques [3].

On the other hand, there also exist *unsupervised* or *self-organizing* methods. These techniques have deep roots in information theory, and in fact some authors interpret all of them in terms of the principle of maximum information preservation [6]. These approach is less common in the fuzzy community; some authors use fuzzy clustering or neurofuzzy techniques [7], but the works that clearly relate the design of fuzzy memberships with the Information Theory are scarce [11]. It is remarked that these works should not be confused with the algorithms that use fuzzy memberships as a tool or 'soft histogram' when estimating the mutual information between continuous variables, which are an active area of research [4].

Information theoretic techniques are able to detect the statistical dependence between two variables, therefore its application in the design of fuzzy partitions is intuitive: it is clear that replacing a continuous variable by a linguistic variable carries a loss of information, thus the best discretization will be, roughly speaking, the one that maximizes the mutual information between the output and the input. But there are certain problems that arise in this procedure. The first, basic one, is that we are not aware of an extended definition of mutual information that can measure the statistical dependence between two discrete random variables, one of them imprecisely observed. We will address this problem in the following sections.

A. Mutual information between discrete random variables

Mutual information represents a general approach to determine the statistical dependence between variables. We are only concerned with discrete data, which we will show in later sections that can be precise or imprecisely observed.

For the time being, let us consider the definition of mutual information between two random variables, precisely observed. For a system A, with a finite set of M possible states $\{a_1, a_2, \ldots, a_M\}$, the Shannon entropy $H(A)$ is defined as [1]

$$H(A) = -\sum_{i=1}^{M} p(a_i) \log p(a_i) \tag{5}$$

where $p(a_i)$ denotes the probability of the state a_i. The entropy of the system A becomes zero if the outcome of a measurement of A is completely determined ($p(a_j) = 1$ and $p(a_i) = 0$ for all $i \neq j$) and becomes maximal if all probabilities are equal. The joint entropy $H(A, B)$ of two systems A and B is defined as

$$H(A) = -\sum_{i=1}^{M_A} \sum_{j=1}^{M_B} p(a_i, b_j) \log p(a_i, b_j) \tag{6}$$

and this leads to the relation

$$H(A, B) \leq H(A) + H(B) \tag{7}$$

with equality only in the case of statistical independence between A and B. The mutual information $\text{MI}(A, B)$ can be defined as

$$\text{MI}(A, B) = H(A) + H(B) - H(A, B) \tag{8}$$

and it is greater or equal than zero, being zero only when A and B are statistically independent.

The mutual information can be used to compare two different crisp discretizations, as shown in the example that follows. Observe that in this example all the values $p(a_i)$, $p(b_i)$ and $p(a_i, b_i)$ have been estimated by their relative frequencies $\hat{p}(a_i)$, $\hat{p}(b_i)$ and $\hat{p}(a_i, b_i)$.

Example 1: Let us suppose that we have to discriminate between three classes (apple, pear, banana), given the weight of a piece of fruit. To design a rule-based classifier, we are given a sample comprising five pieces, whose weights and classes are given in table I. The weight will be divided into three intervals, labeled "small", "medium" and "large". We have to choose the best design between two alternatives:

1) sma. $= [90, 100)$, med. $= [100, 110)$, lar. $= [110, 120]$
2) sma. $= [90, 110)$, med. $= [110, 115)$, lar. $= [115, 120]$

Solution:.

	crisp weight	class
1	111	pear
2	96	apple
3	116	pear
4	91	banana
5	101	apple

TABLE I
DATASET FOR THE EXAMPLE PROBLEM 'FRUIT'

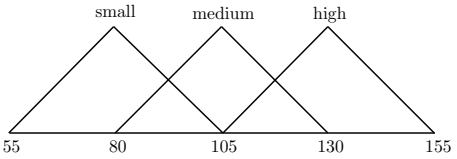

Fig. 3. One possible definition of the linguistic variable "weight", as used in the example problem "fruit."

The discrete values of the weight are as shown in the table that follows:

	weight	discrete 1	discrete 2	class
1	111	large	medium	pear
2	96	small	small	apple
3	116	large	large	pear
4	91	small	small	banana
5	101	medium	small	apple

The entropies of the variables 'discrete 1', 'discrete 2' and 'class' are, respectively,

$$H_1 = 0.4\log 0.4 + 0.2\log 0.2 + 0.45\log 0.4 = 1.0549$$

$$H_2 = 0.9503$$

$$H_3 = 1.0549$$

$$H_{1,3} = 1.3322$$

$$H_{2,3} = 1.3322$$

thus

$$\mathrm{IM}_{1,3} = H_1 + H_3 - H_{1,3} = 0.7776$$

$$\mathrm{IM}_{2,3} = H_2 + H_3 - H_{2,3} = 0.6730$$

so we can conclude that the first discretization keeps more of the information of the weight about the class, and is the preferred one. ■

B. Mutual information between a random variable and a random set

Let us suppose now that we are not given the value of the state of the system, but a set that contains it. In other words, we have a tolerance in our numerical measures, and therefore, when the value of the state is near the boundary of an element of the partition we can not assign it a label but two of them.

In this case, we do not know the values of the sample frequencies $\hat{p}(a_i)$, $\hat{p}(b_i)$ and $\hat{p}(a_i, b_i)$, but we can obtain sets of values that contain them, $\hat{p}(a_i) \in \Gamma_{a_i}$, $\hat{p}(b_i) \in \Gamma_{b_i}$ and $\hat{p}(a_i, b_i) \in \Gamma_{a_i,b_i}$. We can not estimate the mutual information from a sample, but we can find a set that contains it:

$$\begin{aligned}\overline{\mathrm{MI}}(A,B) = \{&\textstyle\sum a_i \log a_i + \sum b_i \log b_i \\ &- \textstyle\sum\sum c_i \log c_i : a_i \in \Gamma_{a_i},\ b_i \in \Gamma_{b_i}\ c_i \in \Gamma_{a_i,b_i}\}\end{aligned} \tag{9}$$

This set will be formed by all the estimations of the mutual information that are compatible with our knowledge. This is made clear with the example that follows

Example 2: Recall the previous example. Suppose that we use the first discretization, and the following set of data:

	weight	discrete values	class
1	110 ± 1	large or medium	pear
2	96 ± 1	small	apple
3	116 ± 1	large	pear
4	91 ± 1	small	banana
5	101 ± 1	medium	apple

Solution:. There are two sets of estimates that are compatible with the example:

1) $\hat{p}(\text{large}) = 0.4$, $\hat{p}(\text{medium}) = 0.2$, $\hat{p}(\text{small}) = 0.4$, $\hat{p}(\text{large,pear}) = 0.4$ etc.
2) $\hat{p}(\text{large}) = 0.2$, $\hat{p}(\text{medium}) = 0.4$, $\hat{p}(\text{small}) = 0.2$, $\hat{p}(\text{large,pear}) = 0.2$ etc.

Operating with the first set, the mutual information is 0.7776, and the second one gives the value 0.6730, therefore all we can say about the amount of information that the discrete input gives about the class number in that it is in the set $\overline{\mathrm{IM}} = \{0.6740, 0.7776\}$. Observe that, if a second example would have lied in a boundary, we would have obtained four sets of estimates, and so on. ■

C. Mutual information between a random variable and a fuzzy random variable

Recall that in the previous section we estimated a set of values of the mutual information given a sample comprising sets of labels, i.e. "weight is large or medium". Now we want to generalize it to the fuzzy case, where a numerical value is transformed in a *fuzzy set over the set of labels*, say "weight is large with degree 0.8" and "weight is medium with degree 0.2."

It is immediate that, in this last case, all we can estimate about the mutual information is a fuzzy restriction of its value. The standard construction applies, and for each α-cut we obtain a problem like that stated in the section before, whose solutions can be stacked again to produce the fuzzy output. The whole process is illustrated in the next example:

Example 3: Let us evaluate the mutual information between the linguistic variable "weight", (understanding "linguistic variable" in the fuzzy sense,) as defined in Figure 3, and the crisp variable "class."

Solution:. The values of the memberships of the examples in table I to the elements of the partition in Figure 3 are as follows:

Example	value	small	medium	high
1	111	0	0.88	0.12
2	96	0.18	0.82	0
3	116	0	0.78	0.22
4	91	0.28	0.72	0
5	101	0.08	0.92	0

There are $2^5 = 32$ different combinations of the discrete labels "small, "medium" and "high" that are compatible with these values. For example, one of them is

Example	num value	linguistic value	membership
1	111	medium	0.88
2	96	small	0.18
3	116	medium	0.78
4	91	small	0.28
5	101	small	0.08

for which the value of the mutual information between the class and the linguistic value is 0.673, and the membership of the value 0.673 to the fuzzy mutual information will not be lower than 0.08. Repeating the process for all the 32 combinations, the result is $\widetilde{\mathrm{MI}} = 0.72/0 + 0.22/0.22 + 0.18/0.40 + 0.28/0.50 + 0.22/0.67 + 0.12/0.78 + 0.12/1.06$. Different partitions have different fuzzy mutual informations. For example, if the fuzzy sets "small", "medium" and "high" are respectively defined as $(78.5; 91; 103.5)$, $(91; 103, 5; 116)$ and $(103.5; 116; 128.5)$, the estimation is $\widetilde{\mathrm{MI}} = 0.40/0.50 + 0.4/0.67 + 0.6/0.78 + 0.4/1.05$ which, under many criteria, is preferable to the former one.

It is remarked that, if we had restricted ourselves to the most compatible of the 32 combinations,

Example	num value	linguistic value	membership
1	111	medium	0.88
2	96	medium	0.82
3	116	medium	0.78
4	91	medium	0.72
5	101	medium	0.92

the punctual estimation of the mutual information is 0, thus if we had dropped the fuzziness of the estimation of the mutual information, we could have concluded (wrongly) that this linguistic partition completely loses the dependence w.r.t. the class number. In an informal sense, the values of the mutual information with lower memberships describe the amount of information carried by the inputs with are less covered by the fuzzy partition, which can be highly relevant to the classifier being designed. ■

IV. Numerical analysis

The practical application of the concepts mentioned in the preceding section pose some difficulties:

- The number of times that the mutual information must be estimated grows exponentially with the number of inputs, and with the number of elements in the training set.
- We have reasoned that the estimation of the mutual information associated to a partition is a fuzzy set. But, when two different partitions are compared, depending on how overlapped their estimations are, it is not immediate how to select one, and we must use a fuzzy ranking [18] or a multicriteria search.

Figure	Fuzzy partition	centroid of IM
A	(-1;0;1)	0.247
B	(-2;0;-2)	0.278
C	(-3;0;-3)	0.345
D	**(-4;0;-4)**	**0.361**
E	(-5;0;-5)	0.273
F	(-6;0;-6)	0.197

TABLE II
Values of the centroid of the fuzzy mutual information for the partitions evaluated in Figure 4

In this paper, whose results have a preliminary nature, the solutions that we have applied to these problems are:

- Only a fraction of the points of the training is used to estimate the mutual information. In the experiments shown in this section, a 1% of the points was chosen.
- Not all compatible combinations of labels need to be evaluated. We have restricted ourselves to those combinations for which no examples have a membership under 0.45. This restriction and the preceding one make it possible to evaluate a fuzzy partition with less that ten thousand evaluations of the mutual information. The restriction may seem excessive, but we intend to include this procedure in a genetic algorithm that automates the search.
- Since the fuzzy part of the estimation of the mutual information carries information about the examples that are less covered by the fuzzy labels, it is meaningful to use a fuzzy ranking that gives more weight to the values of mutual information that with higher membership. We have chosen Yager's ranking [18], the same centroid-based defuzzification used in control systems.

The problem we have selected to evaluate the method is the Gauss problem, proposed in [6]. 4000 points taken from two overlapping bi-dimensional Gaussian distributions (centered in $(0, 0)$ and $(2, 0)$) with different covariance matrix (I and $4I$). The Gauss problem is quadratic. The density of the points in the left part (see Figure 4) is lower than the density in the right area, and contributes very little to the global error, thus small changes can produce large deviations of the decision surface.

We have chosen to do a graphical analysis, which gives us more insight into the method than a numerical table. In Figure 4 we have plotted the optimal decision surface (dotted line) and the decision surface obtained by the boosting algorithm when the fuzzy partitions shown in Table II are used. The partition that is used in the original implementation of fuzzy Adaboost is labeled 'F'. Observe that the value of the centroid of the mutual information is highly correlated with the output obtained by fuzzy Adaboost, as desired. Further experimentations are needed, but these preliminary results are coherent with both the definition of the fuzzy entropy and the fuzzy ranking that was selected.

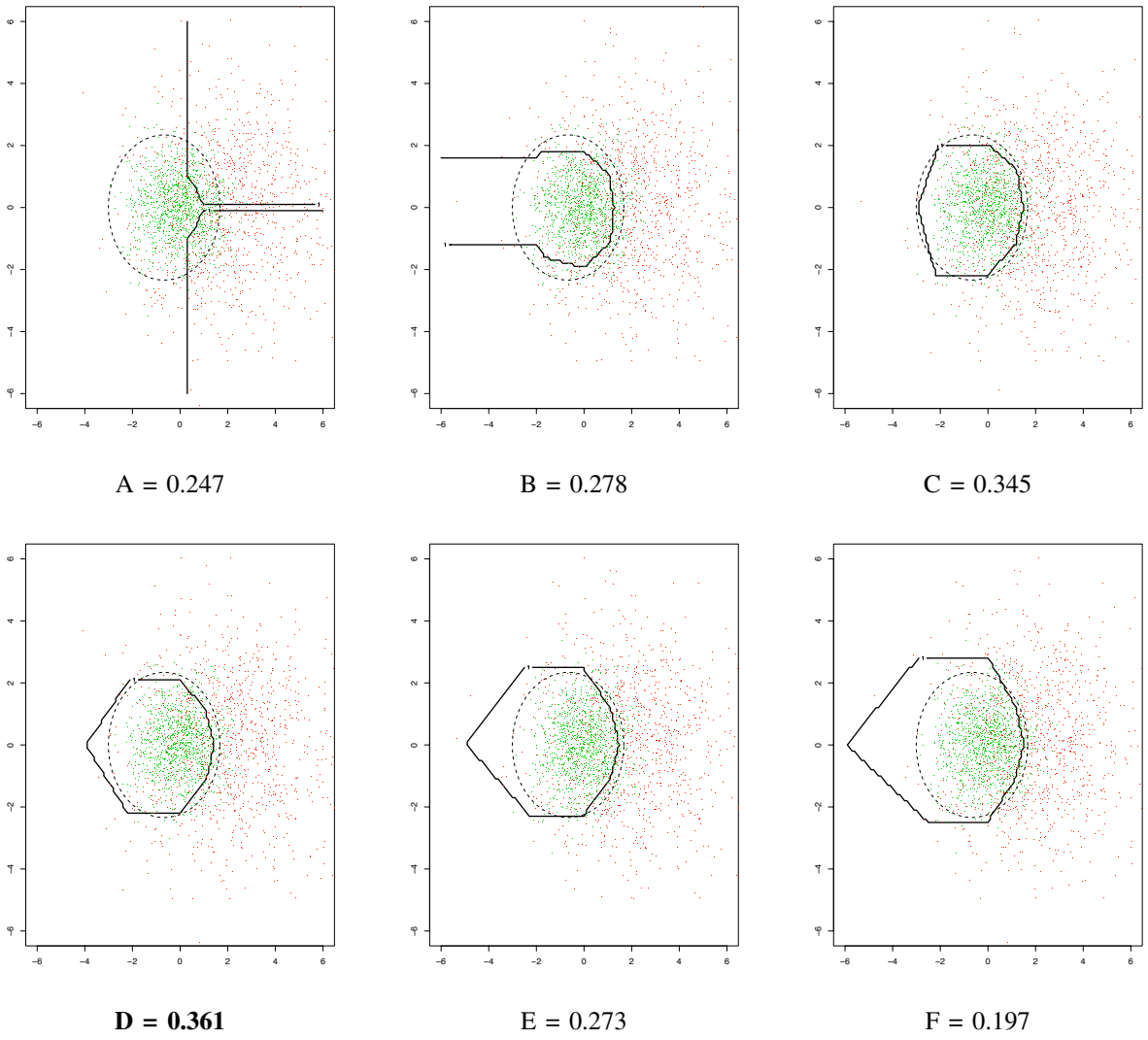

Fig. 4. Decision surfaces obtained by the fuzzy Adaboost algorithm for the partitions mentioned in Table II. The partition which keeps the highest amount of information, according to our estimator, is labeled 'D'. The numbers shown beside the labels are the centroids of the fuzzy mutual information of the input variables about the class.

V. Concluding remarks and future work

Most of the soft computing techniques suitable to design or tune fuzzy membership functions are supervised, and oriented to reduce the classification error of an specific algorithm. In this paper we have proposed an unsupervised criterion, unrelated to this error, to select the fuzzy partitions. We have needed an unsupervised algorithm because we intend to combine it with fuzzy boosting, for which we have stated that a tuning is not appropriate.

As a result, it has been proposed a fuzzy quality index of a partition that takes into account how much information is lost when certain memberships are used. The usefulness of this approach might transcend its application to fuzzy boosting: observe that, contrary to most other criteria found in the literature, this index copes in a natural way with missing values (that can be codified with a membership of 0.5 for all labels in the corresponding variable) and is not based in heuristics neither in simplifications of the problem.

The main drawbacks of the algorithm are in its exponential growth in time and memory with the dimension of the training set. In future works, we intend to design a resampling based estimator of the fuzzy mutual information, integrate it in a genetic search and test it in various benchmark problems.

Acknowledgments

This work was funded by Spanish M. of Science and Technology and by FEDER funds, under the grant TIC-04036-C05-05.

References

[1] Ash, R. Information Theory. Dover. 1965
[2] Bacardit, J. Pittsburgh Genetic Based Machine Learning in the Data Mining Era: Representations, Generalization, and Run Time. Ph. D. Thesis. La Salle-Univ. Ramón Llull, 2005.

[3] Cordón, O., Herrera, F., Hofmann, F., Magdalena, L. Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases. Advances in Fuzzy Systems, Applications and Theory, Vol 19. World Scientific. 2001
[4] Daub., C., Steuer, R., Selbig, J., Kloska, S. "Estimating mutual information using B-spline functions - an improved similarity measure for analysing gene expression data". BMC Bioinformatics 2004, 5:118. 2004.
[5] Del Jesus, M. J., Hoffmann, F., Junco, L., Sánchez, L. Induction of Fuzzy Rule Based Classifiers with Evolutionary Boosting Algorithms. IEEE Transactions on Fuzzy Sets and Systems 12 (3): 296-308, 2004.
[6] Haykin, S. *Neural Networks*. Prentice Hall, 1999.
[7] Hong, T., Lee, C. Induction of fuzzy rules and membership functions from training examples. Fuzzy Sets and Systems 84, 1, pp 33-47. 1996.
[8] Ishibuchi, H., Nakashima, T. and Morisawa, T., Voting in fuzzy rule-based systems for pattern classification problems. Fuzzy Sets and Systems, vol 103, no. 2, pp 223-239, 1999.
[9] Junco, L., Sanchez, L."Using the Adaboost algorithm to induce fuzzy rules in classification problems", Proc. ESTYLF 2000, Sevilla, pp 297-301. 2000.
[10] Kuncheva, L. I. Fuzzy Classifier Design. Springer-Verlag, NY, 2000.
[11] Makrehchi, M. , Kamel, M. An Information Theoretic Approach to Generate Membership Functions from Real Data, North American Fuzzy Information Processing Socierty- NAFIPS 2003, Chicago, USA, 2003
[12] Otero, J., Sánchez, L. Induction of descriptive fuzzy classifiers with the Logitboost algorithm. Admitted for publication in Soft Computing.
[13] Sánchez, L., Otero, J. Boosting fuzzy rules in classification problems under single-winner inference. Admitted for publication in the International Journal of Intelligent Systems.
[14] Sánchez, L., Otero, J. Tuning fuzzy partitions or assigning weights to fuzzy rules: which is better? Accuracy Improvements in Linguistic Fuzzy Modeling. J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.). Physica-Verlag. pp 266-386. 2003
[15] Schapire, R., Singer, Y. Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning 37(3): pp. 297-336. 1999
[16] I.B. Turksen. Measurement of membership functions and their acquisition. Fuzzy Sets and Systems, 40:5–38, 1991
[17] N. Watanabe. Statistical Methods for Estimating Membership Functions. Japanese Journal of Fuzzy Theory and Systems, 5(4), 1979
[18] Xang, X., Kerre, E.E. Reasonable properties for the ordering of fuzzy quantities. Fuzzy Sets and Systems 118, 3. pp 375-386. 2001